

Manual del Usuario de Marxan

Para la version Marxan 1.8.10

Escrito por Edward T. Game y Hedley S. Grantham

Editado por Jeff Ardron, Carissa Klein, Dave Nicolson, Hugh Possingham y Matt Watts

Traducido al español por Milay Cabrales

Versión al español revisada y corregida por José L. Gerhartz

Publicado: Febrero 2008

Publicado de conjunto por:

Centro para el Análisis de las
Decisiones Medioambientales
Aplicadas
El Centro de Ecología
La Universidad de Queensland
St. Lucia, Queensland, Australia

Asociación para la Investigación y
Análisis Marino del Pacífico
(PacMARA, por sus siglas en
inglés)
Vancouver, British Columbia,
Canadá



Applied Environmental Decision Analysis
Commonwealth Environmental Research Facility



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA



PacMARA
Pacific Marine Analysis
& Research Association

Cita recomendada:

Game, E. T. y H. S. Grantham. (2008). Manual del Usuario de Marxan: Para la versión Marxan 1.8.10. Universidad de Queensland, St. Lucia, Queensland, Australia, y la Asociación para la Investigación y Análisis Marino del Pacífico, Vancouver, British Columbia, Canadá.

(Esta página está en blanco intencionalmente)

Prólogo

Perspectiva y Objetivos

Con este manual se pretende brindar a los lectores el conocimiento básico necesario para utilizar Marxan. Abarcamos todos los parámetros pertinentes y las entradas de información necesarias, así como los pasos que se requieren para ejecutar el programa de manera satisfactoria e interpretar sus resultados. . Nos centramos más en los aspectos prácticos del uso de Marxan que en la teoría del diseño de sistemas de reservas y en los algoritmos de optimización que solucionan este problema de sistema de reservas. Parte de esta información se encuentra disponible en los apéndices, aunque en otros casos intentamos guiar a los lectores hacia la fuente apropiada de información. En este manual se ofrece una guía para el tipo de problemas que se pueden solucionar con Marxan pero, en realidad, únicamente su imaginación establece las fronteras sobre cómo puede ser aplicado. Sin embargo, es importante comprender como funciona Marxan para evitar dar solución al problema equivocado o malinterpretar las soluciones que se puedan encontrar. Marxan puede resultar una herramienta muy poderosa, pero su mala utilización, afectaría un gran cúmulo de trabajo complejo, consistente en la recopilación y confrontación de información exacta, sin mencionar que puede conducir a consejos erróneos y disminuir la credibilidad de este software de planificación sistemática de la conservación.

Información complementaria

En ocasiones puede parecer que el manual no ahonda en detalles o carece de una dirección específica en determinados temas, sin embargo, para esos casos, casi con total seguridad se podrá encontrar información adicional en la Guía para las Buenas Prácticas de Marxan (MGPH, por sus siglas en inglés), que deberá estar disponible para su descarga en el 2008. Este manual y la MGPH deberán ser utilizados de conjunto, así brindarán los recursos necesarios para llevar a cabo un análisis bien documentado, con un alto nivel de experiencia haciendo uso de Marxan. Asimismo, consideramos muy recomendable la lectura de algunos de los tantos artículos similares que se han publicado y que utilizan Marxan para diversas aplicaciones de conservación. En esos artículos se demuestran los tipos de interrogantes que se pueden solucionar con Marxan, como se establecen los problemas de conservación, el tipo de información que puede ser utilizada y como diferentes objetivos y limitaciones influyen en las soluciones de reserva que se obtienen. Detalles referentes a algunas de esas publicaciones pueden encontrarse en la sección de Referencias Principales.

Agradecimientos

La preparación de esta versión del manual de Marxan ha resultado ser tanto un ejercicio de recopilación como de escritura, y como resultado debemos agradecer profundamente a un gran número de personas.

En primer lugar y de manera destacada, debemos reconocer a Ian Ball y Hugh Possingham, los promotores originales de Marxan y autores del manual anterior. Es con el uso de ese manual que tanto nosotros, como muchos otros, hemos aprendido sobre el uso de Marxan. Mucha de la información contenida en su manual original ha sido incorporada en esta versión, específicamente en el apéndice técnico.

Queremos agradecer también a Jeff Ardron por su importante contribución. Jeff no sólo ofreció valiosa asesoría técnica y de redacción, sino que sin su entusiasmo y confianza en las buenas prácticas de conservación, ni este manual ni la guía para las buenas prácticas se hubieran hecho realidad. Asimismo, queremos agradecer a Carissa Klein, Dave Nicolson y Matt Watts por la excelente asesoría técnica y de redacción ofrecida durante el proceso. El Apéndice C basa su información mayormente en el curso Marxan 101 impartido por la Universidad de Queensland, redactado por Carissa Klein y Matthew Watts. Detalles referentes a este y otros cursos pueden encontrarse en <http://www.ecology.uq.edu.au/marxan.htm>.

Algunos de los párrafos introductorios han sido modificados del capítulo 1 del MGPH, va nuestro agradecimiento por ello a Hugh Possingham, Jennifer Smith, Krista Royle, Dan Dorfman y Tara Martin.

Lindsay Kircher y Dan Segan, en su condición de usuarios nuevos de Marxan, brindaron comentarios inteligentes y útiles en un borrador previo de este manual y durante sus instrucciones. Finalmente, extendemos nuestro agradecimiento a los que financiaron este trabajo, el Packard Foundation's Marine Ecosystem-Based Management Tool Innovation Fund (*Fondo de Innovación de Herramientas para el Manejo basado en el Ecosistema Marino, de la Fundación Packard*), y a la Asociación para la Investigación y Análisis Marino del Pacífico (PacMARA), por presentar la propuesta de financiamiento.

Nosotros también querríamos dar gracias WWF-CANADA que permitió José L. Gerhartz para dedicar tiempo a la traducción española de este manual.

Tabla de contenidos

Prólogo	i
Perspectiva y Objetivos.....	i
Información complementaria.....	i
Agradecimientos	ii
Tabla de contenidos	iii
Listar a Tablas:	vi
1. Introducción	1
1.1 ¿Qué es Marxan?	1
1.1.1 Otras Versiones de Marxan.....	3
1.2 Planificación Sistemática de la Conservación.....	3
1.3 Interrogantes que Marxan puede ayudar a resolver	4
1.4 Limitaciones de Marxan	5
1.5 La función objetivo	5
1.6 Supuestos fundamentales.....	7
1.7 Pre-procesamiento de la información	8
1.7.1 Seleccionar las unidades de planificación.....	8
1.7.2 Determinar la distribución de los objetos de conservación	9
2. Comienzo.....	11
2.1 Requerimientos del sistema.....	11
2.2 Instalación de Software.....	11
2.3 Apoyo al Freeware	12
2.3.1 CLUZ (por sus siglas en inglés) (Zonación del Uso de la Tierra para la Conservación)	12
2.3.2 P.A.N.D.A. (por sus siglas en inglés) Aplicación para el Diseño de Redes de Áreas Protegidas.....	12
2.3.3 C-Plan.....	12
2.4 Condiciones requeridas para ejecutar Marxan	13
3. Archivos de Entrada, Parámetros y Variables.....	15
3.1 Introducción.....	15
3.1.1 Tipos de Archivos de Entrada	16
3.1.2 Manejo de los Archivos de Entrada.....	17
3.2 Archivos requeridos	18
3.2.1 El Archivo de Parámetros de Entrada	18
3.2.1.1 Problema	24
3.2.1.1.1 Ejecuciones Repetidas	24
3.2.1.1.2 Modificador de Longitud de Frontera.....	25
3.2.1.1.3 Tipo de archivo de entrada	27
3.2.1.2 Opciones de ejecución.....	28
3.2.1.2.1 Opciones de ejecución	28
3.2.1.2.2 Mejoramiento Iterativo	30
3.2.1.2.3 Heurística.....	30
3.2.1.3 Templado	31

3.2.1.3.1 Cantidad de Iteraciones, Disminución de las Temperaturas, Temperatura Inicial y Factor de Enfriamiento	31
3.2.1.4 Entrada	32
3.2.1.4.1 Nombre del Archivo de Especies, Unidades de Planificación, Unidades de Planificación versus Especies. Definición de Bloques, Longitud de Fronteras y Carpeta de Entrada	32
3.2.1.5 Salidas	33
3.2.1.5.1 Salidas en Pantalla	33
3.2.1.5.2 Salva de archivos y nombres de archivos	34
3.2.1.5.3 Directorio de Salida	35
3.2.1.5.4 Proporción no alcanzada de las especies	35
3.2.1.6 Umbral de Costo	36
3.2.1.6.1 Umbral, Factor de Penalidad A y Factor de Penalidad B	36
3.2.1.7 Misceláneas	37
3.2.1.7.1 Iniciar Prop	37
3.2.1.7.2 Semilla aleatoria	38
3.2.1.7.3 Regla de Agrupamiento	38
3.2.1.7.4 Agilización del mejor resultado	39
3.2.2 El Archivo de Objetos de Conservación	40
3.2.2.1 ID del Objeto de Conservación	41
3.2.2.2 Tipo de Objeto de Conservación	41
3.2.2.3 Meta Cuantitativa de Representación del Objeto	42
3.2.2.4 Factor de Penalidad del Objeto de Conservación	43
3.2.2.5 Tamaño Mínimo de Agrupamiento	44
3.2.2.6 Metas Cuantitativas para las Incidencias de Objetos de Conservación	46
3.2.2.7 Nombre del Objeto de Conservación	46
3.2.2.8 Meta Cuantitativa para Incidencias de Objetos Aislados	46
3.2.2.9 Distancia Mínima de Separación	47
3.2.3 El Archivo de Unidades de Planificación	48
3.2.3.1 Identificación de la Unidad de Planificación	49
3.2.3.2 Costo de las Unidades de Planificación	49
3.2.3.3 Condición de la Unidad de Planificación	50
3.2.3.4 Ubicación de la Unidad de Planificación en el eje X	52
3.2.3.5 Ubicación de la Unidad de Planificación en el eje Y	52
3.2.4 El Archivo de Unidades de Planificación versus Objetos de Conservación	53
3.2.4.1 Formato Vertical	53
3.2.4.1.1 ID del Objeto de conservación	54
3.2.4.1.2 ID de la Unidad de Planificación	54
3.2.4.1.3 Cantidad del Objeto de Conservación	54
3.2.4.2 Formato Horizontal	55
3.3 Archivos Opcionales	56
3.3.1 El Archivo de Longitud de Frontera	56
3.3.1.1 ID de las Unidades de Planificación	56
3.3.1.2 Longitud de Frontera	57
3.3.2 El Archivo de Definición de Bloques	58
3.3.2.1 Tipo de Objeto de Conservación	60
3.3.2.2. Meta Cuantitativa de Proporción para la Representación del Objeto	60
3.3.2.3 Todas las variables restantes	61
4. Ejecución del Software	63
5. Resultados	65
5.1 Manejo del Archivo de Salida	65
5.2 Resultados en pantalla	66

5.2.1 Resultados Básicos	66
5.2.1.1 Run (Corrida o ejecución)	67
5.2.1.2 Value (Valor)	67
5.2.1.3 Cost (Costo)	67
5.2.1.4 PUs (unidades de planificación)	67
5.2.1.5 Boundary (Frontera)	67
5.2.1.6 Missing (No alcanzado).....	67
5.2.1.7 Shortfall (Déficit).....	68
5.2.1.8 Penalty (Penalidad).....	68
5.2.2 Progreso General	69
5.2.3 Progreso Detallado.....	70
5.3 Archivos de Salida	70
5.3.1 Formato de los Archivos de Salida.....	71
5.3.2 Soluciones para cada ejecución.....	71
5.3.3 Mejor solución de todas las ejecuciones.....	72
5.3.4 Valores no alcanzados para cada ejecución.....	72
5.3.4.1 Conservation Feature (Objeto de Conservación)	73
5.3.4.2 Feature Name (Nombre del Objeto)	73
5.3.4.3 Target (Meta Cuantitativa)	73
5.3.4.4 Amount Held (Cantidad Incluida)	73
5.3.4.5 Occurrence Target (Meta Cuantitativa de Incidencia)	73
5.3.4.6 Occurrence Held (Incidencias Incluidas)	73
5.3.4.7 Separation Target (Meta Cuantitativa de Separación)	73
5.3.4.8 Separation Achieved (Separación Alcanzada)	73
5.3.4.9 Target Met (Meta Cuantitativa Cumplida)	74
5.3.5 Información sobre el valor no alcanzado en la mejor ejecución	74
5.3.6 Información de Resumen	74
5.3.7 Detalles de Escenario.....	75
5.3.8 Solución Sumada	76
5.3.9 Archivo de Registro en Pantalla	77
5.3.10 Archivo de muestra instantánea.....	77
6. Obtención de Buenos Resultados	79
6.1 Experimentación	79
6.2 Inspección Visual	79
6.3 Análisis de sensibilidad	80
6.4 Hágase un experto.....	81
Glosario	83
Referencias Principales.....	89
Apéndice A – Solución de problemas	99
A-1. Marxan se detiene porque no se ha encontrado algún archivo o parámetro de entrada	99
A-2. Marxan se detiene a consecuencia de un identificador no reconocido	102
A-3. Marxan comienza la primera corrida pero se detiene porque no puede salvar los resultados requeridos.....	103
A-4. Marxan se ejecuta pero advierte que no puede localizar una variable específica	104
A-5. No se están guardando los resultados en el directorio de salida	105
A-6. Marxan muestra una falla cuando se ejecuta y la pantalla de Marxan se cierra.....	105

Apéndice B – Información Técnica de Marxan.....	107
B-1. La Función Objetivo	107
B-1.1 Costo	107
B-1.2 Frontera y Modificador de Longitud de Frontera (BLM).....	107
B-1.3 Penalidad y Factor de Penalidad de Especies (SPF)	108
B-1.3.1 Penalidades de configuración espacial	111
B-1.4 Penalidad de Umbral de Costo	113
B-2 Métodos de Optimización.....	115
B-2.1 Templado Simulado	115
B-2.1.1 Templado de Esquema Adaptable	116
B-2.1.2 Templado de esquema fijo.....	117
B-2.1.3 Configurar un esquema fijo de templado.....	117
B-2.2 Mejoramiento iterativo.....	120
B-2.3 Otros Algoritmos Heurísticos.....	121
B-2.3.1 El heurístico 'Ávido'.....	122
B-2.3.1.1 Riqueza.....	123
B-2.3.1.2 Ávido Puro	123
B-2.3.2 Algoritmo de Rareza	123
B-2.3.2.1 Rareza Máxima.....	124
B-2.3.2.2 Mejor Rareza	125
B-2.3.2.3 Rareza Sumada.....	125
B-2.3.2.4 Rareza Promedio.....	125
B-2.3.3 Irreemplazabilidad.....	126
B-2.3.3.1 Producto	126
B-2.3.3.2 La Irreemplazabilidad Sumada.....	126

Apéndice C – Asesoría sobre el desarrollo de archivos de entrada de Marxan y muestra de los resultados en SIG	127
C-1 Recursos	127
C-1.1 Software	127
C-1.2 Cursos y Guías Didácticas.....	128
C-2 Crear el archivo de unidad de planificación	129
Para usuarios de ArcView 3:	130
Para usuarios de ArcMap 8 y 9:	131
C-3 Crear el archivo de unidad de planificación versus objeto de conservación.....	133
Para Usuarios de ArcView 3.....	134
Para usuarios de ArcMap	134
C-4 Archivo de Objeto de Conservación.....	135
C-5 Crear el Archivo de Longitud de Frontera	136
C-6 Conectar Archivos de Salida con ArcGIS	137

Listar a Tablas:

Tabla 1: Archivos de entrada de Marxan y nombres predeterminados	15
Tabla 2: Nombres de Marxan y valores predeterminados	22
Tabla 3: Valores de Unidad de Planificación.....	50
Tabla 4: Tipos y nombres de archivos de salida.....	70

1. Introducción

1.1 ¿Qué es Marxan?

Marxan es un software que propicia ayuda en la toma de decisiones para el diseño de sistemas de reservas¹. La idea básica de un problema de diseño de reservas es que un planificador de la conservación cuenta con gran número de sitios potenciales (o unidades de planificación) a partir de los cuales seleccionar nuevas áreas de conservación. Sería deseable concebir un sistema de reservas constituido por una selección de esas unidades de planificación, que dé solución a un problema que incluye diferentes principios y criterios ecológicos, sociales y económicos. Marxan está diseñado, en primer lugar, para solucionar un tipo particular de problema de diseño de reservas, conocido como ‘el problema de conjunto mínimo’, donde el objetivo es alcanzar una representación mínima de determinados rasgos de la biodiversidad con el menor costo posible. (McDonnell et al. 2002). Esto se basa en el hecho de que hay más posibilidades de implementar aquellas redes de reserva menos costosas o que afectan menos a la sociedad. Asimismo, al alcanzar un conjunto de metas cuantitativas para todos los objetos de conservación, se proporciona una plataforma sólida propicia para la futura expansión de un sistema de reservas; con frecuencia, resulta difícil expandir los sistemas de reservas sesgados hacia habitats de poco valor comercial. En los problemas de conjunto mínimo, los elementos de la biodiversidad que se desean conservar se introducen como limitaciones a las soluciones del problema (Possingham et al 2000). Contando con una información razonablemente abarcadora referente a las especies, los habitats u otros objetos relevantes de la biodiversidad, Marxan se propone identificar el sistema de reservas (una combinación de unidades de planificación) que cumplirá, con un costo mínimo, las metas cuantitativas de biodiversidad² definidas por el usuario (Ball y Possingham 2000; Possingham et al 2000).

Como ejemplo, una posible meta cuantitativa de la biodiversidad pudiera ser garantizar que al menos se represente en una red de áreas protegidas, el 30 % de la abundancia de cada tipo de vegetación. Si dicha protección sólo es posible mediante la compra de terrenos, es entonces probable que un planificador (y los políticos), deseen contar con un sistema de reservas que minimice el costo monetario total de la compra del terreno necesario para cumplir con las metas de

¹ Aunque puede utilizarse también con otros propósitos, como se explica en la sección 1.3.

² Empleamos el término ‘meta cuantitativa’ como traducción del término inglés “target”, pues los autores se refieren con él no a las especies o los objetos presentes en una región de planificación, sino a la representación deseada de éstas en un sistema de reservas

conservación.(Carwardine et al. 2006). En los casos donde el costo real del terreno no esté disponible, el área de reservas pudiera ser utilizada como sustituto del costo, basado en la suposición de que mientras más extenso sea el sistema de reservas, más costoso resultará implementarlo y manejarlo (aunque éste no sea siempre el caso). El costo utilizado en Marxan puede ser también una medida relativa de costos, ya sean de carácter social, económico o ecológico, o una combinación de estas.

La cantidad de posibles soluciones, incluso para un pequeño problema de selección, es muy grande (sólo para una pequeña cantidad de 200 unidades de planificación se obtienen más de 1.6×10^{60} formas posibles de configurar un sistema de reservas). Debido a que encontrar la mejor solución a este tipo de problema resulta un proceso complejo y que demanda mucho tiempo de trabajo, se han desarrollado algoritmos de computación para resolverlo. Un algoritmo es un proceso matemático o una serie de reglas para la solución de problemas. Se han concebido dos tipos generales de herramientas para solucionar los problemas de “diseño de reservas”: algoritmos exactos y algoritmos heurísticos (no exactos). Aunque los algoritmos exactos pueden identificar la solución óptima única, resulta difícil y muchas veces imposible, encontrar la solución única a problemas de diseño de sistemas de reservas extensos, en un periodo de tiempo razonable (Possingham et al. 2000; Cabeza 2003). Los algoritmos heurísticos, por otra parte, brindan gran variedad de soluciones buenas, casi óptimas, que no sólo proporcionan a los planificadores y las partes interesadas (stakeholders) una serie de opciones a tomar en consideración, sino que además, pueden ser generadas con gran rapidez. (Possingham et al. 2000; McDonnell et al. 2002; Cabeza 2003). Como resultado, por lo general se prefieren los algoritmos heurísticos a los exactos. Marxan posibilita encontrar, en poco tiempo, una gama de soluciones casi óptimas (incluso para problemas de planificación realmente grandes), utilizando un algoritmo heurístico muy poderoso conocido como “templado simulado” (simulated annealing) (Apéndice B-2.1). El algoritmo de templado simulado, por lo general, se acercará mucho más a la solución óptima que otros métodos heurísticos como el llamado “Ávido” (Apéndice B-2.3.1). Si se desea, Marxan también posibilita la búsqueda de soluciones empleando diversos algoritmos menos sofisticados, pero con frecuencia más rápidos (vea Sección 3.2.1.2.1). Marxan forma parte de una cadena de herramientas para el diseño de reserva que incluye a su predecesor directo, SPEXAN.

1.1.1 Otras Versiones de Marxan

En este manual sólo se describe la versión de Marxan 1.8.10., el software clásico de Marxan. Sin embargo, existen otras variaciones de Marxan con funcionalidades modificadas que están disponibles o en fase de desarrollo por el Centro de Ecología de la Universidad de Queensland (<http://www.ecology.uq.edu.au>). Entre estas se incluyen: una versión óptima de Marxan para solucionar problemas muy extensos que incluyen a más de 20 mil unidades de planificación (se ha logrado encontrar, de manera exitosa, soluciones a problemas con medio millón de unidades de planificación); una versión que permite la información de probabilidades referente a las amenazas o a la presencia de objetos de conservación en los sitios que serán incluidos en el problema del diseño de la reserva; y “Marxan with Zones”(Marxan con zonas), que se está desarrollando para trabajar con zonación de objetivos múltiples. Aunque la operación básica de esas versiones es la que se describe en este manual, cada una de ellas tiene sus idiosincrasias, que serán explicadas en un apéndice habilitado para cada una de las versiones. Aunque Marxan y sus variaciones se pueden descargar de manera gratuita, su desarrollo siempre ha dependido parcialmente de los organismos de financiamiento externo, que se han ofrecido a apoyar este trabajo.

1.2 Planificación Sistemática de la Conservación

La planificación sistemática de la conservación es ampliamente considerada como ‘mejor práctica’ en la conservación, ya que facilita un proceso transparente, integral y bien argumentado en la toma de decisiones. La transparencia alude a como las personas llegan a comprender los procedimientos para la toma de decisiones y los resultados. Un proceso de planificación, que se desarrolle con total transparencia, tiende a incrementar la responsabilidad y la credibilidad en la planificación de la conservación y en la toma de decisiones. Los procesos de planificación inclusivos están destinados a incorporar la información y los valores, que brindan las partes interesadas, de manera que se reduzcan los conflictos de intereses.

Recíprocamente, esto deviene en decisiones más sólidas y de mayor aceptación. La buena argumentación se deriva de la capacidad de los planes sistemáticos para analizar, de manera explícita, con qué exactitud una selección específica de reservas cumple sus metas cuantitativas, así como la validez del razonamiento utilizado para lograrlo. En el MGPH se hace un análisis detallado de cada uno de esos principios. Aunque Marxan puede ser utilizado con diferentes objetivos y en diferentes etapas del proceso de planificación sistemática de la conservación, fue diseñado principalmente para ayudar en la selección de nuevas áreas de conservación con un

“costo” mínimo, así como facilitar el estudio del balance existente entre la conservación y los objetivos socio-económicos. Marxan puede contribuir a establecer prioridades para las acciones de conservación, destacando aquellos lugares que pudieran ser inclusiones importantes dentro de una red de reservas efectiva. Marxan también puede ser empleado como una herramienta para evaluar la representación y la integridad de las redes de reservas existentes (Stewart et al. 2003).

Es importante comprender, que la función real de Marxan, junto a otros software de apoyo a las decisiones, es apoyar el proceso de la toma de decisiones. Las soluciones ofrecidas por Marxan pueden constituir la base del debate de un plan definitivo que incorpora factores adicionales ya sean políticos, socio-económicos o pragmáticos. Algunas de las limitaciones que se encuentran al utilizar Marxan se explican posteriormente en la Sección 1.4

1.3 Interrogantes que Marxan puede ayudar a resolver

La función principal de Marxan es facilitar la selección eficiente de subconjuntos a partir de un conjunto grande de objetos permanentes en el espacio, que son representados en un mapa. Aunque Marxan fue diseñado en sus orígenes para garantizar la representación de las especies y los ecosistemas en la planificación de la conservación de la biodiversidad, y se ha aplicado fundamentalmente a este campo, ha quedado demostrado que es aplicable a una amplia gama de actividades de planificación. Marxan puede ayudar fundamentalmente en la solución de problemas relacionados con la selección espacial explícita de ‘conjuntos mínimos’. Por ejemplo, se ha utilizado para identificar un conjunto eficiente, desde el punto de vista espacial, de “zonas de pesca” (Ban, Comunicación Personal); mientras que en el campo del manejo de los recursos naturales marinos y de zonas costeras, Marxan se ha utilizado para apoyar planes de zonación de uso múltiple que equilibran los diversos intereses de la pesca, la transportación y la conservación entre otros. Por ejemplo. Fernandes et al. 2005). Chan et al. (2006) han explorado el uso de Marxan para lograr tanto metas deservicios de ecosistema, como de biodiversidad. Algunas de estas aplicaciones requerirán de un uso más creativo de Marxan y no contamos con espacio suficiente para ofrecer una guía en este sentido, Sugerimos que una vez que se familiarice con las operaciones básicas de Marxan, usted busque algunos de los diversos ejemplos de operaciones con Marxan que han sido publicados, y consulte el MGPH.

1.4 Limitaciones de Marxan

Las limitaciones técnicas de Marxan se harán evidentes a medida que se avance en la lectura del manual, y en muchos casos, se pueden reducir mediante la manipulación de los datos y de los escenarios (vea ejemplos en el MGPH). Sin embargo, no tan obvias pero sí más importantes, son las limitaciones filosóficas del software de diseño de reservas. Estas deben ser comprendidas. Marxan opera como parte de un proceso de planificación y no está diseñado para funcionar como una solución independiente para el diseño de reservas. Su efectividad depende de la participación de las personas, de la adopción de importantes principios ecológicos, el establecimiento de objetivos y metas de conservación bien argumentadas desde el punto de vista científico, así como el desarrollo y la inclusión de información espacial de calidad. Marxan debe ser utilizado como parte de un proceso de planificación sistemática de la conservación y en colaboración con otras formas de conocimiento. Esas otras formas de conocimiento son de vital importancia para perfeccionar los aportes hechos por Marxan, la interpretación de los resultados obtenidos y el mejoramiento de las fronteras definitivas de las zonas de conservación.

1.5 La función objetivo

Para que Marxan encuentre buenas soluciones a un problema debe contar con una base para comparar soluciones alternativas (es decir, conjuntos de unidades de planificación) y por consiguiente, identificar las mejores. Esto se logra mediante el uso de una función matemática objetivo que calcula un valor para varias unidades de planificación, basándose en los diferentes costos que puede tener el conjunto seleccionado y en las penalidades por no cumplir con las metas cuantitativas de conservación u otras metas. De esta manera, una solución donde la cantidad de unidades de planificación sea cero, a pesar de que su implementación sería muy poco costosa, no cumplirá ninguna meta de conservación de la biodiversidad y por consiguiente, el valor de la función objetivo, deberá ser muy bajo. Poder contar con una función objetivo que asigna un valor a un determinado sistema de reservas, nos permite automatizar la selección de buenas redes de reservas (al menos, según la función objetivo). Marxan no hace más que comprobar de manera continua las selecciones alternativas de unidades de planificación, con el propósito de mejorar el valor de todo el sistema de reservas en conjunto.

El valor de la función objetivo debe, por supuesto, reflejar la conveniencia de ese sistema de reservas específico. En su forma más simple, la función objetivo de Marxan es una combinación del costo total del sistema de reservas y la penalidad por aquellas metas ecológicas que no se cumplan. La función objetivo está diseñada de

forma que mientras más bajo sea su valor, mejor sea el resultado. Asimismo, Marxan permite que se tenga en cuenta una medida de la fragmentación del sistema de reservas, de modo que, por lo general, se va a desear contar un sistema de reservas que no esté muy fragmentado. Un sistema de reservas fragmentado no solo conducirá, con frecuencia, a una fragmentación no deseada de las comunidades ecológicas, sino que es probable que el manejo y el cumplimiento se tornen más difíciles y costosos. Una red de reservas más fragmentada tendrá una mayor longitud total de frontera. Esa longitud de frontera, adicionando una mayor importancia en comparación con otros componentes del objetivo (costo y objetivos a cumplir), puede ser incluida en la función objetivo. La posible adición final a la función objetivo constituye una penalidad por exceder algunos costos pre- establecidos³. Aunque Marxan siempre intenta encontrar la red de reservas efectiva menos costosa, en ocasiones (o con frecuencia) pudiera encontrar restricciones fiscales invariables para las acciones de conservación. En esos casos, deseamos garantizar las mejores soluciones teniendo en cuenta que se utilice sólo el presupuesto disponible.

Así, la función objetivo en Marxan toma la siguiente forma:

$$\sum_{PUs} Cost + BLM \sum_{PUs} Boundary + \sum_{Con Value} SPF \times Penalty + CostThresholdPenalty(t)$$

1. El costo total de la red de reservas (requerido)
2. La penalidad por no representar adecuadamente los objetos de conservación (requerido)
3. La longitud total de frontera de la reserva, multiplicada por un modificador (opcional)
4. La penalidad por exceder un umbral de costo preprogramado (opcional- vea nota al pie 3)

Los términos uno y tres pueden analizarse como ‘costos’, mientras que los términos dos y cuatro son penalidades por incumplir diferentes criterios. Por lo general, no aconsejamos utilizar la penalidad de umbral de costo. Información más detallada referente a la función objetivo y al cálculo de cada uno de los diferentes términos puede encontrarla en el Apéndice B-1. La Sección 3 de este manual contiene detalles sobre como controlar cuáles objetos contribuyen a la función objetivo y cual será la dimensión de las penalidades.

³ Debido a resultados inconsistentes que se obtienen en ocasiones, la Penalidad de Umbral de Costo de Marxan está siendo reprogramada en estos momentos. Los usuarios de Marxan 1.8.10 deben ser conscientes de esto. Se recomienda que dicha función se ejecute cuidadosamente.

1.6 Supuestos fundamentales

El uso de una herramienta como Marxan para la selección automatizada de reservas está respaldado por supuestos fundamentales. A pesar de ser una herramienta muy eficiente para solucionar problemas difíciles de selección de sitios, hay pequeños detalles, como el conocimiento de la calidad de la información, que no siempre se pueden incorporar, por tanto, el empleo de Marxan debe basarse siempre en determinados supuestos. Quizás el supuesto más difícil de lograr y por tanto el que se viola con más frecuencia, es que se asume que la distribución espacial de la información utilizada en un análisis hecho con Marxan es consistente. Ello no significa que en todos los lugares encontramos los mismos objetos, sino que la información se recopiló de forma tal, que los mismos objetos de conservación pudieran ser encontrados en todos los lugares si existieran, es decir, la información no está sesgada espacialmente. Por ejemplo, si utilizamos datos referentes a la incidencia* de las especies para seleccionar las reservas, es muy probable que la detección de las especies no haya sido uniforme en toda la región de planificación. Las recopilaciones u observaciones pueden haber resultado más intensas cerca de las estaciones de investigación, zonas pobladas o lugares de fácil acceso como los caminos cercanos. Ello será interpretado por Marxan como un reflejo fiel de la distribución total de las especies y, por consiguiente, va a dirigir las soluciones de reserva a esas zonas que han sido estudiadas a fondo. Esto pudiera tener una relación sustancial con la forma final de todo el sistema de reservas, especialmente, si se hace énfasis en lo compacto del sistema. Una forma de disminuir un poco esta parcialidad es modelar la distribución probable de las especies o hábitat basándose en información biofísica. Otra manera adicional, y quizás más simple, de disminuir las parcialidades provocadas por la intensidad del muestreo, es emplear otras mediciones sustitutas como el tipo de hábitat o incluso variables físicas para representar la distribución de la biodiversidad que queremos conservar. En algunos casos, no obstante, sería muy irresponsable no tomar en cuenta las incidencias de objetos de conservación importantes, como pueden ser especies muy amenazadas. En la Sección 3.2.3.3 se describe un método para lidiar con situaciones como esa. Cuando se cree que la información pueda ser parcial en términos de espacio, dicha parcialidad debe ser documentada.

Marxan no considera que la información sea inexacta⁴. Asume que todas las representaciones de objetos son certeras, y que las incidencias de un objeto son de igual valor. En realidad, un planificador de la conservación pudiera estar muy seguro

* Se utiliza en el texto la palabra “*incidencia*” como traducción del término inglés “*ocurrence*”. N.d.T.

⁴ Versiones de Marxan en fase de desarrollo son capaces de lidiar con niveles de inexactitud.

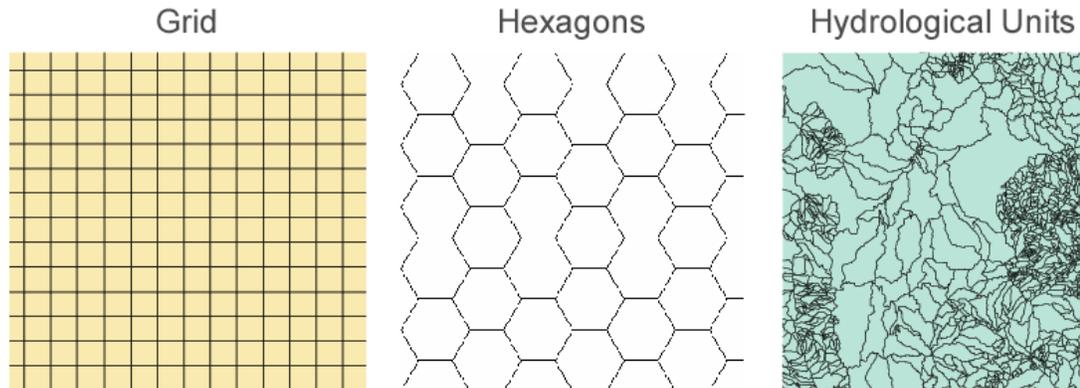
sobre la presencia de cierto objeto de conservación en algunos sitios de su distribución y menos seguro en otros casos. Detalles como ese, requieren de una valoración cuidadosa de los resultados de Marxan para garantizar que realmente expresen los objetos de conservación deseados. Siempre recuerde el cliché que indica que la calidad de los resultados que se obtienen con Marxan no puede ser mayor que la calidad de la información original o modelada que se introdujo. El MGPB sugiere algunos métodos para desarrollar un análisis profundo utilizando conjuntos de información más débiles.

1.7 Pre-procesamiento de la información

De hecho, ejecutar Marxan para generar soluciones de reservas será, por lo general, la fase de rápida ejecución de un ejercicio de planificación de la conservación. Primeramente, se deben completar un número de pasos que, con frecuencia, demandan tiempo de trabajo.

1.7.1 Seleccionar las unidades de planificación

Un paso esencial del Pre-procesamiento es dividir la región de planificación en un conjunto de unidades de planificación. En el apéndice C-2 se brinda asesoramiento sobre algunos métodos utilizados para crear las unidades de planificación. En su forma más simple, las unidades de planificación pueden ser definidas cubriendo la región de planificación con una red de cuadrículas o un entramado de hexágonos. Ellos deben abarcar todas las zonas que pudieran ser seleccionadas como parte del sistema de reservas, y su tamaño debe ser de una escala apropiada, considerando tanto las características ecológicas que se desean captar como el tamaño de las zonas protegidas que se espera sean establecidas. En general, su resolución no debe ser más fina que la de los datos sobre los objetos de conservación, ni tan gruesa que no facilite las decisiones de manejo. Sin embargo, no es necesario que las unidades de planificación sean uniformes. Ni es siempre cierto que unidades de planificación más pequeñas son mejores. En algunos casos, tendría más sentido tener unidades de planificación que informan sobre divisiones ecológicas naturales, tales como las unidades hidrológicas, o incluso divisiones político- gubernamentales como las parcelas catastrales. Una unidad de planificación uniforme arrojará resultados más útiles para otros usos.



Tres tipos posibles de unidades de planificación que se pueden utilizar con Marxan.

Existe un límite en la cantidad de unidades de planificación con las que Marxan puede operar. No obstante, no es una cifra fija ya que ello depende también de la cantidad de objetos de conservación que se desea planificar e incluso, en alguna medida, de la potencia de su computadora. Desafortunadamente, no conocemos ninguna regla general efectiva para evaluar esa cifra, se han ejecutado análisis con Marxan, de manera bastante exitosa, con 10 mil unidades de planificación y 100 objetos de conservación. Análisis de grandes volúmenes (ej, > 20 mil unidades de planificación) deben realizarse ejecutando la versión optimizada de Marxan (2.0.2.) que está también disponible en el sitio web del Centro de Ecología (vea Sección 1.1.1) La cantidad de unidades de planificación y objetos con las que puede operar esta versión se ve limitada únicamente en dependencia de la memoria disponible.

Se debe tener extremo cuidado al escoger las unidades de planificación adecuadas, ya que ellas influirán en los resultados del análisis hecho con Marxan. En la sección de Referencias Principales se hace alusión a este tema. Aunque rara vez se hace, no hay razón alguna por la que no se puedan ejecutar dos análisis que utilizan diferentes unidades de planificación.

1.7.2 Determinar la distribución de los objetos de conservación

Un segundo paso importante que debe darse antes de emplear Marxan es determinar la distribución de los objetos de conservación en las unidades de planificación. Esto significa recopilar toda la información necesaria acerca de los objetos de conservación y luego calcular cuánto (área, número de incidencias, etcétera) de cada uno de ellos hay en cada unidad de planificación. Para hacer esto, generalmente se necesitará tener algún conocimiento sobre sistemas de información geográfica (SIG); en el apéndice C se brinda una explicación sobre una manera de hacerlo. En la mayoría de los casos, recopilar la información necesaria y calcular la

representación de los objetos de conservación en las unidad de planificación implica un esfuerzo mayor que la ejecución de Marxan. Los administradores de proyectos deben tener mucho cuidado de destinar tiempo suficiente para este paso.

2. Comienzo

2.1 Requerimientos del sistema

Marxan no exige grandes requerimientos de sistema para ejecutarse . Bastará con cualquier sistema operativo de Microsoft, incluso uno antiguo. Como regla general, si la computadora tiene potencia suficiente para ejecutar un software comercial de SIG, será suficiente para ejecutar Marxan. Mientras más unidades de planificación, más objetos de conservación y más configuraciones opcionales avanzadas se tengan, más lenta resultará la ejecución de Marxan. Por supuesto, mientras más potente sea su computadora (MHz y RAM), con mayor rapidez se ejecutará Marxan. En dependencia de esos factores, el tiempo requerido para que Marxan obtenga cien buenas soluciones a su problema puede variar de algunos minutos a varios días.⁵

2.2 Instalación de Software

Es posible que si está leyendo este manual ya haya realizado la descarga de Marxan. De lo contrario, Marxan puede descargarse desde <http://www.ecology.uq.edu.au/marxan.htm>. Se necesitará cerca de 2 MB de espacio libre en disco para instalar Marxan y los archivos asociados.

Cuando realice la descarga de Marxan recibirá los archivos siguientes:

1. Marxan.exe (el programa ejecutable de Marxan)
2. Inedit.exe (un programa que permite generar fácilmente el Archivo de Entrada de Parámetro -el archivo que controla el funcionamiento de Marxan)
3. input.dat (un ejemplo del Archivo de Entrada de Parámetros)
4. Una carpeta denominada 'Muestra', que contiene ejemplos de otros archivos de entrada utilizados para ejecutar Marxan (los detalles de este archivo se explican más adelante).
5. Este manual

Esos archivos pueden ser salvados en cualquier lugar de la computadora. En aras de hacerlo más simple, al ejecutar Marxan, el ejecutable, 'Marxan.exe', debe localizarse en la misma carpeta de los archivos de entrada para el proyecto (vea Sección 3.1.2). En lugar de mover los archivos constantemente, recomendamos copiar el ejecutable de Marxan a cada carpeta que contenga un proyecto de Marxan.

⁵ Usualmente son las características avanzadas de Marxan (como la distancia de separación y el tamaño de agrupamiento mínimo) las que pueden hacer el proceso notablemente más lento, en especial, con grandes cifras de unidades de planificación. Recomendamos, por lo tanto, que las ejecuciones iniciales de prueba no hagan uso de esas características avanzadas para que la operabilidad básica de la entrada de Marxan, sea primeramente verificada y comprobada

2.3 Apoyo al Freeware

En este manual se describe como ejecutar Marxan como un programa independiente, sin embargo, existen varias interfaces, disponibles gratuitamente, que pueden ayudar en la ejecución del programa. Muchos usuarios consideran que esas interfaces han resultado ser de gran ayuda para la generación de archivos de entrada adecuados y para mostrar los resultados de Marxan. Guías para el uso de esos programas (descritos más adelante) pueden encontrarse en sus respectivos sitios web o manuales de usuario.

2.3.1 CLUZ (por sus siglas en inglés) (Zonación del Uso de la Tierra para la Conservación)

CLUZ es una interfase de ArcView GIS que se enlaza con Marxan. Fue desarrollado por Bob Smith en el Instituto para la Conservación y Ecología de Durrell y se encuentra disponible en <http://www.mosaic-conservation.org/cluz/>. CLUZ nos provee de una conexión dinámica con Marxan, de forma que el usuario pueda ejecutar el programa sin dificultad y transfiera a un mapa los resultados de dicha ejecución. De igual manera, contiene herramientas que ayudan a desarrollar los archivos de entrada que requiere Marxan. CLUZ ofrece útiles ejercicios de práctica para guiar a los usuarios en el desarrollo de archivos de entrada, la modificación de los parámetros de corrida y la exposición de los resultados de Marxan.

2.3.2 P.A.N.D.A. (por sus siglas en inglés) Aplicación para el Diseño de Redes de Áreas Protegidas

P.A.N.D.A. es una aplicación independiente que utiliza los softwares Visual Basic y ArcObjects. Fue desarrollado por Francesca Riolo para proveer a los usuarios de ArcGIS de un marco fácil de utilizar para el diseño sistemático de redes de áreas protegidas. La ejecución del programa requiere de ArcGIS. Está disponible en http://www.mappamondogis.it/panda_en.htm

2.3.3 C-Plan

C-Plan es un software de apoyo en la toma de decisiones referentes a la conservación que se enlaza con un SIG para representar en mapas las opciones con las que se cuenta para cumplir metas explícitas de conservación. Fue desarrollado por Matt Watts y Bob Pressey. C-Plan permite que los usuarios decidan qué unidades de planificación deben ser colocadas bajo alguna forma de manejo de la conservación, mediante selección manual. Permite además la selección automatizada utilizando algoritmos heurísticos. La versión más moderna, C-Plan 3.4,

tiene una interfase de Marxan que puede generar archivos de entrada de Marxan desde archivos de C-Plan. Permite también que los usuarios ejecuten Marxan desde C-Plan e importen los resultados de Marxan de vuelta a C-Plan para que sean expuestos en un SIG. C-Plan está disponible en <http://www.uq.edu.au/~uqmwatts/cplan.html>.

2.4 Condiciones requeridas para ejecutar Marxan

Hay cuatro pasos fundamentales para ejecutar Marxan:

1. Configurar los archivos de entrada
2. Configurar los parámetros de escenario
3. Ejecutar Marxan
4. Interpretar los resultados

En esta sección pretendemos brindar una guía resumida sobre los pasos esenciales a seguir para el uso de Marxan. En los capítulos posteriores se ofrece un análisis más detallado de cada uno de los pasos. El uso efectivo de Marxan nunca incluirá una aplicación excepcional ni consecutiva de esos pasos. Por el contrario, en un proyecto determinado esos pasos deben repetirse en muchas ocasiones, específicamente los últimos tres, y los resultados de cada ejecución deben ser utilizados para perfeccionar los detalles de las ejecuciones posteriores. Debido a esto, es primordial estar bien organizados y disponer de un protocolo eficiente de gestión de archivos. En las secciones posteriores se ofrecen sugerencias de posibles protocolos. Una vez que se crean los archivos de entrada, debe resultar bastante fácil modificar el escenario, re-ejecutar Marxan e investigar los resultados.

Marxan es una herramienta para el apoyo a la toma de decisiones que ayuda en la selección de sistemas eficientes de reservas, sus resultados nunca deben interpretarse como “la respuesta”. Aunque los resultados de una ejecución única de Marxan van a representar una buena solución a su problema de diseño de red de reservas, no necesariamente será la solución preferida. Marxan comprueba la utilidad de las unidades de planificación en un estilo pseudo-aleatorio (vea Apéndice B para más detalles), por consiguiente, cada ejecución puede ser ligeramente (y en ocasiones, en gran medida) diferente. En casi todos los casos gran variedad de buenas soluciones quedará disponible. Esto es una característica positiva ya que propicia la flexibilidad en la planificación y en las negociaciones con partes interesadas. Ejecutando Marxan varias veces es posible hacer un muestreo de la gama de diferentes soluciones posibles (el espacio de soluciones).

(Esta página está en blanco intencionalmente)

3. Archivos de Entrada, Parámetros y Variables

3.1 Introducción

Para ejecutar Marxan se necesita un conjunto de archivos de entrada. Esos archivos contienen toda la información con la que se desea trabajar y los detalles referentes al problema de conservación que se desea solucionar con Marxan. Se necesitan cuatro archivos de entrada, sin ellos Marxan no se podrá ejecutar. A continuación, se resumen los archivos requeridos y los opcionales;

Tabla 1: Archivos de entrada de Marxan y nombres predeterminados.

Archivo de entrada	Nombre Predeterminado	Requerido
Archivo de Parámetros de Entrada	input.dat	Si
Archivo de Objetos de Conservación ⁶	spec.dat	Si
Archivo de Unidades de Planificación	pu.dat	Si
Archivo de Unidades de Planificación versus Objetos de Conservación	puvspr2.dat	Si
Archivo de Longitud de Frontera	bound.dat	No
Archivo de Definición de Bloques	blockdef.dat	No

El **Archivo de Parámetros de Entrada** se utiliza para establecer valores para todos los parámetros principales que controlan el funcionamiento de Marxan. Se utiliza también para indicar a Marxan dónde colocar los archivos de entrada que contienen la información y otras variables, y dónde colocar los archivos de resultados.

El **Archivo de Objetos de Conservación** contiene información referente a cada uno de los objetos de conservación analizados, como sus nombres, metas cuantitativas y requerimientos de presentación, así como la penalidad que se debe aplicar en caso de que no cumplir con dichos requerimientos

El **Archivo de Unidades de Planificación** contiene información sobre las propias unidades de planificación, dígame el número de identificación, el costo, ubicación y su situación.

⁶ Conocido anteriormente como 'Archivo de las Especies '.

El **Archivo Unidades de Planificación versus Objetos de Conservación** contiene información referente a la distribución de los objetos de conservación en cada unidad de planificación.

El **Archivo de Longitud de Frontera** contiene información referente a la longitud o 'la longitud efectiva' de las fronteras compartidas entre unidades de planificación. Este archivo es necesario si se desea utilizar el Modificador de Longitud de Fronteras para mejorar la compacidad de las soluciones de reserva y, aunque no es obligatorio, sí se recomienda.

El **Archivo de Definición de Bloques** es muy similar al **Archivo de Objetos de Conservación** y se puede emplear para establecer una serie de valores predeterminados de variables para grupos de objetos de conservación.

En esta sección se describen en detalles esos seis archivos de entrada potenciales; sus funciones, los formatos y las variables que contienen. En las explicaciones contenidas en el Apéndice C encontrará más información referente a algunas maneras posibles de generar esos archivos.

Simbología para las variables



Parámetros muy importantes – hemos intentado describir cuidadosamente sus funciones



Parámetros muy técnicos – La explicación mayormente se encuentra en el Apéndice B



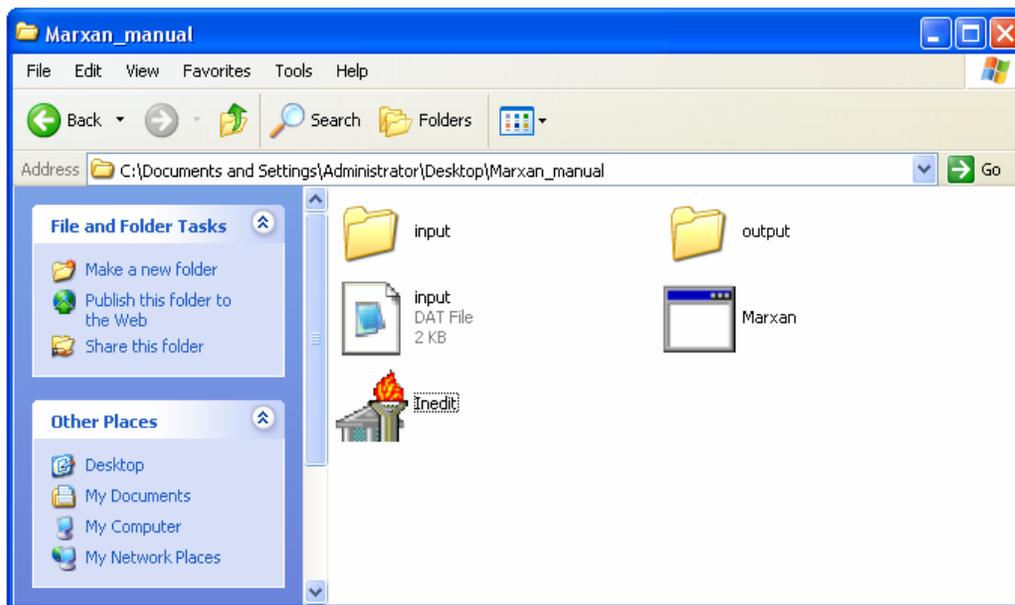
Errores comunes que pueden acarrear consecuencias serias.

3.1.1 Tipos de Archivos de Entrada

Todos los archivos de entrada de Marxan utilizan la extensión de archivo **.dat**. Esos archivos se pueden ver en programas básicos de edición de textos como Windows Notepad o TextPad. Para crear un archivo **.dat**, sólo agregue el sufijo '.dat' después del nombre cuando vaya a guardar el archivo.

3.1.2 Manejo de los Archivos de Entrada

Todos los archivos de entrada exceptuando el **Archivo de Parámetros de Entrada**, 'input.dat', deben ser almacenados en la misma carpeta. Esa carpeta por lo general se nombra, 'input' (entrada), aunque puede ser denominada con cualquier nombre que usted indique. Esta carpeta debe ser almacenada dentro de la misma carpeta que contenga el programa ejecutable de Marxan, 'Marxan.exe'. El **Archivo de Parámetros de Entrada**, 'input.dat', también debe ser almacenado en el mismo lugar que el programa ejecutable de Marxan.



Un ejemplo de cómo recomendamos que se cree una carpeta de Marxan.

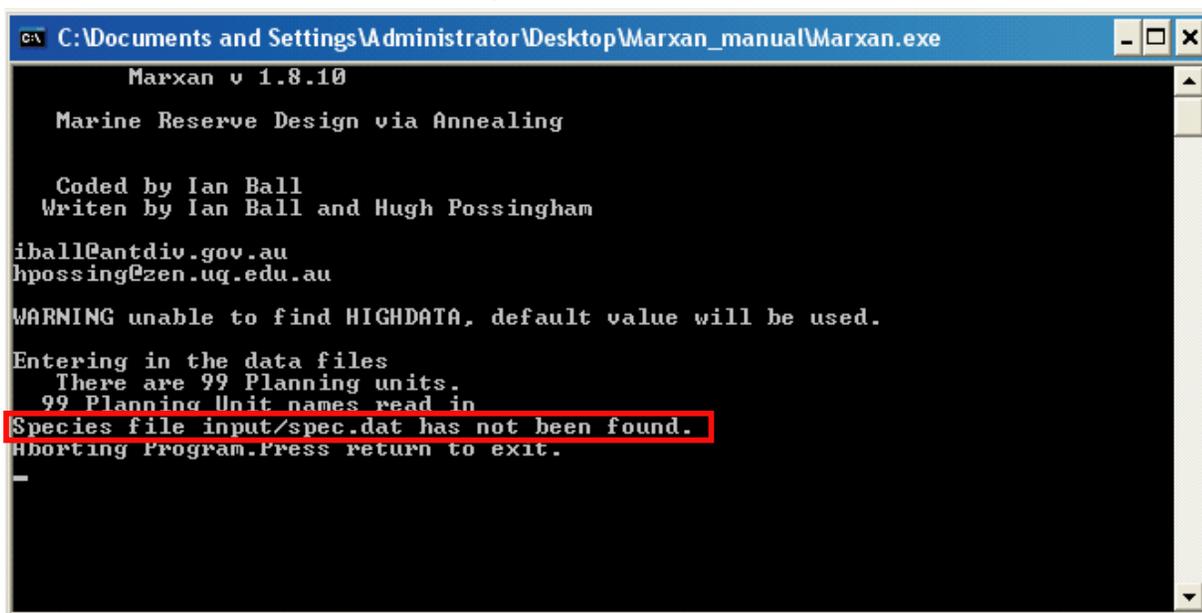
En este manual nos referimos a los diferentes archivos de entrada por sus nombres predeterminados. Usted puede nombrar a los archivos de entrada como desee, siempre que los nombres se ajusten a aquellos incluidos en el **Archivo de Parámetros de Entrada**, y siempre que se siga el procedimiento adecuado para comenzar la corrida de Marxan. Sin embargo, la utilización de nombres de archivos consistentes, ayuda a simplificar la organización de los archivos de entrada de Marxan. El **Archivo de Parámetros de Entrada**, debe ser nombrado 'input.dat', dado que ese es el nombre que Marxan va a buscar, a menos que se le ordene algo diferente con un parámetro de línea de comando.

Si usted le da otro nombre a este archivo que no sea 'input.dat' (por ejemplo, escenario 1.dat), inicie Marxan con un parámetro de línea de comando como este para hacer que reconozca el **Archivo de Parámetros de Entrada**: "Marxan.exe escenario1.dat".

Si utiliza el predeterminado 'input.dat' para nombrar el **Archivo de Parámetros de Entrada**, no es necesario utilizar el parámetro de línea de comando para que Marxan reconozca el archivo. Puede iniciarlo de la siguiente forma: "Marxan.exe".

3.2 Archivos requeridos

Si falta uno de los cuatro archivos requeridos (vea Tabla 1 en la Sección 3.1), Marxan se detendrá con un mensaje de error.



```
C:\Documents and Settings\Administrator\Desktop\Marxan_manual\Marxan.exe
Marxan v 1.8.10
Marine Reserve Design via Annealing
Coded by Ian Ball
Written by Ian Ball and Hugh Possingham
iball@antdiv.gov.au
hpossing@zen.uq.edu.au
WARNING unable to find HIGHDATA, default value will be used.
Entering in the data files
There are 99 Planning units.
99 Planning Unit names read in
Species file input/spec.dat has not been found.
Aborting Program.Press return to exit.
```

Un ejemplo del mensaje de error que muestra Marxan si no puede encontrar los archivos de entrada necesarios. En ese caso, falta el **Archivo de Objetos de Conservación** (spec.dat).

3.2.1 El Archivo de Parámetros de Entrada

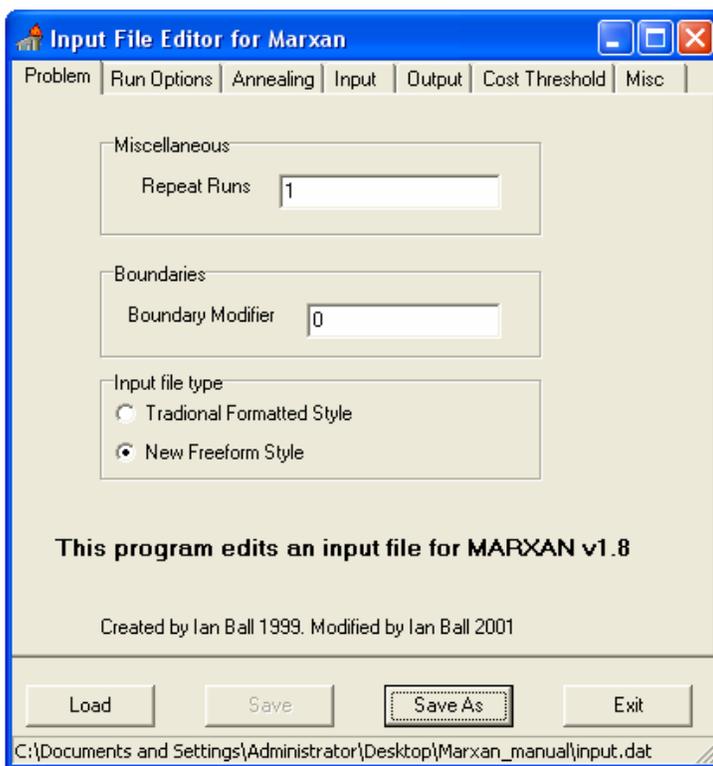
Marxan es un programa extremadamente flexible. No obstante, se deben establecer determinados parámetros para lidiar de manera adecuada con el problema específico al que se le debe dar solución. Eso se logra fundamentalmente, mediante el **Archivo de Parámetros de Entrada**. Ese archivo contiene los parámetros principales que controlan la forma en que Marxan encuentra soluciones (por ejemplo, que algoritmo (s) se utiliza (n), que parámetros contribuyen a la función objetivo). Se emplea también para indicar a Marxan donde encontrar los archivos de entrada requeridos, si se debe utilizar cualquiera de los dos archivos opcionales de entrada y qué archivos de resultados desea, así como el lugar donde deben ser ubicados.

No se preocupe si la cantidad de variables en ese archivo puede parecer un poco desalentadora, muchas de esas variables casi nunca necesitarán ser modificadas. Algunas pocas, sin embargo, deberán ser cambiadas para casi todas las ejecuciones de escenarios.

Hay dos maneras posibles de crear y modificar el **Archivo de Parámetros de Entrada**:

1. Utilizando el programa, '**Inedit**', que viene en el paquete de Marxan (recomendado)
2. Directamente mediante un editor de archivo de texto (ejemplo, Windows Notepad-para usuarios aventajados)

Al crear ese archivo por primera vez, por lo general lo mejor es utilizar **Inedit**. Ese programa ofrece una interfase gráfica de usuario que le lleva por cada uno de los parámetros que deben establecerse en el **Archivo de Parámetros de Entrada**.



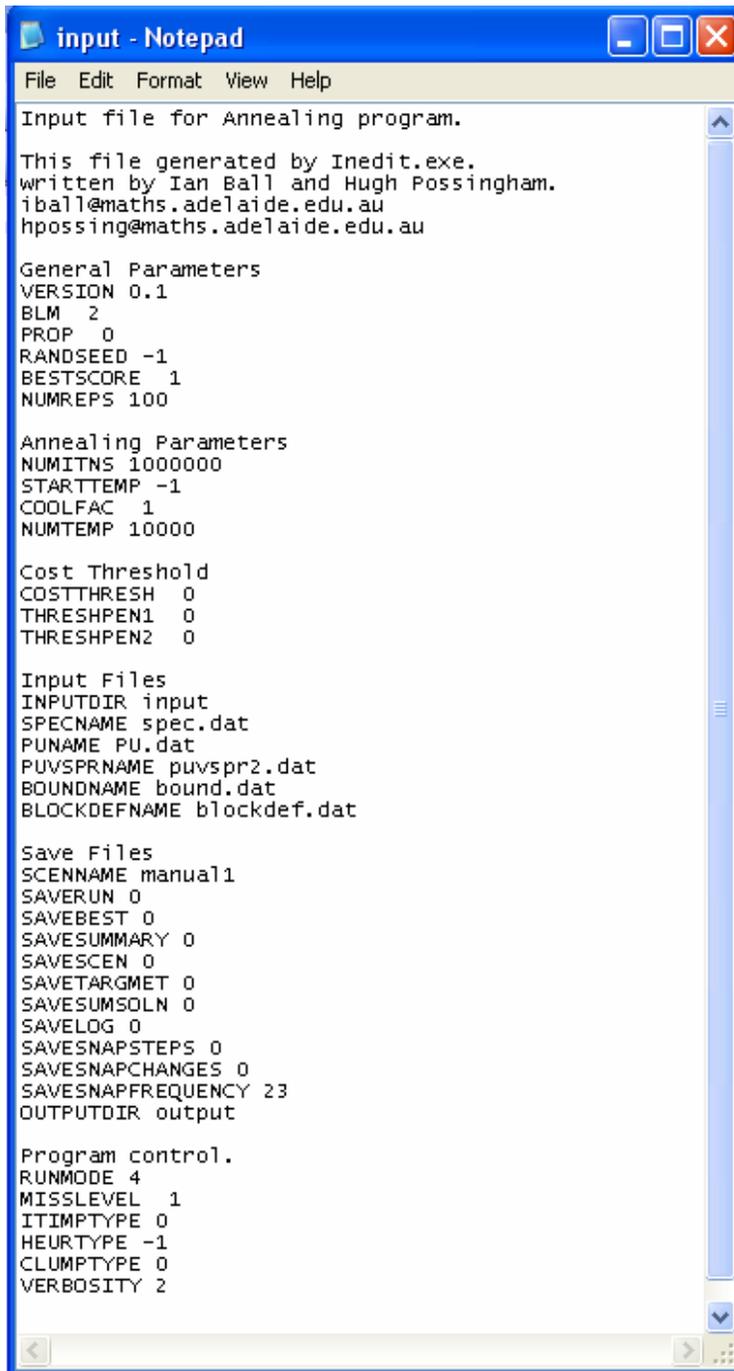
El programa **Inedit** empleado para crear el Archivo de Parámetros de Entrada (input.dat) utilizado en Marxan.

Una vez establecidos los valores para cada uno de los parámetros en **Inedit**, sólo oprima 'Save' (guardar) y automáticamente se generará el archivo correspondiente 'input.dat'. Al utilizar el comando 'Save', el archivo 'input.dat' se guardará en la misma carpeta del programa **Inedit**. Si se utiliza 'Save As' (guardar como), se podrá seleccionar donde guardar el archivo 'input.dat'.

No existen valores que puedan considerarse universalmente mejores para los parámetros incluidos en el **Archivo de Parámetros de Entrada**. Aunque valores

similares pueden funcionar bien para diferentes aplicaciones, será necesario determinar los valores más apropiados de los parámetros para cada proyecto. Eso se logra mejor empleando un estilo iterativo donde se investigan los resultados, se cambian los parámetros, se ejecuta el programa nuevamente y los nuevos resultados se comparan con los anteriores. Hay algunos métodos más cortos en ese proceso. En sub-secciones posteriores brindamos orientación sobre como determinar los valores adecuados para cada uno de los parámetros principales. Al igual que para crear el archivo, el ajuste de los parámetros se puede realizar con **Inedit** o directamente en el archivo con un editor de texto. Una vez creado por primera vez con **Inedit** y estando los usuarios familiarizados con los aspectos básicos del archivo, resulta con frecuencia más fácil realizar cambios a los parámetros directamente en el archivo 'input.dat'. Para hacer eso, solo se necesita abrir el archivo utilizando un programa editor de texto como Windows Notepad o su equivalente, cambiar los valores de parámetros correspondientes y volver a guardar el archivo. Para modificar un **Archivo de Parámetros de Entrada** existente, empleando **Inedit**, sólo cargue el archivo 'input.dat' existente utilizando el comando 'Load' (cargar), cambie los parámetros en las fichas adecuadas y vuelva a guardar el archivo.

El **Archivo de Parámetros de Entrada** se ve de la siguiente manera:



```
input - Notepad
File Edit Format View Help
Input file for Annealing program.

This file generated by Inedit.exe.
written by Ian Ball and Hugh Possingham.
iball@maths.adelaide.edu.au
hpossing@maths.adelaide.edu.au

General Parameters
VERSION 0.1
BLM 2
PROP 0
RANDSEED -1
BESTSCORE 1
NUMREPS 100

Annealing Parameters
NUMITNS 1000000
STARTTEMP -1
COOLFAC 1
NUMTEMP 10000

Cost Threshold
COSTTHRESH 0
THRESHPEN1 0
THRESHPEN2 0

Input Files
INPUTDIR input
SPECNAME spec.dat
PUNAME PU.dat
PUVSPRNAME puvspr2.dat
BOUNDNAME bound.dat
BLOCKDEFNAME blockdef.dat

Save Files
SCENNAME manual1
SAVERUN 0
SAVEBEST 0
SAVESUMMARY 0
SAVESCEN 0
SAVETARGMET 0
SAVESUMSOLN 0
SAVELOG 0
SAVESNAPSTEPS 0
SAVESNAPCHANGES 0
SAVESNAPFREQUENCY 23
OUTPUTDIR output

Program control.
RUNMODE 4
MISSLEVEL 1
ITIMPTYPE 0
HEURTYPE -1
CLUMPTYPE 0
VERBOSITY 2
```

Cada variable se expone como una sola palabra en letras mayúsculas. El valor para esa variable sigue en la misma línea con un único espacio entre el nombre y el valor de la variable. Marxan ignorará todas las líneas que no son nombres de variables válidos o no estén en letras mayúsculas, de manera que sea posible incluir comentarios o notas entre las variables de ese archivo. Las variables pueden estar en cualquier orden, sin embargo, Marxan se detendrá con un mensaje de error si alguna de ellas se define dos veces. La mayoría de las variables en el **Archivo de Parámetros de Entrada** tienen valores predeterminados que se utilizarán si la variable no se define. Las excepciones son las variables que indican a Marxan los archivos de entrada necesarios, donde guardar los archivos de resultados y la variable 'RUNMODE', que indica a Marxan qué método debe utilizar para encontrar el mejor sistema de reservas (por ejemplo, templado simulado, heurístico o ambos).

Un ejemplo del **Archivo de Entrada de Parámetros** (input.dat).

La siguiente tabla contiene una breve descripción de cada variable así como sus nombres predeterminados.

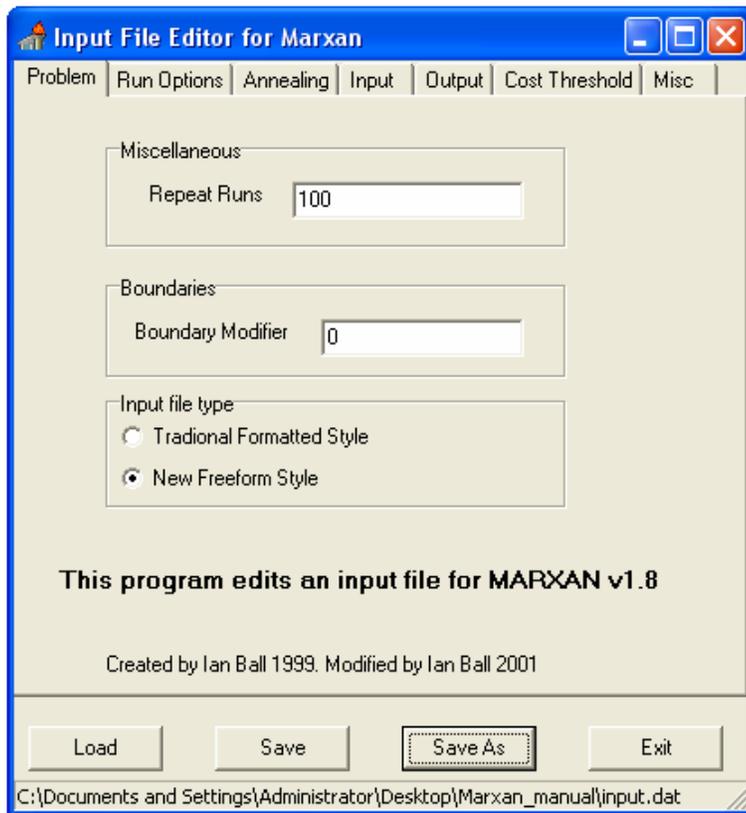
Tabla 2: Nombres de Marxan y valores predeterminados.

Nombre de la Variable	Valor Predeterminado	Descripción
VERSION	0.1	Tipo de archivo de entrada
BLM	0	Modificador de longitud de frontera
PROP	0	Proporción de las unidades de planificación en el sistema inicial de reserva
RANDSEED	-1	Número de 'semillas' aleatorias
BESTSCORE	0	Mejor resultado
NUMREPS	1	Cantidad de ejecuciones repetidas que se desean realizar
NUMITNS	0	Cantidad de iteraciones para el templado
STARTTEMP	1	Temperatura inicial para el templado
COOLFAC	0	Factor de enfriamiento para el templado
NUMTEMP	1	Cantidad de descensos de temperatura para el templado
COSTTHRESH	0	Umbral de costo
THRESHPEN1	0	Control de la cuantía de la penalidad del umbral de costo
THRESHPEN2	0	Control de la forma de la curva de penalidad del umbral de costo
INPUTDIR	<i>Definido por el usuario</i>	Nombre de la carpeta que contiene los archivos de entrada de datos
SPECNAME	spec.dat	Nombre del Archivo de Objetos de Conservación
PUNAME	pu.dat	Nombre del Archivo de Unidades de Planificación
PUVSPRNAME	puvspr2.dat	Nombre del Archivo de Unidades de Planificación versus Objetos de conservación
BOUNDNAME	bound.dat	Nombre del Archivo de Longitud de Frontera
BLOCKDEFNAME	blockdef.dat	Nombre del Archivo de Definición de Bloques
SCENNAME	Temp	Nombre del escenario para los archivos de resultados guardados
SAVERUN	0	¿Guardar cada ejecución? (0 = no)
SAVEBEST	0	¿Guardar la mejor ejecución? (0 = no)
SAVESUM	0	¿Guardar la información de resumen? (0 = no)

SAVESCEN	0	¿Guardar la información del escenario? (0 = no)
SAVETARGMET	0	¿Guardar la información de metas alcanzadas?(0 = no)
SAVESUMSOLN	0	¿Guardar la información de solución sumada? (0 = no)
SAVELOG	0	¿Guardar los archivos de registro? (0 = no)
SAVESNAPSTEPS	0	Guardar muestras instantáneas cada n pasos (0 = no)
SAVESNAPCHANGES	0	Guardar muestras instantáneas después de cada n cambios (0 = no)
SAVESNAPFREQUENCY	0	Frecuencia de las muestras instantáneas si se están utilizando
OUTPUTDIR	<i>Usuario Definido</i>	Nombre de la carpeta donde guardar archivos de resultados
RUNMODE	<i>Usuario Definido</i>	El método utilizado por Marxan para encontrar soluciones
MISSLEVEL	1	Cantidad a partir de la cual las metas se consideran no alcanzadas
ITIMPTYPE	1	Tipo de mejoramiento iterativo
HEURTYPE	1	Tipo de heurística a utilizar
CLUMPTYPE	0	Tipo de penalidad de agrupamiento
VERBOSITY	1	Cantidad de resultados mostrados en pantalla

Cada una de esas variables se describe en las sub-secciones posteriores. Hemos dividido las variables en grupos, basándonos en cuál ficha del programa **Inedit** se modifican. El nombre de la ficha es el mismo que aparece en el encabezamiento de la sub-sección.

3.2.1.1 Problema



La ficha 'Problem' (problema) de **Inedit**.

3.2.1.1.1 Ejecuciones Repetidas

Variable – 'NUMREPS'

Requerido: Si



Descripción: Indica la cantidad de repeticiones en la ejecución que se desea que Marxan realice; o dicho de otro modo, la cantidad de soluciones al problema de reservas que usted desea sean generadas por Marxan.

Cada ejecución nueva es independiente de la anterior, aunque utilizan los mismos parámetros y valores de variables. La frecuencia con que se seleccionan las unidades de planificación en ejecuciones múltiples, indica la importancia de la unidad de planificación para cumplir con eficiencia las metas cuantitativas para su sistema de reservas (vea Sección 5.3.7).

Iniciar: Al ejecutar un escenario nuevo por primera vez, se recomienda siempre comenzar con una cantidad muy pequeña de ejecuciones (ej. 5), de forma que se pueda comprobar si el programa está operando como se desea (ej. Si las soluciones cumplen las metas requeridas), sin necesidad de esperar mucho tiempo. Sin

embargo, para tener una idea de la frecuencia de selección, será necesario hacer varias ejecuciones. Probablemente, cien ejecuciones es un valor mínimo para comenzar y es un valor intuitivo a partir del cual se puede calcular la frecuencia de selección. Al adicionar más ejecuciones se realizarán más muestreos del espacio de soluciones, pero, por supuesto, aumentará el tiempo del procesamiento. La cifra definitiva que usted decida, debe representar un equilibrio entre el tiempo consumido y la información obtenida. El MGPH ofrece algunas sugerencias útiles para la determinación de la cantidad óptima de ejecuciones.

3.2.1.1.2 Modificador de Longitud de Frontera

Variable – ‘BLM’

Requerido: No



Descripción: La variable ‘BLM’ (Modificador de Longitud de Frontera), se utiliza para determinar cuánto énfasis se debe poner en minimizar toda la longitud de frontera del sistema de reservas. Minimizar esa longitud llevará a un sistema de reservas más compacto, que pudiera desearse por varias razones pragmáticas. Hacer énfasis en la importancia de una red compacta significa que hay probabilidades de que sus metas cuantitativas se cumplan en menor cantidad de reservas extensas, generando, por lo general, un sistema de reservas más costoso y más extenso en su totalidad. De este modo, el BLM opera en oposición al otro objetivo de mayor importancia de Marxan, minimizar el costo general de la solución. El BLM puede analizarse como una escala móvil relativa, que abarca desde soluciones fragmentadas menos costosas (BLM bajo) hasta soluciones costosas más compactas (BLM alto). Como esto tendrá gran influencia en las soluciones definitivas, se necesitará tomar acciones para garantizar que se encuentre el valor (o rango de valores) apropiado.

Iniciar: El BLM debe ser ‘0’ o un número positivo. Es permisible para el BLM incluir puntos decimales (ej. 0.1). Llevando el BLM a ‘0’ hará que se elimine el análisis de la longitud de fronteras. No existe un valor universal para el BLM, ya que funciona en relación con los costos y la geometría de las unidades de planificación o regiones en estudio. Con un BLM, pequeño Marxan se concentrará en minimizar el costo general de reserva y sólo apuntará a lo compacto cuando se incurra en pequeños gastos extras. De manera alternativa, un BLM extenso hará un énfasis marcado en minimizar la longitud de las fronteras, incluso cuando eso signifique una solución más costosa. El usuario debe examinar los impactos de diferentes valores de BLM en aras de determinar un BLM adecuado para los objetivos del proyecto.

Aunque el nivel ‘correcto’ de compactación espacial es más bien un valor subjetivo, el cuadro siguiente ofrece algunos consejos prácticos. Como guía aproximada, un buen comienzo para el BLM es llevarlo a escala, de forma que la frontera más extensa entre las unidades de planificación se convierta en un orden de magnitud similar para las unidades de planificación más costosas. Por ejemplo, si su costo de unidad de planificación más alto es 100 y su frontera más extensa es 1000, se querrá comenzar con el BLM igual a 0.1. Nótese que, por lo general, es mejor explorar un rango de valores que se separan utilizando un multiplicador fijo por ejemplo, 0.04, 0.2, 1, 5, 25. En este ejemplo los valores se multiplican por 5. Casi siempre, los valores aumentan de manera exponencial o por órdenes de magnitud, en aras de realizar un muestreo de un rango de valores y escoger uno que equilibre el orden de magnitud de los términos de la función objetivo que compiten entre sí.

Establecer el BLM

El siguiente método para determinar un Modificador de Longitud de Frontera (BLM) eficiente fue tomado de Stewart y Possingham (2005).

1. Mantener todos los otros parámetros invariables, repetir el análisis de Marxan utilizando una serie de valores diferentes para el BLM, ej. 0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000, 1000000.
2. En una hoja de cálculo, registrar el BLM para cada escenario, la longitud total de fronteras del sistema de reservas, así como el costo promedio de las soluciones en columnas adyacentes. Si la zona de reserva es más importante que el costo, o está siendo utilizada como un sustituto del costo, registre la zona de reserva promedio de soluciones.
3. Determinar la longitud total de frontera de las reservas en oposición al costo/área total para los diferentes valores BLM, como se muestra en el siguiente cuadro.

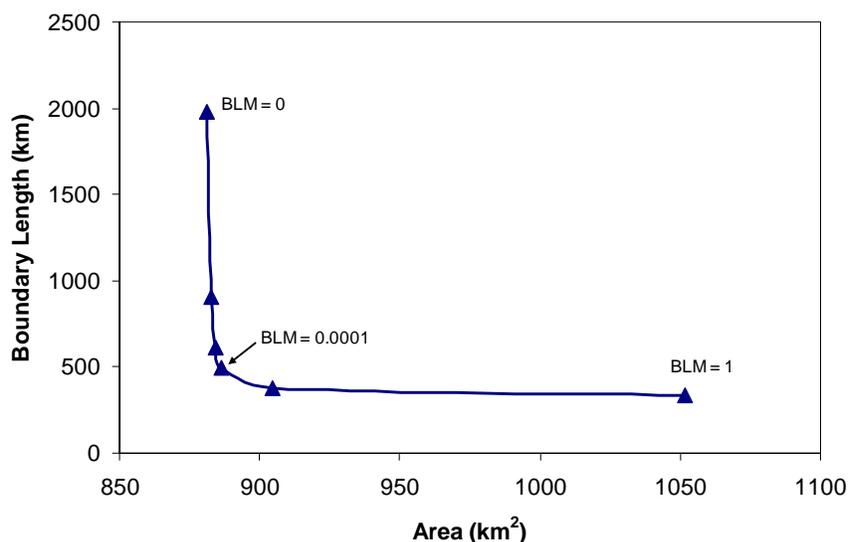


Fig 1 El equilibrio entre la longitud de frontera del sistema de reservas y su área total (modificado de Stewart y Possingham 2005, figura 1).

Los retornos, cada vez menores, al incrementar el BLM son claramente visibles en la Fig 1. En ese caso, un BLM de 0.0001 es posiblemente el más eficiente. Lo que se busca es el punto crítico en el cual el incremento en el costo del sistema de reservas o su área se torna alto con relación a la correspondiente reducción de la longitud de frontera del sistema. Esto representa un buen BLM de inicio. No obstante, las soluciones deben ser siempre inspeccionadas visualmente antes de fijar un BLM definitivo para garantizar que el sistema de reservas tenga un nivel adecuado de compactación.

Más información referente al impacto de la compactación espacial y sobre la cómo determinar un nivel eficiente para ésta, puede encontrarse en Stewart y Possingham (2005). Vea el Apéndice B-1.2 para una descripción detallada de su papel en la función objetivo. Asimismo, se puede encontrar más información referente a la aplicación de BLM en el MGPH.

3.2.1.1.3 Tipo de archivo de entrada

Variable – ‘VERSION’

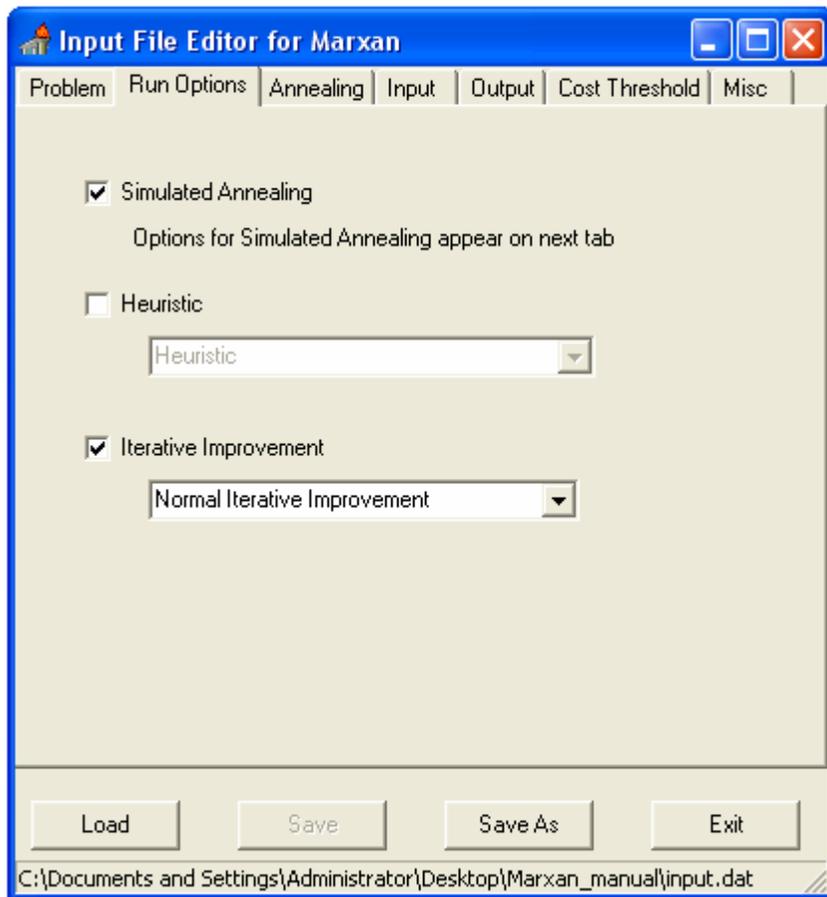
Requerido: Si

Descripción: Esta variable se ofrece sólo para permitir la compatibilidad con los archivos de SPEXAN (un precursor de Marxan). La diferencia entre los dos tipos de archivos es que Marxan puede leer valores de variables basándose en sus nombres, mientras que SPEXAN utilizaba su posición en el archivo, requiriendo un formato exacto.

Iniciar: Seleccione siempre ‘New Freeform Style’ (nuevo estilo de forma libre) en **Inedit**, o ‘0.1’ en el archivo ‘input.dat’ file.⁷

⁷ Teniendo en cuenta que Marxan tiene determinadas mejoras en comparación con SPEXAN, no hay beneficio alguno al utilizar SPEXAN, a menos que se replique un trabajo que utilizó a ese software en el pasado.

3.2.1.2 Opciones de ejecución



La ficha 'Run Options' (opciones de ejecución) de **Inedit**.

3.2.1.2.1 Opciones de ejecución

Variable – 'RUNMODE'

Requerido: Si

Descripción: Es una variable fundamental que define el método que Marxan va a utilizar para determinar buenas soluciones de reserva. Como se explicó en la introducción, la verdadera fortaleza de Marxan recae en el uso del templado simulado para encontrar soluciones al problema de selección de reservas. Sin embargo, Marxan es capaz de utilizar métodos más simples pero más rápidos para ubicar soluciones potenciales, como las reglas heurísticas y el mejoramiento iterativo (Vea el Apéndice B-2.2 para más detalles sobre esos métodos). Como las reglas heurísticas pueden ser aplicadas de manera muy rápida y generan resultados razonables, son incluidas para ser utilizadas en conjuntos de información extremadamente grandes. Las computadoras actuales tienen ya tanta potencia que los métodos heurísticos no son tan necesarios como artificios para ahorrar tiempo,

aunque aún se utilizan como herramientas de investigación. Ejecutar el Mejoramiento Iterativo por sí solo, puede resultar en soluciones muy pobres. Además de utilizar cada uno de esos tres métodos de manera independiente, Marxan los puede utilizar de manera coordinada. Si se selecciona más de uno, serán aplicados en el siguiente orden: Templado Simulado, Heurístico, Mejoramiento Iterativo. Eso significa que hay siete opciones diferentes de ejecución:

- 0 Aplicar Templado Simulado seguido de un Heurístico
- 1 Aplicar Templado Simulado seguido de un Mejoramiento Iterativo
- 2 Aplicar Templado Simulado seguido de un Heurístico, seguido de un Mejoramiento Iterativo
- 3 Utilizar sólo un Heurístico
- 4 Utilizar sólo un Mejoramiento Iterativo
- 5 Utilizar un Heurístico seguido de un Mejoramiento Iterativo
- 6 Utilizar sólo Templado Simulado

Aunque basta con insertar un número en el archivo 'input.dat' para que cada una de las combinaciones de ejecución mencionadas anteriormente puedan establecerse, sí se necesitan los tres campos del tabulador 'Run Options' de **Inedit** para definirlos. Por ejemplo, si se deseaba seleccionar Templado Simulado seguido de Mejoramiento Iterativo, primero necesita marcar la casilla de 'Templado Simulado', asegurarse de que la casilla 'Heurístico' esté en blanco, y luego marcar el cuadro de 'Mejoramiento Iterativo'.

Iniciar: Dentro de esas combinaciones, la más útil es el Templado Simulado seguido únicamente de Mejoramiento Iterativo (valor de variable '1'). Eso se debe a que el Templado Simulado busca de manera efectiva el espacio de solución y el Mejoramiento Iterativo garantiza entonces que la solución represente la mejor opción en la zona inmediata del espacio de decisión (conocido como 'mínimo local'). Esta combinación será la mejor para la mayoría de las aplicaciones y rara vez tendrá que ser cambiada.

3.2.1.2.2 Mejoramamiento Iterativo

Variable – 'ITIMPTYPE'

Requerido: No

Descripción: Si el Mejoramamiento Iterativo se utiliza para ayudar a encontrar soluciones, esa variable define qué tipo de Mejoramamiento Iterativo se aplicará. Existen cuatro opciones diferentes. En el Apéndice B-2.2 podrá encontrar detalles de dichas opciones:

- 0 Mejoramamiento Iterativo Normal
- 1 Mejoramamiento Iterativo de dos Pasos
- 2 Mejoramamiento Iterativo 'de Reemplazo'
- 2 Mejoramamiento Normal seguido de Mejoramamiento Iterativo de Dos Pasos

Iniciar: Para especificar esto en **Inedit** se utiliza el cuadro con el nombre 'Mejoramamiento Iterativo'. La predeterminada para esa variable es el 'Mejoramamiento Iterativo de Dos Pasos', y será apropiado para la mayoría de los escenarios.

3.2.1.2.3 Heurística

Variable – 'HEURTYPE'

Requerido: No

Descripción: Si se está utilizando un método heurístico opcional para encontrar soluciones de reserva, esa variable define qué tipo de algoritmo heurístico será aplicado. En el Apéndice B-2.3 se ofrecen detalles sobre los diferentes métodos heurísticos mencionados a continuación.

- 0 Riqueza
- 1 'Ávido'
- 2 Rareza Máxima
- 3 Rareza Óptima
- 4 Rareza Promedio
- 5 Rareza de Sumatoria
- 6 Irreemplazabilidad del Producto
- 7 Irreemplazabilidad de la Sumatoria

Iniciar: Para especificar esa variable en **Inedit** se utiliza el cuadro denominado 'Heurístico'. Sin embargo, recomendamos que los principiantes utilicen el templatado simulado para iniciar.

3.2.1.3 Templado

3.2.1.3.1 Cantidad de Iteraciones, Disminución de las Temperaturas, Temperatura Inicial y Factor de Enfriamiento

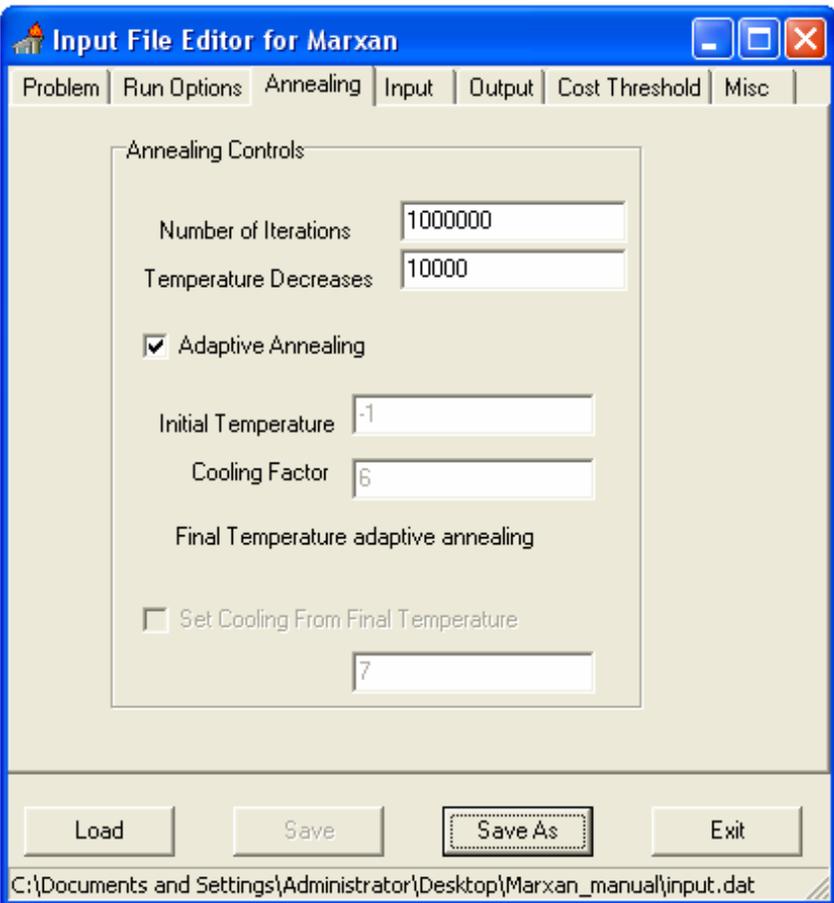
Variables –

‘NUMITNS’,
‘STARTTEMP’,
‘COOLFAC’, y
‘NUMTEMP’

Requerido: Si (si se usa el Templado Simulado)



Descripción:
Esas cuatro variables



La ficha ‘Annealing’ (templado) de **Inedit**.

controlan la forma en que procede el algoritmo de Templado Simulado. Ellas operarán cuando se selecciona el Templado Simulado en ‘RUNMODE’ (vea Sección 3.2.1.2.1). Esas variables son bastante técnicas y es necesario comprender todos los detalles para utilizar Marxan con efectividad. Dichas variables se explican en el Apéndice B-2.1.

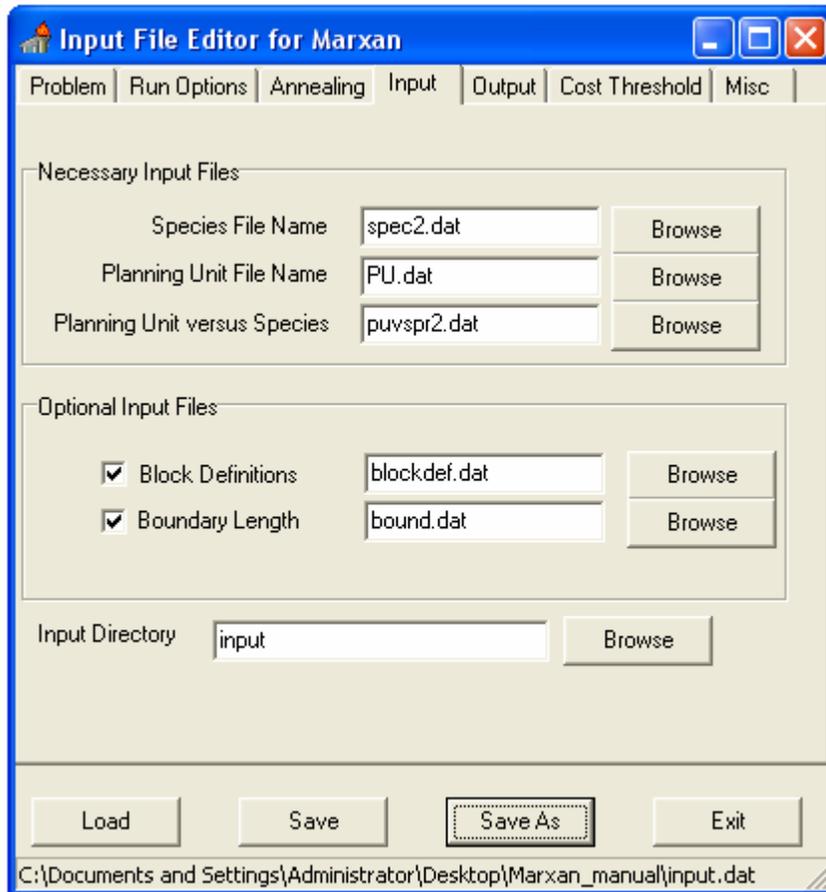
Iniciar: En la práctica, rara vez tendrá que hacerle ajustes. Las variables ‘STARTTEMP’ y ‘COOLFAC’ se establecerán adecuadamente con sólo marcar el cuadro de **Inedit** nombrado ‘Adaptive Annealing’ (Templado Adaptable), y para la mayoría de las aplicaciones es bastante razonable llevar el número de disminución de temperaturas (variable, ‘NUMTEMP’) a 10 000. El número de iteraciones que se establece (variable, ‘NUMITNS’) tiene un efecto sustancial en el tiempo que toma realizar cada ejecución. Por lo general, la cantidad de iteraciones determina cuánto se acercó Marxan a la solución óptima (o al menos a una solución muy buena). Se debe comenzar con una cifra alta (ej. 1,000,000) y luego incrementarla (ej. 10 millones o más se aplica comúnmente a conjuntos de información de gran escala) hasta que no se observe una mejora sustancial en los resultados y las iteraciones

continúen incrementándose. En un determinado punto, el tiempo extra que se requiere para obtener mayor cantidad de iteraciones, será mejor empleado en realizar mayor cantidad de ejecuciones que destinar mucho tiempo a cada una por separado. Seleccione un balance aceptable entre la eficiencia de la solución (costo total final o cantidad de unidades de planificación) y el tiempo de ejecución (cantidad de iteraciones).

3.2.1.4 Entrada

3.2.1.4.1 Nombre del Archivo de Especies, Unidades de Planificación, Unidades de Planificación versus Especies. Definición de Bloques, Longitud de Fronteras y Carpeta de Entrada

Variables –
 ‘INPUTDIR’,
 ‘SPECNAME’,
 ‘PUNAME’,
 ‘PUVSPRNAME’,
 ‘BOUNDNAME’, y
 ‘BLOCKDEFNAME’
 Requerido: Si



Descripción e Inicio:

La ficha ‘Input’, **Inedit**

Esas variables

encierran una explicación razonable de sí mismas, y sus protocolos para denominaciones y almacenaje se han analizado previamente (vea Sección 3.1.2). Aunque en la pantalla de **Inedit** se puede especificar el lugar de cada archivo utilizando los botones ‘Browse’ (examinar), los archivos deben situarse en la carpeta correcta para poder ejecutar Marxan.

3.2.1.5 Salidas

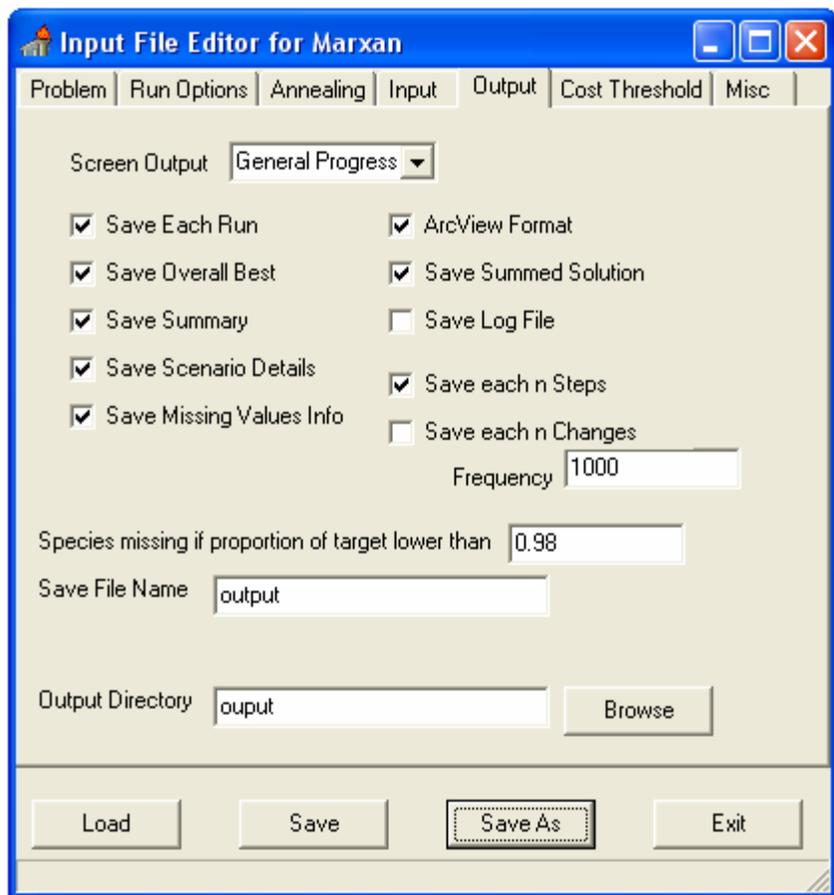
3.2.1.5.1 Salidas en Pantalla

Variable –

‘VERBOSITY’

Requerido: Si

Descripción: Esa variable controla la cantidad de información que Marxan muestra en la pantalla mientras se está ejecutando. En **Inedit** se establece utilizando el cuadro nombrado ‘Resultados en Pantalla’. Los usuarios pueden especificar la cantidad de información que Marxan refleja en la pantalla (la verbosidad).



La ficha ‘Output’, **Inedit**.

Existen cuatro

opciones para mostrar en la pantalla:

- 0 Ejecución Silenciosa – Solo se muestra el título del programa.
- 1 Sólo Resultados – Marxan mostrará que ejecución se realiza y el resultado de cada ejecución así como el tiempo total.
- 2 Progreso General – Además de la información sobre cada ejecución, Marxan mostrará información sobre los datos que ha leído así como los detalles referentes a cualquier objeto de conservación, cuyos objetivos y requerimientos no pueden ser reservados adecuadamente en el sistema.
- 3 Progreso Detallado – Muestra exactamente el estado del programa y brinda el valor del sistema cada vez que la temperatura cambia.

Inicio: El nombre predeterminado para esa variable es 'General Progress' (progreso general) y en la mayoría de los casos resultará ser la mejor opción. Reflejar los resultados en la pantalla no representa un incremento sustancial del tiempo de ejecución de Marxan, a menos que se utilice 'Detailed Progress' (progreso detallado). Por lo general, se recomienda utilizar al menos 'Results Only' (sólo resultados) para tener una idea de cuántas ejecuciones se han completado. 'Detailed Progress' es útil para ver como funciona el proceso de templado, y puede ayudar además a identificar los problemas durante la ejecución de Marxan (ej. Si las cifras no cambian y se encuentra "paralizado"). Por esa razón, algunos usuarios utilizan siempre esta configuración, para comprobar visualmente que el programa se está ejecutando sin problemas. Utilice 'Silent Running' (ejecución silenciosa) únicamente si está seguro de la ejecución de Marxan y se están guardando todos los resultados necesarios.

3.2.1.5.2 Salva de archivos y nombres de archivos

Variables – 'SAVERUN', 'SAVEBEST', 'SAVESUM', 'SAVESCEN', 'SAVETARGETMET', 'SAVESUMSOLN', 'SAVELOG', 'SAVESNAPSTEPS', 'SAVESNAPCHANGES', 'SAVESNAPFREQUENCY', y 'SCENNAME'

Requerido: No

Descripción e Inicio: Exceptuando 'SCENNAME' y 'SAVESNAPFREQUENCY', todas esas variables se utilizan para indicar a Marxan qué resultados debe guardar como salida. Al utilizar **Inedit** usted puede indicar a Marxan guardar cada uno de esos archivos sólo con marcar la casilla correspondiente. En el archivo 'input.dat', lleve el valor '1' para cada resultado que desee que Marxan guarde. Si desea expresar los resultados en un SIG, se preferirán los resultados delimitados por tabuladores a los delimitados por comas. En **Inedit** esto puede hacerse marcando la casilla denominada 'ArcView Format', y se hace directamente en el 'input.dat' llevando el valor de esas variables a '2'. En la próxima sección se brinda una explicación bien detallada referente a las diferentes salidas y sus usos.

En caso de seleccionar SAVESNAPSTEPS ('Guardar cada n pasos' en **Inedit**) o SAVESNAPCHANGES ('Guardar cada n cambios' en **Inedit**) se debe especificar también un valor de SAVESNAPFREQUENCY ('Frecuencia' en **Inedit**). Esa es la cantidad predeterminada de iteraciones del sistema (SAVESNAPSTEPS) o cambios del sistema (SAVESNAPCHANGES) en la que se guardan muestras instantáneas del progreso de solución del procedimiento de optimización.



Atención: guardar muestras instantáneas puede generar una enorme cantidad de archivos de salida que repletan su carpeta de salida y provocan

que la ejecución de Marxan se haga extremadamente lenta. Ellas se emplean para diagnósticos avanzados y deben ser utilizadas sólo por usuarios de experiencia. De usarlas, asegúrese de que la frecuencia de las muestras instantáneas sea suficientemente espaciada, de manera que no resulten en decenas de miles de archivos de salida. La cifra real que seleccionará va a depender de la cantidad de iteraciones que se esté utilizando (vea Sección (3.2.1.3.1)). Para un millón de iteraciones, una frecuencia de 100 mil muestras instantáneas resultará en 10 archivos de salida.

La variable 'SCENNAME', o ('Save File Name' en **Inedit**) (Guardar el Nombre del Archivo), es la denominación que usted desea que Marxan dé a todos los archivos de salida que guarde (ej. 'scenario1_ssoln.dat' sería el nombre dado a la salida de solución sumada). El nombre debe ser algo que usted utilice para identificar el escenario que generó los resultados.

3.2.1.5.3 Directorio de Salida

Variable – 'OUTPUTDIR'

Requerida: Si

Descripción e Inicio: La variable se utiliza para indicar a Marxan el nombre de la carpeta (nombrada directory (directorio) o DIR en Marxan) donde debe salvar los archivos de salida. Las denominaciones y protocolos referentes a esa carpeta se explican en la Sección 5.1 y como mismo sucede con la carpeta de entrada, es de suma importancia que se haga correctamente o, de lo contrario, Marxan no se ejecutará.

3.2.1.5.4 Proporción no alcanzada de las especies

Variable – 'MISSLEVEL'

Requerido: No

Descripción: Es la proporción de la meta que debe alcanzar un objeto de conservación para que se reporte su meta como cumplida. Utilizando **Inedit**, se especifica en la casilla nombrada 'Species missing if proportion of target lower than' (especies no alcanzadas si la proporción de la meta es menor que). Hay situaciones donde Marxan puede acercarse mucho a la meta (ej. 99% del nivel deseado) sin llegar a cumplirla realmente. Se puede especificar la proporción de representación, con cifras cercanas a la meta con la que quedaría pragmáticamente satisfecho para reportar dicha meta como cumplida.

Iniciar: Ese valor siempre debe ser alto, por ejemplo, mayor o igual a 0.95. Si lo establece por debajo de esa cifra, quizás debiera considerar cambiar las metas cuantitativas. Como guía, con frecuencia resulta útil ejecutar Marxan llevando el 'MISSLEVEL' a '1' y ejecutar Marxan nuevamente, disminuyendo un poco el valor de la variable, para analizar si hay diferencias muy marcadas en el costo del sistema. Configurar esa variable no cambia la forma en que opera el algoritmo de Marxan, únicamente cambia la forma de reportar las metas cuantitativas en pantalla y en el archivo de salida

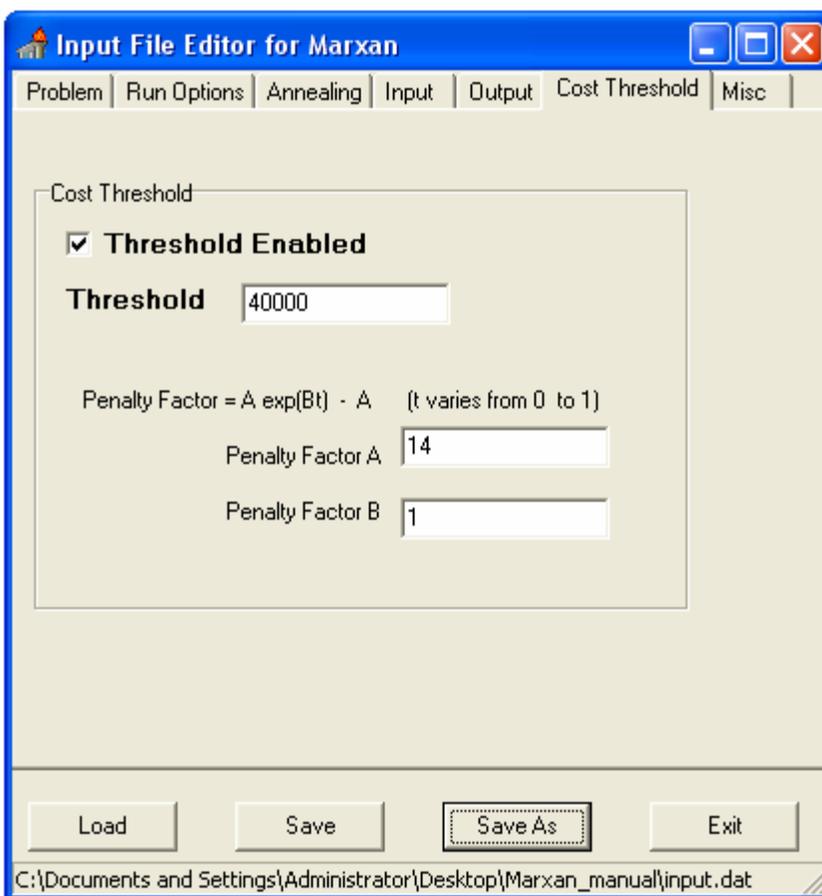
3.2.1.6 Umbral de Costo

3.2.1.6.1 Umbral, Factor de Penalidad A y Factor de Penalidad B

Variables –
 'COSTTHRESH',
 'THRESHPEN1', y
 'THRESHPEN2'

Requerido: No

Nota: Algunos usuarios, al utilizar esa variable, han reportado que el costo total resultante de las redes de reservas, excede el umbral del costo especificado. Si el problema ocurre, resalta en las tablas de salida. Actualmente, Marxan está siendo rediseñado en aras de lograr mayor confiabilidad con respecto a esa variable. Si obtiene como resultado redes de reserva que excedan el umbral especificado, deseche esos resultados.



La ficha 'Cost Threshold', **Inedit**.



Descripción: Esas variables se pueden incluir si se desea que Marxan encuentre soluciones de reservas por debajo de un costo total. Como se explicó en la introducción, Marxan está diseñado para dar solución a un problema 'de conjunto mínimo', siendo su propósito satisfacer todas nuestras metas de conservación con un costo mínimo. Otro tipo de problema de conservación es conocido como problema de 'máxima cobertura', donde el objetivo es lograr los mejores resultados de conservación para un presupuesto fijo determinado. En

muchos casos, esto representa mejor la manera en operan las acciones de conservación. Aunque la inclusión de un umbral de costo no hace que Marxan solucione el problema específico de ‘cobertura mínima’, sí es comparable y se puede utilizar en muchos casos donde se cuenta con varias metas de conservación con las que desea cumplir y, sin embargo, no se puede exceder un presupuesto predeterminado. En el Apéndice B-1.4 se describe en detalles la manera real en que ese umbral de costo se aplica dentro de los algoritmos.

Inicio: Llevando esa variable a ‘0’ en el archivo ‘input.dat’, hará que se deshabilite.



Se debe ser cauteloso al utilizar esa función ya que puede afectar la capacidad de Marxan de encontrar soluciones efectivas.

3.2.1.7 Misceláneas

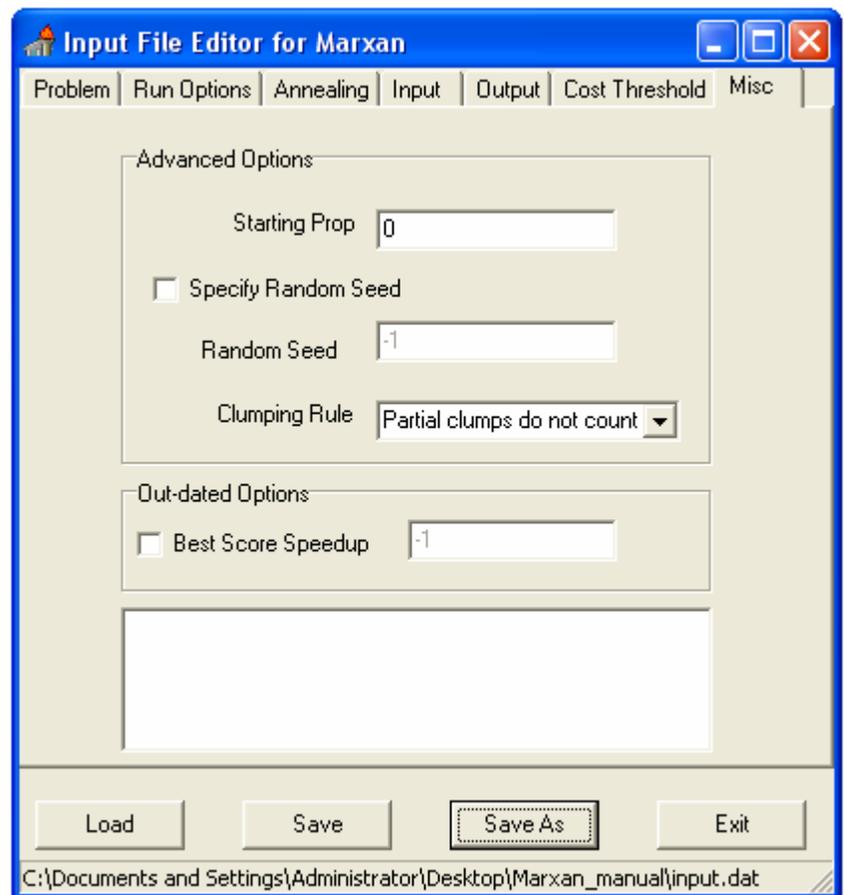
3.2.1.7.1 Iniciar Prop

Variable – ‘PROP’

Requerida: No

Descripción: Cuando Marxan comienza una ejecución, se debe general un sistema de reservas inicial. Esa variable define la proporción de unidades de planificación que serán incluidas en el sistema de reservas inicial al comienzo de cada ejecución.

Iniciar: La variable ‘PROP’ debe ser una cifra entre 0 y 1, y en **Inedit** se establece utilizando la casilla denominada ‘Starting Prop’ (Proporción inicial). Si se escoge cero, no se incluirán entonces unidades de planificación en la reserva inicial, un valor de 1 significa que se incluirán todas las unidades de planificación, y un valor de 0.5 significa que el 50 % de las unidades de planificación será incluido de forma aleatoria. En la práctica, la configuración no tendrá efecto alguno en la operación de templado simulado, siempre que se utilicen suficientes iteraciones.



La ficha ‘Misc’, **Inedit**.

Eso sólo se aplicará a aquellas unidades de planificación en las que su condición no las pre-incluya o pre-excluya de las soluciones. (Vea Sección 3.2.3.3).

3.2.1.7.2 Semilla aleatoria

Variable – ‘RANDSEED’

Requerido: No

Descripción: No se preocupe mucho por esa variable. Controla si se incluye la misma selección ‘aleatoria’ en el sistema de reservas inicial de cada ejecución. Utilizar un número entero positivo constante para esa variable, hará que Marxan utilice la misma semilla aleatoria cada vez que se ejecute.

Iniciar: Excepto para una depuración, esa variable no se debe marcar en **Inedit** (ej. hágala ‘-1’ en el archivo de entrada).

3.2.1.7.3 Regla de Agrupamiento

Variable – ‘CLUMPTYPE’

Requerido: No

Nota: Algunos usuarios, al utilizar esta variable han encontrado que las configuraciones de reservas no cumplen los requisitos de agrupamiento especificados. Si el problema ocurre, resalta en las tablas de salida. Recomendamos que se deseche cualquier resultado que no cumpla los requisitos de agrupamiento. (vea Sección 3.2.2.5 para más información).



Descripción: Esa variable es útil si algunos objetos de conservación tienen establecidos un tamaño mínimo de agrupamiento (target2, vea Sección 3.2.2.5). Ella indica a Marxan si incidencias por debajo del tamaño mínimo de agrupamiento pudieran contribuir a lograr la meta general, y de ser así, cómo. Conozca que esto hace la ejecución de Marxan más lenta por un orden de magnitud.

Inicio: El ‘CLUMPTYPE’ se establece en **Inedit** utilizando la casilla denominada ‘Clumping Rule’(regla de agrupamiento). Hay tres opciones para esta variable:

- 0 Los agrupamientos parciales no cuentan – Agrupamientos más pequeños que la meta no acumulan.
- 1 Agrupamientos parciales cuentan la mitad – Agrupamientos más pequeños que la meta acumulan la mitad de sus cantidades.
- 3 Penalidad Graduada – El resultado es proporcional al tamaño del agrupamiento.

3.2.1.7.4 Agilización del mejor resultado

Variable – ‘BESTSCORE’

Requerido: No

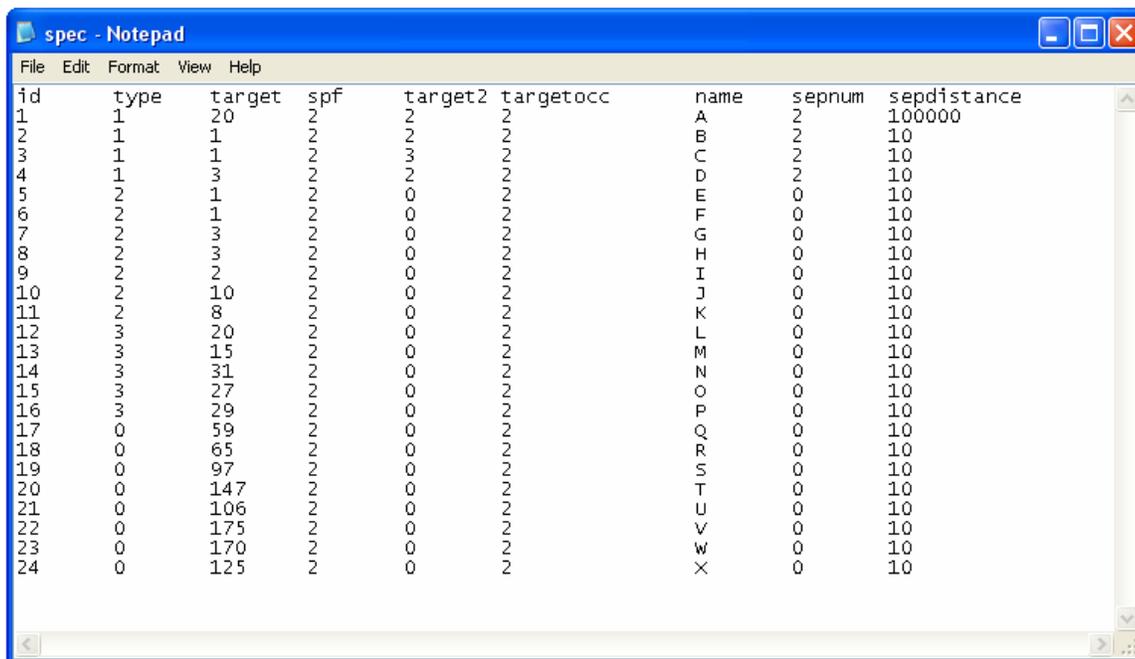
Descripción: Esta variable indica a Marxan que no haga registros del mejor resultado hasta que no alcance un nivel mínimo especificado. En **Inedit** se establece utilizando la casilla denominada ‘Best Score Speedup’ (agilizar el mejor resultado). Esta variable se creó como medida para ahorrar tiempo, sin embargo, rara vez se requiere su uso.

Inicio: Siempre debe llevarse a ‘-1’.

3.2.2 El Archivo de Objetos de Conservación

El **Archivo de Objetos de Conservación** contiene información sobre cada uno de los objetos de conservación que están siendo analizados, como sus nombres, presentación de la meta y la penalidad si no se cumple con la representación de la meta. Su nombre predeterminado es 'spec.dat'. Debido a este nombre, en ocasiones se hace referencia a él como el **Archivo de Especies**, aunque algunos objetos de conservación serán con frecuencia sustitutos, como el tipo de hábitat en lugar de especies reales. Algo muy importante, es que este archivo no contiene información sobre la distribución de los objetos de conservación en las unidades de planificación. Esta información aparece en el **Archivo de Unidades de Planificación versus Objetos de Conservación**.

El **Archivo de Objetos de Conservación** puede contener hasta siete variables, aunque no se requieren todas. Cada una esas variables, cuando son incluidas, se presenta en una columna con el nombre de esa variable como título de la columna.



id	type	target	spf	target2	targetocc	name	sepnum	sepdistance
1	1	20	2	2	2	A	2	100000
2	1	1	2	2	2	B	2	10
3	1	1	2	3	2	C	2	10
4	1	3	2	2	2	D	2	10
5	2	1	2	0	2	E	0	10
6	2	1	2	0	2	F	0	10
7	2	3	2	0	2	G	0	10
8	2	3	2	0	2	H	0	10
9	2	2	2	0	2	I	0	10
10	2	10	2	0	2	J	0	10
11	2	8	2	0	2	K	0	10
12	3	20	2	0	2	L	0	10
13	3	15	2	0	2	M	0	10
14	3	31	2	0	2	N	0	10
15	3	27	2	0	2	O	0	10
16	3	29	2	0	2	P	0	10
17	0	59	2	0	2	Q	0	10
18	0	65	2	0	2	R	0	10
19	0	97	2	0	2	S	0	10
20	0	147	2	0	2	T	0	10
21	0	106	2	0	2	U	0	10
22	0	175	2	0	2	V	0	10
23	0	170	2	0	2	W	0	10
24	0	125	2	0	2	X	0	10

Un ejemplo del **Archivo de Objeto de Conservación** (spec.dat) utilizado en Marxan.

Nota: Es esencial que los nombres de encabezamiento sean exactos. Nótese que todas las letras son minúsculas. Para todas las variables, exceptuando 'id' y 'name', el valor predeterminado es 0. Marxan podrá ejecutarse incluso si falta información de esas variables, pero los objetos de conservación tomarán los valores predeterminados ante la ausencia de atributos.

3.2.2.1 ID del Objeto de Conservación

Variable – ‘id’

Requerido: Si

Descripción: Un identificador numérico único para cada objeto de conservación. Se debe ser cuidadoso de no duplicar ningún número de identificación ya que Marxan obviará todos excepto el último.

Inicio: Es útil establecer un sistema lógico para la enumeración de los objetos. De esta manera, con sólo mirarlos se puede tener una idea de lo que son. Por ejemplo, todos las aves tiene un 1 en el cuarto dígito, ej. 1003. El quinto dígito puede representar regiones si se le da un diferente tratamiento a los objetos en las diferentes regiones, ej 21003 significaría ave 3 en la región 2. Independientemente del sistema por el que se decida, documéntelo y sea consistente en su uso.

3.2.2.2 Tipo de Objeto de Conservación

Variable – ‘type’

Requerido: No

Descripción: Se utiliza para definir los grupos de objetos de conservación para los cuales se pueden establecer determinados atributos que son similares para los mismos objetos dentro del grupo especificado (o “type”). Cada grupo de objetos debe tener un único identificador numérico. Esta variable se utiliza en combinación con el **Archivo de Identificación de Bloques** (vea Sección 3.3.2) donde se incluyen los atributos asignados a un grupo particular de objetos de conservación.

Iniciar: Todas las variables que se deseen incluir en los atributos del Archivo de Definición de Bloque deben ser introducidas con un valor de –1 en el **Archivo de Objeto de Conservación** (vea Sección 3.3.2 para un ejemplo). De lo contrario, el valor en el **Archivo de Objeto de Conservación** anulará la definición de bloque. Recomendamos que, siempre que sea factible, se utilice el Tipo de Objeto de Conservación junto al **Archivo de Definición de Bloque**, ya que simplifica los cambios, evita los errores tipográficos, y hace posible el fácil establecimiento de metas cuantitativas proporcionales (porcentajes). Este es el único mecanismo para establecer metas proporcionales en ésta y las versiones anteriores de Marxan.

3.2.2.3 Meta Cuantitativa de Representación del Objeto

Variable – ‘target’

Requerido: Si



Descripción: Es la meta cuantitativa para cada objeto de conservación que será incluido en la solución. Esos valores representan limitantes en las soluciones potenciales al problema de selección de reservas. Significa, que para que una solución de reservas sea factible, debe incluir, al menos, una meta para cada objeto. La meta deseada se expresa en las mismas unidades utilizadas para definir la cantidad de cada objeto presente en cada unidad de planificación, que se incluye en el **Archivo Unidad de Planificación versus Objetos de Conservación**. (Vea Sección 3.2.4). Sin embargo, las unidades de los diferentes objetos de conservación pueden variar (ej. Hectáreas de hábitat para un objeto y cantidad de incidencias para otra, nidos para una tercera y longitud de corrientes fluviales para una cuarta).

Iniciar: La meta es definida por el usuario y puede tomar cualquier valor desde 0 hasta la suma total de ese objeto encontrado en todas las unidades de planificación. Se debe ser cuidadoso para no establecer una meta cuantitativa mayor de la que realmente se pueda alcanzar, basándose únicamente en la incidencia de un objeto en las unidades de planificación, ya que esas metas pudieran no ser alcanzables. La selección de las metas adecuadas para los objetos de conservación puede reflejar los objetivos de representación establecidos en la legislación o en una convención, para el sistema de áreas protegidas. (ej. 20% del territorio nacional protegido). Sin embargo, las metas cuantitativas no tienen que ser valores uniformes para todas las especies (ej. siempre 20%). Por el contrario, deben reflejar la importancia de la conservación percibida para ese objeto, por ejemplo, se pudiera desear que los objetos de conservación poco comunes o más amenazados dispongan de metas cuantitativas mayores que las especies comunes. Cualquiera que sean las metas escogidas, es importante que estén bien argumentadas ya que tendrán un gran impacto en el carácter de los sistemas de reservas potenciales. Mientras mayor sea la meta, menor cantidad de posibles soluciones Marxan podrá encontrar. Explicaciones sobre este aspecto se brindará en el MGPH.

Si el **Archivo de Definición de Bloques** se utiliza para un objeto, entonces la Meta de Representación de ese objeto debe hacerse -1.

3.2.2.4 Factor de Penalidad del Objeto de Conservación

Variable – ‘spf’

Requerido: Si



Descripción: Las letras ‘spf’ son la abreviatura de *Species Penalty Factor*, (factor de penalidad de especies). La manera más correcta de referirse a esa variable es Factor de Penalidad del Objeto de Conservación. El factor de penalidad es un multiplicador que determina la magnitud de la penalidad que se adicionará a la función objetivo si no se cumple con la meta del objeto de conservación en el escenario de reserva que se analiza (vea Apéndice B-1.4 para los detalles referentes al cálculo y aplicación de esa penalidad.) Mientras mayor sea el valor, mayor será la penalidad relativa, y mayor será el énfasis que pondrá Marxan en garantizar que se cumplan las metas del objeto. De esta manera, las SPF (por sus siglas en inglés) son una manera de distinguir la importancia relativa de los diferentes objetos de conservación. Los objetos de alto valor de conservación, por ejemplo, los objetos muy amenazados o aquellos de gran importancia social o económica, deben tener valores de SPF mayores que los objetos menos importantes. Esto significa que se está menos dispuesto a comprometer sus representaciones en el sistema de reservas. Resulta esencial seleccionar un valor adecuado para esa variable en aras de lograr buenas soluciones en Marxan. Si resulta muy bajo, la representación de los objetos de conservación puede no cumplir con suficientes metas cuantitativas. Si resulta muy alto, se verá afectada la capacidad de Marxan para encontrar buenas soluciones (ej. Sacrificará otras propiedades del sistema como menor costo y mayor compacidad en un esfuerzo por cumplir a cabalidad con las metas de los objetos de conservación).

Iniciar: Con frecuencia se requerirá cierto grado de experimentación para determinar SPFs adecuados. Esto debe hacerse de forma iterativa. Un buen comienzo es seleccionar el valor más bajo que se encuentre en el mismo orden de magnitud de las cifras de objetos de conservación, por ejemplo, si se tiene 30 objetos, comience examinando los SPFs, de digamos, 10 para todas las objetos. Realice diferentes repeticiones de ejecuciones (quizás 10) y analice si sus metas se cumplen en la solución. Si no se cumplen todas las metas, pruebe incrementar el SPF por un factor o dos, y realice nuevamente las repeticiones de las ejecuciones. Cuando llegue al punto donde se han cumplido todas las metas cuantitativas, disminuya ligeramente los SPFs y analice si aún se siguen cumpliendo. Una vez que se organicen las ejecuciones de prueba, se pueden aplicar los valores relativos diferidos, basados en consideraciones como la rareza, el significado ecológico, etc., como se explicó anteriormente.

Incluso si se cumple con todas sus metas, trate siempre con valores más bajos. Al tratar de obtener los SPF más bajos que originan soluciones satisfactorias, Marxan tiene mayor flexibilidad para generar buenas soluciones. En general, a menos que se tenga alguna razón específica para ponderar la inclusión de objetos en su sistema de reservas, se debe comenzar con el mismo SPF para todos los objetos. Sin embargo, si constantemente se incumple la meta para uno o dos objetos, incluso cuando todos los demás objetos estén representados de manera adecuada, pudiera resultar conveniente incrementar el SPF para esos objetos. Una vez más, lo remitimos al MGPH para más detalles referentes al SPFs.

Si se utiliza el **Archivo de Definición de Bloque** para un objeto, entonces el Factor de Penalidad del Objeto de Conservación debe aquí hacerse igual a -1.

3.2.2.5 Tamaño Mínimo de Agrupamiento

Variable – ‘target2’

Requerido: No

Nota: Algunos usuarios, al utilizar esta variable, han encontrado configuraciones de reservas que no cumplen los requerimientos de agrupamiento especificados. Si el problema se manifiesta, resalta en las tablas de salida y recomendamos que deseche cualquier resultado que no cumpla sus requerimientos de agrupamiento.



Descripción: Esta variable especifica un tamaño mínimo de agrupamiento para la representación de los objetos de conservación en el sistema de reservas. Si la cantidad de objetos de conservación encontrada en un agrupamiento es menor que este valor, entonces no cuenta para el cumplimiento de la meta de conservación (la variable– ‘target’) de ese objeto. Esto resulta útil en casos donde pequeños y aislados territorios o poblaciones tienen valores de conservación menores que otros territorios o poblaciones mayores y bien conectados.

Iniciar: Para iniciar, es mejor no utilizar esa variable y analizar cómo los objetos se agrupan sin ella. Si, posteriormente, algunos objetos específicos requieren agrupamiento adicional, se podrá aplicar la variable.

Como sucede con el objetivo de conservación, el valor ‘target2’ debe ubicarse en las mismas unidades utilizadas para definir la cantidad de cada objeto en cada Unidad de Planificación, que se incluye en el **Archivo de Unidad de Planificación versus Objeto de Conservación** (vea la Sección 3.2.4). Por ejemplo, si usted ha incluido información sobre diferentes tipos de hábitat dentro de las Unidades de Planificación,

entonces el valor 'target2' especifica un área mínima (que puede de hecho cumplirse en una única unidad de planificación). En el caso de la información de presencia y ausencia, los valores 'target2' indican la cantidad de incidencias en las unidades de planificación contiguas, que deben tener lugar antes de que el agrupamiento contribuya a cumplir las metas (que se pueden cumplir también en una única unidad de planificación).



Se debe tener cuidado al establecer el tamaño mínimo de agrupación ya que las metas para algunos objetos no tendrán muchas opciones sino ser cumplidas en pequeñas y aisladas incidencias.

Recomendaciones para utilizar la opción de Agrupamiento de Marxan

Hay dos razones principales por las que los usuarios enfrentan dificultades con la opción de agrupamiento de Marxan. Esta funcionalidad se relaciona con los parámetros target2, sepnum y sepdistance.

- 1) No existe una solución para el problema de agrupamiento. Se puede determinar si ese es el caso realizando un análisis SIG con las capas de objetos y unidades de planificación.
- 2) El algoritmo relativamente poco sofisticado que se utiliza para encontrar soluciones que contienen agrupamientos no puede encontrar las soluciones al problema de agrupamiento. El algoritmo no realiza una búsqueda exhaustiva en el espacio de decisión para encontrar una solución de agrupamiento.

A pesar de esa limitación, la funcionalidad de agrupamiento se ha utilizado con efectividad en todo el mundo para solucionar problemas de investigación y manejo con una gama de juegos de datos. Hay varias técnicas generales que se pueden utilizar para manejar esta limitación. No existe actualmente un proyecto de investigación y desarrollo destinado a hacer frente a esa limitación.

Para gran cantidad de reinicios de Marxan, se pueden encontrar las soluciones si estas existen para algunos de esos reinicios. La proporción de reinicios estará relacionada con la dificultad de Marxan para encontrar soluciones al problema de agrupamiento.

En la rutina de inicialización de Marxan, se calcula una solución al problema, y se establecen las penalidades de los objetos de conservación sobre la base de esa solución. Si dicha solución no cumple los objetivos de agrupamiento, por consiguiente se utilizarán penalidades incorrectas para esa ejecución de Marxan, conllevando a operaciones incorrectas.

Se pueden seguir los siguientes pasos para computar las soluciones de agrupamiento con Marxan;

- 1) Reiniciar Marxan.
- 2) Observar si las metas de objetos se cumplen con la ejecución 1. Si las metas no se cumplen, detenga la ejecución de Marxan y reinícielo.
- 3) Repetir esa operación hasta que se computen las buenas soluciones requeridas.

Una forma simple y elegante de lograr el agrupamiento es utilizando el modificador de longitud de frontera. Este método es muy efectivo y rápido, sin embargo, no tiene un control preciso del tamaño, la cantidad y la separación de los agrupamientos.

3.2.2.6 Metas Cuantitativas para las Incidencias de Objetos de Conservación

Variable – ‘targetocc’

Requerido: No

Descripción: Esa variable especifica la cantidad mínima de incidencias de un objeto de conservación requeridas en el sistema de reservas. Ese valor puede ser utilizado en situaciones donde, aún cuando sus metas de conservación se cumplan en una unidad de planificación, se desea que el objeto estuviese representado en una mayor cantidad de unidades de planificación, posiblemente para la dispersión de riesgos.

Iniciar: Esta es una variable más bien especializada que se utiliza sólo en ocasiones. A diferencia de ‘target’ y ‘target2’, el valor de ‘targetocc’ no guarda relación con las unidades utilizadas para describir la incidencia de objetos de conservación, es sencillamente la cantidad de unidades de planificación en las que debe aparecer el objeto para una solución de reservas viable. Esta variable puede ser utilizada en combinación con ‘target’, o en lugar de esta.

Si el **Archivo de Definición de Bloques** se utiliza para esta característica, entonces la Meta para las Incidencias de Objetos debe hacerse igual a -1.

3.2.2.7 Nombre del Objeto de Conservación

Variable – ‘name’

Requerido: No

Descripción: El nombre alfabético (no números) de cada objeto de conservación (ej. *bosque nublado*). No es una variable usual ya que en ella se incluyen espacios. Si desea incluir espacios en el nombre, utilice palabras y no números para permitirle a Marxan leer una serie de palabras como una variable única. Si se desea, se pueden utilizar también números como el nombre, sin espacios entre ellos.

3.2.2.8 Meta Cuantitativa para Incidencias de Objetos Aislados

Variable – ‘sepnum’

Requerido: No

Nota: Algunos usuarios, al utilizar esta variable, han encontrado configuraciones de reserva resultantes que no cumplen los objetivos de agrupamiento especificados. Si el problema se manifiesta, resalta en las tablas de salida y recomendamos que se deseche cualquier resultado que no cumpla con los requerimientos de agrupamiento (vea Sección 3.2.2.5 para más información).



Descripción: La cantidad de incidencias, *separadas* mutuamente, de un objeto que es requerida en el sistema de reservas. Es similar a 'targetocc', excepto que si desea incluir múltiples incidencias de objetos para la diseminación del riesgo, entonces será conveniente que las unidades de planificación donde se encuentran esas incidencias no sean adyacentes entre ellas. Cuando se establece un tamaño mínimo de agrupamiento utilizando 'target2', la variable 'sepnum' hace referencia a la cantidad requerida de incidencias separadas mutuamente en agrupamientos válidos.

Iniciar: Esta es una característica avanzada dirigida a la replicación. Marxan puede ejecutarse primero sin ella y posteriormente, aplicándola.



Esta característica hace que Marxan sea más lento por un orden de magnitud. Si se utiliza el **Archivo de Definición de Bloques** para esta característica, entonces la Meta para las Incidencias de Objetos Separados debe hacerse igual a -1.

3.2.2.9 Distancia Mínima de Separación

Variable – 'sepdistance'

Requerido: No

Nota: Algunos usuarios, al utilizar esta variable, han encontrado configuraciones de reservas resultantes que no cumplen los requerimientos de agrupamiento especificados. Si el problema se manifiesta, resalta en las tablas de salida, y recomendamos que se deseche cualquier resultado que no cumpla los requerimientos de agrupación (vea la Sección 3.2.2.5 para más información).



Descripción: Utilizada en combinación con 'sepnum' (arriba) esta variable especifica la distancia mínima en la que se consideran separadas las unidades de planificación que contienen una objeto de conservación. Esto pudiera resultar de utilidad en situaciones donde se desea trabajar con las incidencias múltiples debido a la amenaza de eventos de gran escala como huracanes o fuegos. Por ejemplo, si los huracanes, por lo general, dañan el hábitat en una distancia conocida, pudiéramos desear que dos incidencias del mismo objeto estén separadas por una distancia mayor. La distancia de separación puede estar también relacionada con la capacidad de dispersión de las especies invasoras.

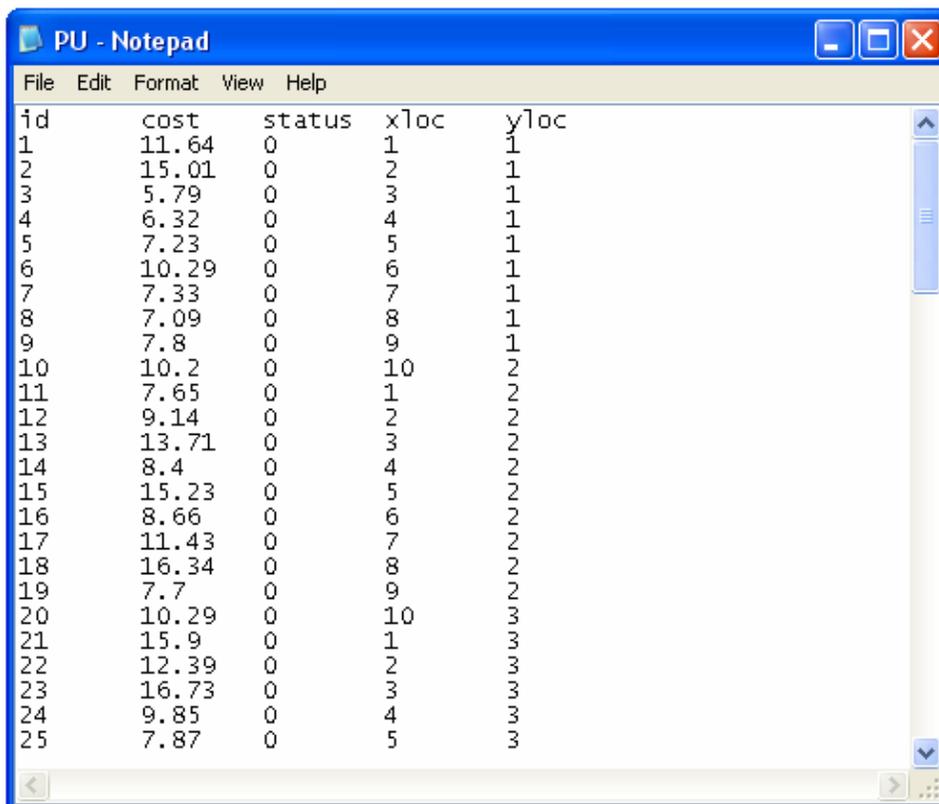
Iniciar: Para utilizar esa variable, la ubicación geográfica de cada unidad de planificación debe especificarse en el **Archivo de Unidad de Planificación** (vea Sección 3.2.3.4).



Esta característica puede hacer que Marxan se ejecute mucho más lento. Si se utiliza el **Archivo de Definición de Bloques** para este objeto, entonces la Distancia Mínima de Separación debe aquí hacerse igual a -1.

3.2.3 El Archivo de Unidades de Planificación

El **Archivo de Unidades de Planificación** contiene toda la información relacionada con las unidades de planificación, excepto la referente a la distribución de los objetos de conservación en las unidades de planificación (que se encuentra en el **Archivo de Unidades de Planificación versus Objetos de conservación**). El nombre predeterminado para este archivo es 'pu.dat'. El **Archivo de Unidad de Planificación** puede contener hasta cinco variables, aunque sólo una de ellas es obligatoria ('id'). Al ser incluidas, cada una de esas variables es representada en una columna con el nombre de la variable encabezando la columna.



id	cost	status	xloc	yloc
1	11.64	0	1	1
2	15.01	0	2	1
3	5.79	0	3	1
4	6.32	0	4	1
5	7.23	0	5	1
6	10.29	0	6	1
7	7.33	0	7	1
8	7.09	0	8	1
9	7.8	0	9	1
10	10.2	0	10	2
11	7.65	0	1	2
12	9.14	0	2	2
13	13.71	0	3	2
14	8.4	0	4	2
15	15.23	0	5	2
16	8.66	0	6	2
17	11.43	0	7	2
18	16.34	0	8	2
19	7.7	0	9	2
20	10.29	0	10	3
21	15.9	0	1	3
22	12.39	0	2	3
23	16.73	0	3	3
24	9.85	0	4	3
25	7.87	0	5	3

Un ejemplo del Archivo de Unidad de Planificación (pu.dat) utilizado en Marxan.

Es esencial que los nombres de encabezamiento sean exactos. Todos se deben escribir en letras minúsculas y sin puntuación, espacios o caracteres numéricos. El orden en que se presenten los encabezamientos no es de importancia.

3.2.3.1 Identificación de la Unidad de Planificación

Variable – ‘id’

Requerido: Si

Descripción: Un identificador numérico único para cada unidad de planificación. Los valores para la variable ‘id’ pueden ser cualquier número (ej. no se requiere comenzar con el número 1) pero no pueden incluir espacios, letras o signos de puntuación. Existe un límite máximo de 20 000 a 30 000 en cuanto a la cantidad de unidades de planificación con las que la versión básica de Marxan puede operar, aunque la versión optimizada no tiene esas restricciones (vea Sección 1.87.1).



Este número no puede confundirse con la variable, ‘id’, en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2).

3.2.3.2 Costo de las Unidades de Planificación

Variable – ‘cost’

Requerido: No



Descripción: Expresa el costo de incluir cada unidad de planificación en el sistema de reservas. El valor ingresado para esta variable será la cantidad adicionada a la función objetivo (vea Sección 1.5) cuando esa unidad de planificación se incluya en el sistema de reservas. El costo de la unidad de planificación puede basarse en cualquier medida. Es una variable con la que se debe tener sumo cuidado ya que puede tener una gran influencia en las soluciones generadas por Marxan.

Iniciar: El uso adecuado de la función de costo puede ser complejo. En su manera más simple, el costo de todas las unidades de planificación puede llevarse a 1. Marxan entonces comprobará y minimizará la cantidad total de unidades de planificación incluidas en el sistema de reservas, pero valorará la selección de las unidades de planificación basándose únicamente en los objetos presentes y no en el costo. Si sus unidades de planificación no son equivalentes en tamaño, una medida equivalente “predeterminada” es utilizar el área de la unidad de planificación como su costo. Como una alternativa, el MGPH describe el uso de un valor de área transformado. Esto se basa en el supuesto de que mientras mayor sea el tamaño de la reserva, más costoso resultará su implementación y manejo, aunque no siempre es el caso, y los costos casi nunca so lineales.

Una alternativa más sofisticada (y generalmente mejor) es utilizar una medida del costo fiscal real que representa incluir las unidades de planificación en un sistema de reservas (por ejemplo vea Naidoo et al. 2006). Este puede ser el costo requerido para comprar una parcela de tierra, o el costo de oportunidad de los usos alternos de la tierra y el mar que no son compatibles con la conservación. El costo puede ser también cualquier medida social, económica o ecológica. Por ejemplo, pudiera reflejar la posibilidad de eventos en diferentes zonas basándose en la disposición social, la observancia por el cumplimiento, o la presencia de amenazas incontrolables.



Aunque sólo puede definirse un costo único para cada unidad de planificación, este puede ser una composición de diferentes medidas, dado que haya una base que justifique la combinación de dichas medidas. Los costos de los mismos tipos de moneda se pueden combinar (si ambos costos son monetarios). No es prudente que se combinen costos expresados en unidades diferentes, a menos que utilice coeficientes arbitrarios (ej. si un costo es monetario y el otro social)

3.2.3.3 Condición de la Unidad de Planificación

Variable – ‘status’

Requerido: No

Descripción: Esta variable define si una unidad de planificación (PU, por sus siglas en inglés) está pre-incluida o pre-excluida en los sistemas de reservas iniciales. Puede tomar uno de cuatro valores:

Tabla 3: Valores de Unidad de Planificación

Estatus	Significado
0	No hay garantía de que la PU se encuentre en el sistema de reservas inicial (o semilla), sin embargo, pudiera suceder. La posibilidad de ser incluida en el sistema de reservas inicial está determinada por la ‘proporción de inicio’ especificada en el Archivo de Entrada de Parámetros (vea Sección 3.2.1).
1	La PU será incluida en el sistema inicial de reserva pero puede o no estar incluida en la solución definitiva.
2	La PU se fija en el sistema de reservas (“pre-incluida”). Comienza en el sistema inicial de reserva y no puede ser eliminada.
3	La PU se fija fuera del sistema de reservas (“pre-excluida”). No se incluye en el sistema de reservas inicial y no puede ser adicionada.

Iniciar: Esta variable no es necesaria y de no incluirse tomará el valor predeterminado de 0. En general, es de utilidad para ejecutar Marxan, por primera vez, sin sitios pre-incluidos o pre-excluidos, en aras de ofrecer una solución imparcial casi óptima. No obstante, esa variable puede resultar de utilidad para examinar diferentes escenarios donde las unidades de planificación tienen estatus '2' (siempre deben estar en el sistema de reservas), o estatus '3' (no pueden ser incluidas nunca en el sistema de reservas).

Como un ejemplo, a las unidades de planificación ubicadas en áreas protegidas se les puede asignar el estatus '2' ya que es muy poco probable que las zonas que ya están protegidas se intercambien por otras. Se puede desarrollar un análisis con todas las áreas protegidas pre-incluidas y pre-excluidas, para identificar hasta qué punto un sistema de áreas protegidas logra los objetivos ecológicos. Sin embargo, se debe ser cuidadoso al bloquear zonas dentro de una red de reservas ya que esto provoca una diferencia marcada en el carácter de las redes de reservas finales. En escenarios donde la compacidad de las reservas es importante (ej. se utiliza un modificador de longitud de frontera, vea Sección 3.2.1.1.2), es posible que Marxan utilice las zonas de conservación existentes como centros, o "semillas", alrededor de los cuales generar la solución. Esto provoca que se limite, de manera notable, la cantidad de soluciones posibles. Pudiera ser una opción deseada ya que, desde el punto de vista político y práctico, por lo general, resulta más fácil extender las zonas protegidas ya existentes que crear zonas nuevas, aunque ello pudiera conllevar también a soluciones de reserva ineficientes y posiblemente costosas. Asimismo, pudiera ser apropiado utilizar el estatus '2' para incidencias conocidas u objetos poco comunes o de un alto valor (ej. barreras de coral en aguas profundas); referente a los cuales sería una actitud irresponsable no incluirlos en un sistema de reservas, pero cuya inclusión en el proceso de selección regular de reserva puede parcializar el resultado sin razón. En una situación como esta, esos objetos deben ser bloqueados para no comprometer la transparencia del ejercicio de planificación o la capacidad para defenderlo.

Si entonces se hace un marcado énfasis en la compacidad como un escenario alternativo, Marxan debe ser ejecutado con la longitud de frontera de las unidades de planificación que contienen esos objetos pre-incluidos, igualada a cero. Esto va a minimizar su influencia en todo el sistema, y permitirá que se develen los sistemas más eficientes. Lo mismo pudiera aplicarse a los sitios de importancia cultural que los planificadores quisieran incluir en un sistema de reservas.

El status '3' es útil en casos donde las unidades de planificación nunca estarán disponibles para su inclusión en el sistema de reservas.

3.2.3.4 Ubicación de la Unidad de Planificación en el eje X

Variable – ‘xloc’

Requerido: No

Descripción: Representa la coordenada de eje-x de la unidad de planificación. Esta variable únicamente se requiere si se ha especificado una separación mínima entre las incidencias de objetos en la columna ‘sepdistance’ del **Archivo de Objeto de Conservación** (vea Sección 3.2.2.9). El valor registrado refleja una localización puntual para una unidad de planificación, que pudiera ser su centro o alguna otra elección acertada. Esta variable debe ser especificada en combinación con un valor para la variable ‘yloc’ (debajo).

En las instrucciones podrá encontrar más información referente a la generación de este valor (Apéndice C-2).

3.2.3.5 Ubicación de la Unidad de Planificación en el eje Y

Variable – ‘yloc’

Requerido: No

Descripción: Es la coordenada del eje-y de la unidad de planificación. Esta variable únicamente se requiere si se ha especificado una separación mínima entre las incidencias de objetos en la columna ‘sepdistance’ del **Archivo de Objetos de Conservación** (vea Sección 3.2.2.9). El valor registrado refleja una ubicación puntual para una unidad de planificación, que pudiera ser su centro o alguna otra elección acertada. Esta variable debe ser especificada en combinación con un valor para la variable ‘xloc’ (arriba).

En las instrucciones podrá encontrar más información referente a la generación de este valor (Apéndice C-2).

3.2.4 El Archivo de Unidades de Planificación versus Objetos de Conservación

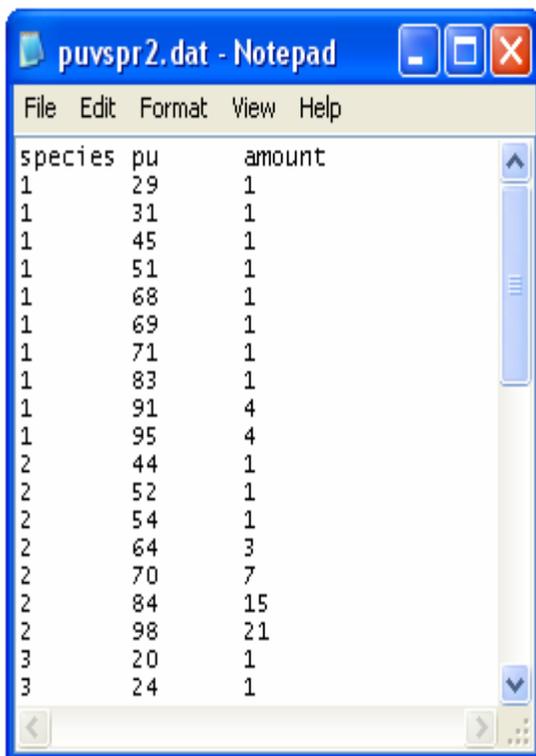
El Archivo de Unidades de Planificación versus Objetos de Conservación

contiene información referente a la distribución de los objetos de conservación en las unidades de planificación. Tiene como nombre de archivo predeterminado, 'puvpsr2.dat'. Este archivo puede tomar dos formatos diferentes, vertical y horizontal⁸. Cualquiera de ellos es aceptable y Marxan comprobará la línea de encabezamiento para determinar que formato está siendo utilizado.

En las instrucciones podrá encontrar más información referente a la generación de cualquier formato (Apéndice C-3).

3.2.4.1 Formato Vertical

Al utilizar el formato vertical, el **Archivo de Unidades de Planificación versus Objetos de Conservación** contiene tres columnas, todas las cuales son requeridas. El archivo comienza con una fila de encabezamiento que contiene el nombre de cada



species	pu	amount
1	29	1
1	31	1
1	45	1
1	51	1
1	68	1
1	69	1
1	71	1
1	83	1
1	91	4
1	95	4
2	44	1
2	52	1
2	54	1
2	64	3
2	70	7
2	84	15
2	98	21
3	20	1
3	24	1

una de las tres variables 'especies', 'pu.' y 'amount'. Cada fila posterior contiene un 'id' para un objeto de conservación (bajo el encabezamiento 'especies'), una identificación para una unidad de planificación (bajo el encabezamiento 'pu'), y un valor para la cantidad de ese objeto de conservación encontrada en la unidad de planificación (bajo el encabezamiento 'amount'). De esa manera, habrá una fila por cada vez que un objeto incida en una unidad de planificación. No hay valores predeterminados para este archivo, y de no incluirse la información necesaria o si se incluyen encabezamientos incorrectos Marxan no se ejecutará.

Un ejemplo de la forma vertical del **Archivo de Unidades de Planificación versus Objetos de Conservación** (puvspr2.dat) utilizado en Marxan.

⁸ En manuales previos de Marxan, esos formatos verticales y horizontales fueron denominados como "relacional" y "tabular," respectivamente.

3.2.4.1.1 ID del Objeto de conservación

Variable – ‘especies’

Requerido: Si

Descripción: Es un número de identificación único para cada objeto de conservación. Debe corresponderse a los números de ‘id’ utilizados en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2.1).

3.2.4.1.2 ID de la Unidad de Planificación

Variable – ‘pu.’

Requerido: Si

Descripción: Es el ‘id’ de cada unidad de planificación donde incide el objeto de conservación que aparece en la misma fila. Los números de ‘id’ de unidades de planificación deben corresponderse con los números utilizados en el **Archivo de Unidades de Planificación** (vea Sección 3.2.3.1).

3.2.4.1.3 Cantidad del Objeto de Conservación

Variable – ‘amount’

Requerido: Si

Descripción: Indica la cantidad del objeto de conservación que incide en la unidad de planificación que aparece en la misma fila. Esa cantidad se puede relacionar con la abundancia de una especie o la extensión de un tipo de hábitat determinado.

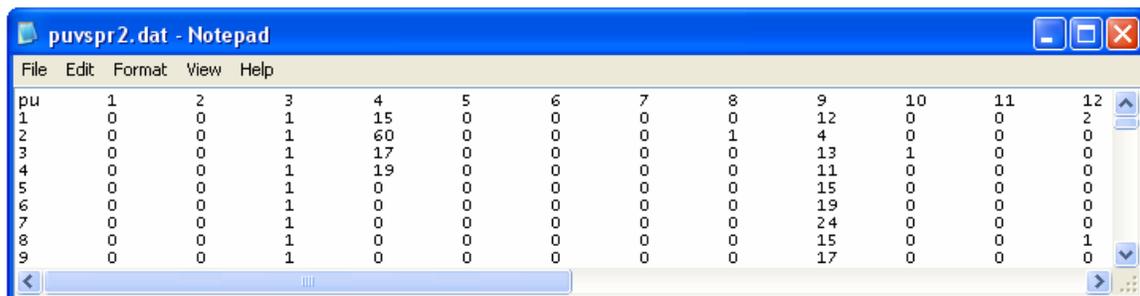
Iniciar: No se requiere utilizar el mismo sistema métrico para todos los objetos de conservación. Es esencial, sin embargo, que la cantidad para un objeto determinado se encuentre en las mismas unidades utilizadas para establecer la representación de las metas cuantitativas para ese objeto (vea Sección 3.2.2.3). En el formato de archivo vertical, no se deben listar casos en los que un objeto determinado no incide en una unidad de planificación, es decir, no debe existir una fila con una cantidad igual a ‘0’. En su lugar, toda la fila debe ser eliminada del archivo. Marxan asumirá que los objetos de conservación solo inciden en unidades de planificación donde se ha registrado una cantidad. El valor predeterminado para un par unidad de planificación/objeto de conservación que sea omitido en el archivo es cero.

Nota: No tiene importancia el orden que se le dé a la información en este archivo (ej. no tiene que ser secuencial para ninguna de las variables), sin embargo, el ordenamiento por objetos de conservación ayuda a garantizar que se ingrese la información para todos los objetos.

En las instrucciones se podrá encontrar más información referente a la generación de cualquier formato (Apéndice C-3).

3.2.4.2 Formato Horizontal

En el formato horizontal, el **Archivo de Unidades de Planificación versus Objetos de Conservación** no es más que una matriz de unidades de planificación versus objetos de conservación. La primera columna comienza con el encabezamiento, 'id', y es el listado de los números de identificación para cada unidad de planificación encontrado en el **Archivo de Unidades de Planificación** (vea Sección 3.2.3.1) Cada columna siguiente tiene un encabezamiento con el número de identificación de un objeto de conservación. Estos deben corresponderse con los números de identificación encontrados en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2.1) y debe haber una columna para cada objeto de conservación. Marxan asumirá que cualquier objeto de conservación que no figure como un encabezamiento de columna es porque no incide en ninguna unidad de planificación. En cada columna figurarán valores que indiquen la cantidad de cada objeto de conservación encontrada en esa unidad de planificación. Esto incluye entradas de '0' en casos donde una característica específica no se encuentre en esa unidad de planificación. Aunque esta sea quizás una forma más intuitiva de presentar la distribución de la información de los objetos de conservación, y sea más fácil de leer, resulta en grandes archivos con abundante información redundante. Como ocurre con el formato vertical, no existen valores predeterminados para este archivo, y de no incluirse la información necesaria, o si se incluyen encabezamientos incorrectos, Marxan no se ejecutará.



pu	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	15	0	0	0	0	12	0	0	2
2	0	0	1	60	0	0	0	1	4	0	0	0
3	0	0	1	17	0	0	0	0	13	1	0	0
4	0	0	1	19	0	0	0	0	11	0	0	0
5	0	0	1	0	0	0	0	0	15	0	0	0
6	0	0	1	0	0	0	0	0	19	0	0	0
7	0	0	1	0	0	0	0	0	24	0	0	0
8	0	0	1	0	0	0	0	0	15	0	0	1
9	0	0	1	0	0	0	0	0	17	0	0	0

Un ejemplo del formato horizontal del **Archivo de Unidad de Planificación versus Objeto de Conservación** (puvspr2.dat) utilizado en Marxan.

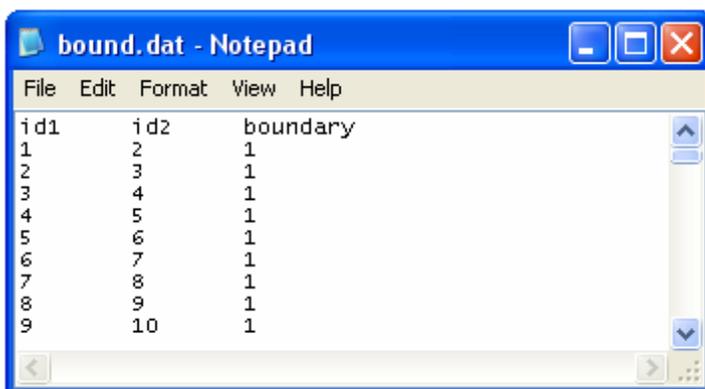
En las instrucciones podrá encontrar más información referente a la generación de cualquier formato (Apéndice C-3).

3.3 Archivos Opcionales

Los dos archivos siguientes son opcionales, si no se les da un nombre en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1), Marxan no intentará encontrarlos.

3.3.1 El Archivo de Longitud de Frontera

El **Archivo de Longitud de Frontera** contiene información referente a la longitud o 'longitud efectiva' de las fronteras compartidas entre unidades de planificación. Este archivo es necesario si se desea utilizar el Modificador de Longitud de Frontera (vea Sección 3.2.1.1.2) para mejorar la compacidad de las soluciones de reserva. No es necesario especificar las longitudes de fronteras para todas las unidades de planificación (donde no se especifiquen, Marxan asumirá que no existen fronteras entre las unidades de planificación). No obstante, la ausencia de cualquier valor dentro del archivo impedirá la ejecución del programa, es decir, si se establecen 'id1' e 'id2', pero no se ingresa valor para 'boundary'.



Un ejemplo del **Archivo de Longitud de Frontera** (bound.dat) utilizado en Marxan.

3.3.1.1 ID de las Unidades de Planificación

Variables – 'id1' e 'id2'

Requerido: SI

Descripción: 'id1' y 'id2' contiene el número de identificación de las dos unidades de planificación que comparten una frontera. Estas no tiene que ser unidades de planificación adyacentes, aunque por lo general lo son (para más detalles vea la información que se ofrece a continuación y el MGPH).

Iniciar: No tiene importancia el orden dado a los números de identificación, sin embargo, sí es de vital importancia no duplicar las fronteras ya que Marxan sumará las entradas duplicadas al calcular la longitud de frontera.

3.3.1.2 Longitud de Frontera

Variable – ‘boundary’

Requerido: Si



Descripción: El valor para la variable ‘boundary’ (frontera) puede derivarse en diferentes maneras, pero, es en esencia, una medida relativa de la importancia de incluir una unidad de planificación en el sistema de reservas, dada la inclusión de la otra unidad. Por ejemplo, si la unidad de planificación de la columna, ‘id1’, ha sido añadida al sistema de reservas, esta variable da la importancia de incluir también la unidad de planificación de la columna, ‘id2’, y viceversa. Al ser análoga a un ‘costo’ que debe pagarse si ambas unidades de planificación no son incluidas, generalmente nos referimos a esta variable como ‘boundary cost’ (costo de frontera).

Iniciar: En su aplicación más común, el costo de frontera refleja la longitud geográfica real de frontera entre dos unidades de planificación adyacentes, y ese es un buen lugar para comenzar. Sin embargo, ese costo puede ajustarse fácilmente para reflejar alguna otra asociación entre las unidades de planificación, por ejemplo, fronteras que son deseadas o no de manera particular. Como un ejemplo, puede no desearse contar con reservas que tengan fronteras adyacentes con zonas densamente pobladas, mientras que se desearía tener fronteras que colinden con reservas privadas u otras zonas de conservación. En ninguno de esos casos, esa información es reflejada en la longitud de frontera real entre las unidades de planificación cercanas entre sí.



Es muy importante que, si se utiliza alguna medida relativa diferente a la longitud de frontera real, el sistema métrico elegido debe estar bien argumentado. Recuerde que está introduciendo parcialidad al proceso de selección de reservas y todas las partes interesadas tienen derecho a saber el porqué. La transparencia y la “defensibilidad” son dos de los puntos fuertes de la planificación sistemática de la conservación. Incluso dos unidades de planificación que no sean adyacentes entre sí, pueden incurrir en un ‘costo de frontera’ si existe una importante relación entre ellas. Por ejemplo, si una especie requiere de un hábitat que se encuentra en diferentes unidades de planificación, ya sea en diferentes épocas del año o durante diferentes etapas de la vida, no tendría mucho sentido proteger una unidad y otra no. Si no se especifica relación alguna entre esas unidades de planificación, Marxan pudiera intercambiar una en lugar de la otra cuando en realidad se requiere de ambas para la persistencia de las especies. Ese método pudiera

utilizarse también para identificar vías de conectividad entre las unidades de planificación, por ejemplo, transportación larvaria en sistemas marinos, o flujos hidrológicos en los sistemas acuáticos.

En algunos casos, no será posible eliminar una frontera con sólo incluir en el sistema de reservas a una unidad de planificación cercana. Esto puede suceder, por ejemplo, en el borde de una jurisdicción territorial o una región de planificación obligatoria. Como las unidades de planificación ubicadas en el borde de una región, por lo general, tienen ‘fronteras compartidas’ más pequeñas, la selección puede ser parcializada hacia esas unidades de planificación. Esto pudiera ser un resultado no deseado. Para evitar esto, la selección de esas unidades de planificación, deben figurar en el **Archivo de Longitud de Frontera** como ‘fronteras inmóviles’. Esto puede lograrse especificando la longitud de frontera de una unidad de planificación con ella misma, por ejemplo, repitiendo la identificación de la misma unidad de planificación en las columnas ‘id1’ e ‘id2’.

El valor ingresado en ‘boundary’ siempre debe ser 0 o mayor. Aunque, por lo general, no es necesario especificar los casos donde no hay costo existente entre las unidades de planificación, pudiera ser útil una frontera de costo cero, si se quiere identificar a dos unidades de planificación como cercanas, pero no hay costo de frontera real. Eso pudiera ser necesario si se ha establecido tamaño mínimo de agrupamiento para algunos objetos de conservación (vea Sección 3.2.2.5).

En las instrucciones podrá encontrar más información referente a la generación de bound.dat (Apéndice C-5).

3.3.2 El Archivo de Definición de Bloques

El **Archivo de Definición de Bloques** es muy similar al **Archivo de Objetos de Conservación** (vea Sección 3.2.2). Se utiliza para establecer valores de variable predeterminados para grupos de objetos de conservación. Siempre se emplea en combinación con el **Archivo de Objetos de Conservación**. Los grupos de objetos de conservación deben definirse primero utilizando la variable, ‘type’, en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2.2). Los objetos pueden formar parte de un grupo porque los valores de algunas de sus variables son, por ejemplo, definidos por legislaciones específicas, o son similares en sus características ecológicas. En cada tipo diferente de objeto de conservación, los valores para todas las demás variables en el **Archivo de Objetos de Conservación** pueden definirse como **Archivo de Definición de Bloques**. Esto ofrece un método rápido y fácil para

la implementación de metas comunes en grupos de objetos, y es de utilidad en casos donde se utilizan diversas metas, y se exploran diferentes niveles de protección. Ofrece además una forma rápida de protección proporcional (porcentaje), sin necesidad de calcular esos valores de manera manual.



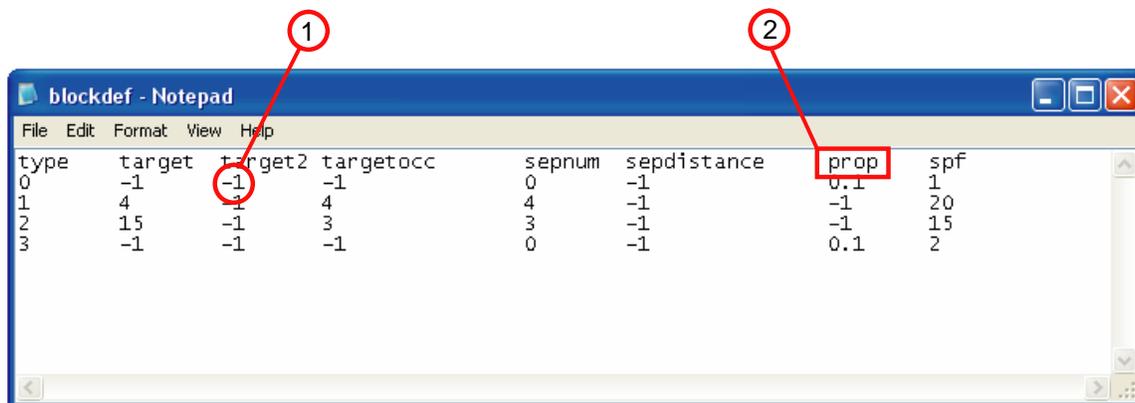
Para que los grupos de objetos adquieran los valores definidos en el **Archivo de Definición de Bloques**, el valor de la variable en el Archivo de Objetos de Conservación debe llevarse a '-1'.

id	type	target	spf	target2	targetocc	name	sepnum	sepdistance
1	1	20	-1	2	-1	A	-1	100000
2	1	1	-1	2	-1	B	-1	10
3	1	1	-1	3	-1	C	-1	10
4	1	3	-1	2	-1	D	-1	10
5	2	1	-1	0	-1	E	-1	10
6	2	1	-1	0	-1	F	-1	10
7	2	3	-1	0	-1	G	-1	10
8	2	3	-1	0	-1	H	-1	10
9	2	2	-1	0	-1	I	-1	10
10	2	10	-1	0	-1	J	-1	10
11	2	8	-1	0	-1	K	-1	10
12	3	20	-1	0	1	L	-1	10
13	3	15	-1	0	1	M	-1	10
14	3	31	-1	0	1	N	-1	10
15	3	27	-1	0	1	O	-1	10
16	3	29	-1	0	1	P	-1	10
17	0	59	-1	0	1	Q	-1	10
18	0	65	-1	0	1	R	-1	10
19	0	97	-1	0	1	S	-1	10
20	0	147	-1	0	1	T	-1	10
21	0	106	-1	0	1	U	-1	10
22	0	175	-1	0	1	V	-1	10
23	0	170	-1	0	1	W	-1	10
24	0	125	-1	0	1	X	-1	10

Un ejemplo de como deben ser establecidos los valores en el **Archivo de Objeto de Conservación** (spec.dat) si desea que esos parámetros tomen los valores definidos en el **Archivo de Definición de Bloques** (blockdef.dat). En ese caso, los 24 objetos de conservación tomarán el valor de Definición de Bloque para las variables 'spf' (Factor de Penalidad) y 'sepnum', y 11 objetos de conservación tomarán el valor de Definición de Bloque para la variable 'targetocc' (Meta para las Incidencias de Objetos Separados).

Esto es conveniente para permitir que algunas variables tomen su valor del grupo y otras sean establecidas de manera individual. Por ejemplo, en el caso anterior la variable 'target' se define por separado para cada objeto, quizás por la diferencia en la abundancia en la región de planificación. Por otra parte, en el mismo ejemplo, la variable 'spf', del **Factor de Penalidad del Objeto de Conservación** (vea Sección 3.2.2.4), siempre toma el valor definido en el **Archivo de Definición de Bloques**. Esto pudiera ser útil en casos donde los diferentes tipos de objetos de conservación tienen diferente grado de importancia pero no tenemos seguridad del valor 'spf' adecuado. Al utilizar el **Archivo de Definición de Bloques** se puede alterar el 'spf' para cada tipo de objeto sin necesidad de alterar cada entrada en el **Archivo de Objetos de Conservación**.

El **Archivo de Definición de Bloques** puede contener hasta ocho variables diferentes y debe estar en el mismo formato del **Archivo de Objetos de Conservación**.



Un ejemplo del **Archivo de Definición de Bloques** (blockdef.dat) utilizado en Marxan. 1. El negativo 1 debe ser registrado en este archivo si desea que la variable tome el valor original definido en el **Archivo de Objetos de Conservación**. 2. 'Prop' es la única variable nueva en este archivo—vea abajo la descripción.

La única variable requerida en este archivo es 'type', y la única variable nueva es 'prop'. Ambas se describen a continuación.

3.3.2.1 Tipo de Objeto de Conservación

Variable – 'type'

Requerida: Si

Descripción: Un identificador numérico único para grupos de objetos de conservación. Cada 'type' debe corresponderse exactamente con los tipos identificados en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2.2).

3.3.2.2. Meta Cuantitativa de Proporción para la Representación del Objeto

Variable – 'prop'

Requerido: No

Descripción: La variable 'prop' se puede utilizar para establecer la proporción (ej. porcentaje) de un objeto de conservación que será incluido en el sistema de reservas.

Iniciar: Debe ser un número entre 0 y 1. Por ejemplo, si 'prop' para un tipo de objeto se lleva a 0.2, entonces Marxan establecerá la meta cuantitativa para todos los objetos de ese tipo con 20% de la abundancia total, basado en la información del

Archivo de Unidades de Planificación versus Objetos de Conservación (vea Sección 3.2.4). La abundancia total es la suma de la cantidad encontrada en todas las unidades de planificación, incluidas aquellas que puedan ser pre-incluidas y pre-excluidas de las posibles soluciones de reservas. Como 'prop' establece un valor de meta, donde se utilice, el valor para la variable 'target' se debe llevar a '-1'. No tiene sentido establecer ambas variables y si se establecen ambas, entonces la variable 'prop' tomará precedencia y la variable 'target' será ignorada.



La proporción se basa en la cantidad total definida en el **Archivo de Unidades de Planificación versus Objetos de Conservación**. Si ocurre un pre-procesamiento de la información, por ejemplo, la eliminación de incidencias pequeñas o fragmentadas de un objeto, pudiera entonces no ser la misma que se ha indicado en el conjunto de datos original.

3.3.2.3 Todas las variables restantes

Las variables restantes en el **Archivo de Definición de Bloques** ('target', 'target2', 'targetocc', 'sepnun', 'sepdistance' y 'spf'), tienen todas la misma definición que en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2) y cuando no se defina el valor aquí, tomarán el valor de ese archivo. Si se desea que alguna de esas seis variables definidas tome el valor original del **Archivo de Objetos de Conservación**, sencillamente lleve el valor en el **Archivo de Definición de Bloques** a '-1'. Ese es también el valor predeterminado para esas variables y la entrada que no se registre tomará ese valor.

(Esta página está en blanco intencionalmente)

4. Ejecución del Software

Ejecutar Marxan es extremadamente simple. Una vez que estén listos los archivos de entrada, todo lo que tiene que hacer es doble clic en el archivo 'Marxan.exe' y el programa comenzará automáticamente. No obstante, para que se ejecute satisfactoriamente, se debe crear la carpeta que contiene el programa de manera que Marxan pueda encontrar los archivos requeridos y guardar las salidas necesarias (vea Sección 2.2).

Si Marxan se ejecuta satisfactoriamente, se visualizará una pantalla de programa mostrando la información sobre los detalles y el progreso de cada ejecución (a menos que se seleccione la Ejecución Silenciosa—vea Sección 3.2.1.5.1). No se preocupe si avanza con tanta rapidez que no lo pueda leer, todos los detalles necesarios se guardarán en la carpeta de archivo de salida en un archivo denominado 'log.dat'. Marxan se detendrá cuando complete la cantidad de ejecuciones pre-establecidas, aunque la pantalla del programa permanecerá visible. Para salir del programa y cerrar la ventana presione 'Enter'. Si Marxan se cierra antes de tiempo o se detiene con un mensaje de error, es probable que haya un problema con el formato de uno o más archivos de entrada (vea Apéndice A para la guía a la solución de problemas).

(Esta página está en blanco intencionalmente)

5. Resultados

Además de indicar cuales son las unidades de planificación que constituirán un sistema de reservas eficiente, Marxan también puede brindar una variedad de resultados, que son descritos posteriormente. Todos los archivos de salida de Marxan se pueden ver con programas básicos de edición de texto como Windows Notepad, o con un programa de hoja de cálculo y base de datos como Microsoft Excel y Access. Los resultados específicos que usted desea que Marxan guarde deben quedar especificados en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1).

Aunque es común que las soluciones generadas por Marxan se muestren gráficamente, Marxan no generará mapas. El usuario debe combinar la información de los archivos de salida de Marxan con las unidades de planificación mostradas en un Sistema de Información Geográfica (SIG). En el Apéndice C se brinda una explicación sobre como hacerlo. Si se va a importar información directamente desde Marxan hacia un SIG, pudiera resultar de utilidad seleccionar la opción 'ArcView Format' brindada en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1.5.2).

5.1 Manejo del Archivo de Salida

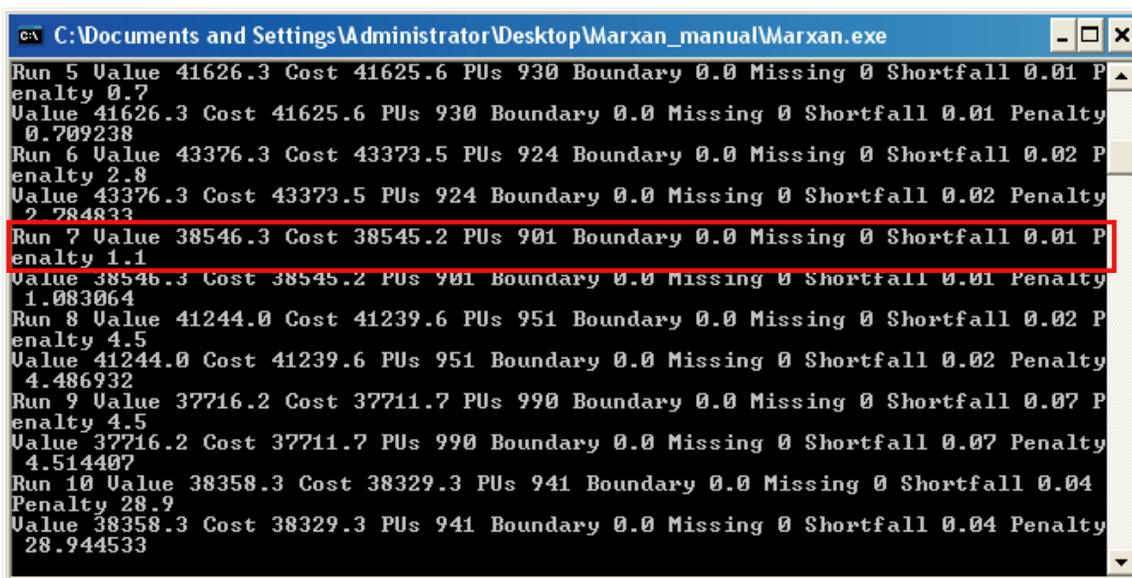
Marxan guardará los archivos de salida en una carpeta cuyo nombre se puede especificar en el **Archivo de Entrada de Parámetros**. Sugerimos utilizar el nombre de carpeta predeterminado 'output' (salida) y luego cambiar el nombre del archivo una vez que se haya completado la ejecución. Debe asegurarse de que el directorio de salida que se especificó haya sido creado antes de ejecutar Marxan. Esto lo ayudará a garantizar que no escriba encima de la información de salida cuando comience una nueva ejecución de Marxan. Como sucede con la carpeta de archivo de entrada, es conveniente separar las carpetas de archivos de salida para cada escenario (ej. dentro de la misma carpeta que el ejecutable de Marxan. Vea Sección 3.1.2). Se recomienda también guardar una copia del archivo 'input.dat' en la misma carpeta, así siempre sabrá que parámetros se utilizaron para generar esos resultados. Contar con ese conocimiento ayudará enormemente a mejorar los escenarios futuros y establecer los parámetros adecuados.

5.2 Resultados en pantalla

A medida que se ejecuta Marxan, algunos resultados se pueden mostrar en la pantalla. Ese resultado es útil para verificar cómo se está ejecutando el programa y dar un pequeño resumen de las soluciones. Ello proporcionará rápidamente (o no, según sea el caso) una idea de cuanto tiempo toma una sola ejecución, y de ahí, podrá calcular con bastante precisión el tiempo que tomará completar todas las ejecuciones para ese escenario. Hay varios niveles de información que pueden solicitarse a los resultados en pantalla, el nivel solicitado se denomina nivel de verbosidad (Vea Sección 3.2.1.5.1).

5.2.1 Resultados Básicos

Si la verbosidad se establece en cualquier opción que no sea 'no output', entonces el siguiente resumen básico de cada ejecución aparecerá en la pantalla. Si se selecciona 'Results Only' (modo de verbosidad 1), esa es la única información que será suministrada.



```
C:\Documents and Settings\Administrator\Desktop\Marxan_manual\Marxan.exe
Run 5 Value 41626.3 Cost 41625.6 PUs 930 Boundary 0.0 Missing 0 Shortfall 0.01 Penalty 0.7
Value 41626.3 Cost 41625.6 PUs 930 Boundary 0.0 Missing 0 Shortfall 0.01 Penalty
0.709238
Run 6 Value 43376.3 Cost 43373.5 PUs 924 Boundary 0.0 Missing 0 Shortfall 0.02 Penalty 2.8
Value 43376.3 Cost 43373.5 PUs 924 Boundary 0.0 Missing 0 Shortfall 0.02 Penalty
2.784833
Run 7 Value 38546.3 Cost 38545.2 PUs 901 Boundary 0.0 Missing 0 Shortfall 0.01 Penalty 1.1
Value 38546.3 Cost 38545.2 PUs 901 Boundary 0.0 Missing 0 Shortfall 0.01 Penalty
1.083064
Run 8 Value 41244.0 Cost 41239.6 PUs 951 Boundary 0.0 Missing 0 Shortfall 0.02 Penalty 4.5
Value 41244.0 Cost 41239.6 PUs 951 Boundary 0.0 Missing 0 Shortfall 0.02 Penalty
4.486932
Run 9 Value 37716.2 Cost 37711.7 PUs 990 Boundary 0.0 Missing 0 Shortfall 0.07 Penalty 4.5
Value 37716.2 Cost 37711.7 PUs 990 Boundary 0.0 Missing 0 Shortfall 0.07 Penalty
4.514407
Run 10 Value 38358.3 Cost 38329.3 PUs 941 Boundary 0.0 Missing 0 Shortfall 0.04 Penalty 28.9
Value 38358.3 Cost 38329.3 PUs 941 Boundary 0.0 Missing 0 Shortfall 0.04 Penalty
28.944533
```

Ejemplo del resumen en pantalla utilizando el modo de resultado en pantalla ('Results Only').

Nota: La misma información se ofrece en el archivo de salida 'información de resumen' (vea Sección 5.3) de modo que no tenga que mirar a los resultados en pantalla después de terminada la ejecución de Marxan.

5.2.1.1 Run (Corrida o ejecución)

Descripción: Expresa a cuál de las ejecuciones repetidas (vea Sección 3.2.1.1.1) pertenece el resultado. En dependencia del nivel de los detalles mostrados en pantalla, el número de la ejecución no siempre aparecerá en la misma línea que los detalles de la solución de reservas (vea 5.2.2) La información de cada ejecución será mostrada tan pronto como se complete la ejecución.

5.2.1.2 Value (Valor)

Descripción: Este es el valor general de la función objetivo para la solución de esa ejecución. Incluye no sólo el costo de las unidades de planificación y la longitud de frontera sino las penalidades por no representar adecuadamente todos los objetos de conservación o por exceder el umbral de costo (vea Sección 1.5). Resulta útil conocer ese valor ya que esa es la forma en que Marxan escoge la ‘mejor’ solución a partir de las repeticiones de las ejecuciones.

5.2.1.3 Cost (Costo)

Descripción: Es el costo total del sistema de reservas determinado únicamente por los costos dados en las unidades de planificación (vea Sección 3.2.3.2)

5.2.1.4 PUs (unidades de planificación)

Descripción: La cantidad de unidades de planificación contenidas en la solución para esa ejecución.

5.2.1.5 Boundary (Frontera)

Descripción: Longitud total de frontera de un sistema de reservas (vea Sección 3.3.1.2). Si las longitudes de fronteras no son tomadas en cuenta en los análisis (no se ofrece un **Archivo de Longitud de Frontera**), entonces el valor se leerá ‘0.0’.

5.2.1.6 Missing (No alcanzado)

Descripción: Cantidad de objetos de conservación que no cumplieron su meta en la solución final para esa ejecución. Serán mostrados según el umbral (‘miss level’) definido en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1.5.4). Si la variable ‘miss level’ se lleva a 1, entonces cada objeto de conservación que se registre por debajo de su nivel de meta cuantitativa será computado como no alcanzado. Si el nivel de objeto no alcanzado se lleva por debajo de 1 (ej. 0.98), puede suceder que Marxan no reporte algún objeto como no alcanzado incluso cuando el sistema de reservas contenga un poco menos que la cantidad predeterminada.

5.2.1.7 Shortfall (Déficit)

Descripción: Cantidad por la cual las metas de los objetos de conservación no se cumplieron en la solución para esa ejecución. El déficit reportado aquí, es el déficit total sumado en todos los objetos de conservación. El déficit es un buen indicador para analizar si los objetos de conservación no alcanzados están muy cercanos o muy lejanos de sus metas. Si hay una cierta cantidad de objetos de conservación que no han cumplido su meta, pero el déficit combinado es muy pequeño, entonces el planificador no tiene que preocuparse.

5.2.1.8 Penalty (Penalidad)

Descripción: Penalidad que fue adicionada a la función objetivo porque el sistema de reservas no cumplió las metas de representación para todos los objetos de conservación. Si se presentan todas los objetos adecuadamente, entonces el valor de penalidad será 0.0 o "-0.0". (Debido al error de redondeo, no es posible que sea exactamente igual a 0, pero sólo con presentar un lugar decimal el error de redondeo posiblemente quedará oculto) El modo de calcular esta penalidad se describe en detalles en el Apéndice B. Es de utilidad que se conozca la penalidad para tener una idea del costo requerido para cumplir las metas restantes, esto es algo que no se logra únicamente analizando el déficit. Es además otra manera de evaluar la efectividad de las ejecuciones, analizando sólo aquellas soluciones que tienen una baja penalidad.

5.2.2 Progreso General

Si la verbosidad de los resultados en pantalla se establece en 'Progreso' General (modo verboso 2), o 'Progreso Detallado' (modo verboso 3) entonces, además de los resultados básicos explicados, se ofrecen los resúmenes siguientes:

```
C:\Documents and Settings\Administrator\Desktop\Marxan_manual\Marxan.exe
Entering in the data files
  There are 3605 Planning units.
  3605 Planning Unit names read in
  30 species read in
  0 boundaries entered
  3682 conservation values read
  Time passed so far is 0 secs

Pre-processing Section.

Run 1 Using Calculated Tinit = 7261.9831 Tcool = 0.99816548
  Creating the initial reserve

  InitValue 458140.9 Cost 0.0 PUs 0 Boundary 0.0 Missing 30 Shortfall 3453.10 Penalty 458140.9
  Value 458140.9 Cost 0.0 PUs 0 Boundary 0.0 Missing 30 Shortfall 3453.10 Penalty 458140.900000
  Annealing Completed Time passed so far is 1 secs
  Value 40138.4 Cost 40138.2 PUs 921 Boundary 0.0 Missing 0 Shortfall 0.00 Penalty 0.2
  Value 40138.4 Cost 40138.2 PUs 921 Boundary 0.0 Missing 0 Shortfall 0.00 Penalty 0.169283
  Best:Value 40138.4 Cost 40138.2 PUs 921 Boundary 0.0 Missing 0 Shortfall 0.00 Penalty 0.2
```

1. Detalles sobre la información que se ingresa.
2. Número de la ejecución y parámetros de templado calculados durante el pre-procesamiento (Si se utiliza el templado adaptable).
3. Los detalles del sistema de reservas inicial o semilla.
4. Los resultados de la solución de reserva de esa ejecución (como se describe arriba).
5. Este resultado es utilizado por Marxan de manera que ordene rápidamente todas las ejecuciones para determinar cuál dio la mejor solución.

Ese nivel de información puede ser particularmente útil por varias razones. Primeramente, si está ejecutando más de un procedimiento de optimización (ej. templado simulado seguido de mejoramiento iterativo, vea Sección 3.2.1.2.1), esto le permite entonces tener una idea referente a la cantidad de trabajo que realiza cada uno de los diferentes procedimientos. Por ejemplo, si la mayoría de las ganancias en los valores de reserva y el cumplimiento de las metas se realiza en la fase de mejoramiento iterativo, se sabe que los parámetros de templado o las penalidades necesitan ser cambiadas. En segundo lugar, si se desea comenzar utilizando un esquema de templado fijo, esta salida puede dar una idea del tipo de valores que Marxan está calculando empleando su módulo de templado adaptable (vea Sección 3.2.1.3.1) Finalmente, si se han establecido algunas limitaciones en el sistema de reservas inicial, por ejemplo, al incluir las áreas protegidas existentes, este resultado

confirmará rápidamente que esa información está siendo incluida, así como los valores de las reservas existentes en términos de cumplir los objetivos. Si está usando esa salida para cualquiera de esos objetivos, va a desear salvar los resultados en pantalla de modo que los pueda revisar una vez que Marxan haya concluido su ejecución (vea Sección 5.3).

5.2.3 Progreso Detallado

Si se desea, Marxan puede suministrar información bien detallada en la pantalla (modo verboso 3). Esa información muestra con exactitud el estado del programa durante cada ejecución y como va progresando el templado. En términos generales, permite confirmar que el algoritmo se está ejecutando como debe, y no se ha “paralizado.” En el Apéndice B se explican con mayor profundidad los detalles de este resultado.

5.3 Archivos de Salida

Marxan puede guardar hasta ocho archivos diferentes de salida, en dependencia de lo que fue especificado en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1.5.2).

Tabla 4: Tipos y nombres de archivos de salida.

Tipo de Archivo de Salida	Nombre de Archivo
Soluciones para cada ejecución	<i>scenario_r001.dat</i>
Mejor Solución de todas las ejecuciones	<i>scenario_best.dat</i>
Información del valor no alcanzado en cada ejecución	<i>scenario_mv001.dat</i>
Información del valor no alcanzado en la mejor ejecución	<i>scenario_mvbest.dat</i>
Información de resumen	<i>scenario_sum.dat</i>
Detalles del escenario	<i>scenario_sen.dat</i>
Solución sumada	<i>scenario_ssoln.dat</i>
Archivo de registro en pantalla	<i>scenario_log.dat</i>
Archivos de muestras instantáneas	<i>scenario_snap_r00001t01000.dat</i>

El archivo con el prefijo ‘*scenario*’ tomará cualquier nombre especificado por el usuario de la variable ‘SCENNAME’ en el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1.5.2). En el caso donde se incluya un número en el nombre del archivo (ej. *scenario_r001.dat*), ese es el número de ejecución que generó ese resultado particular.

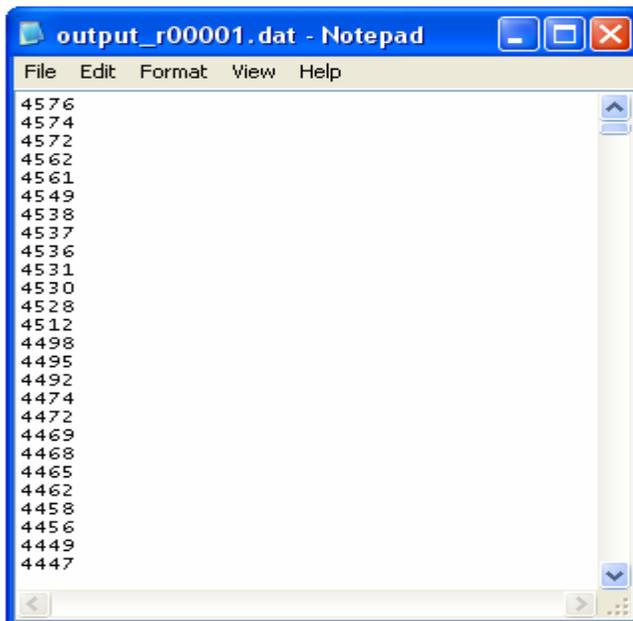
5.3.1 Formato de los Archivos de Salida

El formato estándar de salida es un archivo de texto separado por tabulador con la extensión '.dat'. Si se selecciona el formato ArcView dentro de **Inedit** o el **Archivo de Entrada de Parámetros** (vea Sección 3.2.1.5.2), entonces los archivos de información y el archivo de registro en pantalla serán separados por comas, y los encabezamientos se darán en el formato necesario para la importación por programas de SIG como ArcView. La extensión para los archivos de formato ArcView es '.txt'.

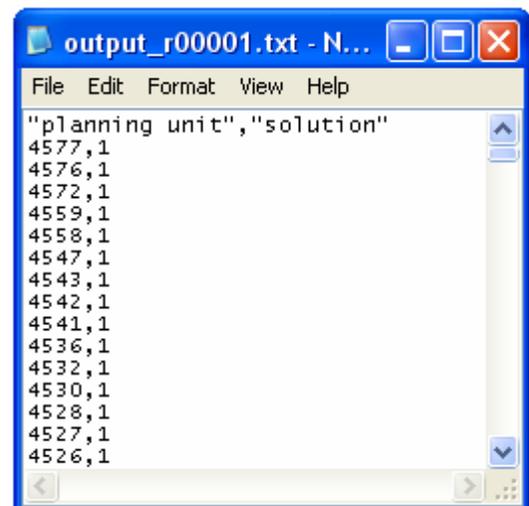
5.3.2 Soluciones para cada ejecución

Nombre del Archivo: escenario_r001.dat

Descripción: Se produce un archivo para cada repetición de ejecución, que contiene un listado de todas las unidades de planificación seleccionadas en la solución para esa ejecución. El número de ejecución se indica por "_r001", por ejemplo; ejecución 1. Las unidades de planificación seleccionadas son identificadas por sus números de identificación (vea Sección 3.2.3.1) y cada una aparece en una línea diferente. Si se especifica el formato de ArcView (.txt), el archivo tendrá también los encabezamientos, "unidad de planificación" y "solución", y cada línea tendrá un número de unidad de planificación seguido de un 1, separado por comas. Este archivo permite mostrar con facilidad la solución del sistema de reservas para cada ejecución, sin embargo, como casi siempre se realizan múltiples repeticiones de ejecuciones, un archivo útil será el archivo de 'solución sumada' (vea Sección 5.3.7).



```
output_r00001.dat - Notepad
File Edit Format View Help
4576
4574
4572
4562
4561
4549
4538
4537
4536
4531
4530
4528
4512
4498
4495
4492
4474
4472
4469
4468
4465
4462
4458
4456
4449
4447
```



```
output_r00001.txt - N...
File Edit Format View Help
"planning unit","solution"
4577,1
4576,1
4572,1
4559,1
4558,1
4547,1
4543,1
4542,1
4541,1
4536,1
4532,1
4530,1
4528,1
4527,1
4526,1
```

Ejemplo del archivo de solución para la ejecución número 1. Formato ArcView.

Ejemplo de archivo para la ejecución número 1-
formato estándar.

5.3.3 Mejor solución de todas las ejecuciones

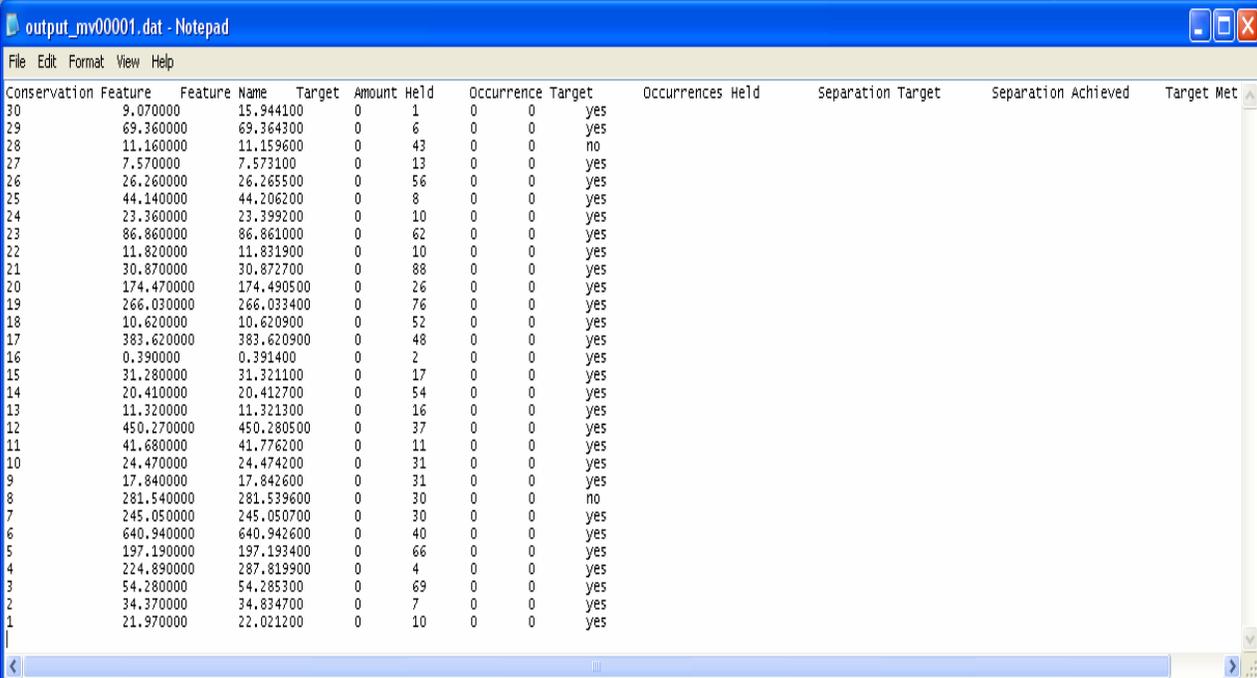
File name: scenario_best.dat

Descripción: Exactamente como la anterior, con excepción de la ejecución que generó el mejor valor objetivo. Tenga cuidado: la solución de la ‘mejor’ ejecución solamente es superior con relación al valor de la función objetivo, en realidad esto no la convierte en el mejor sistema de reservas. Asimismo, la mejor solución puede ser mejor que las demás soluciones sólo por un estrecho margen. De modo que “mejor” aquí tiene un significado bastante limitado y no debe comunicarse a las partes interesadas o los encargados de la toma de decisiones como la solución ideal. En su lugar, debe ser vista como una muy buena solución dentro de una variedad de opciones. En el MGPB encontrará más análisis sobre este tema.

5.3.4 Valores no alcanzados para cada ejecución

File name: scenario_mv001.dat

Descripción: Este archivo contiene información referente a la representación de los objetos de conservación en la solución de cada ejecución. El archivo contiene un total de nueve columnas que básicamente reportan cómo operó la solución con relación a las metas. Algunas son sencillamente un resumen de la información suministrada en el **Archivo de Objetos de Conservación**



Conservation	Feature	Feature Name	Target	Amount	Held	Occurrence	Target	Occurrences Held	Separation Target	Separation Achieved	Target Met
30	9.070000	15.944100	0	1	0	0	yes				
29	69.360000	69.364300	0	6	0	0	yes				
28	11.160000	11.159600	0	43	0	0	no				
27	7.570000	7.573100	0	13	0	0	yes				
26	26.260000	26.265500	0	56	0	0	yes				
25	44.140000	44.206200	0	8	0	0	yes				
24	23.360000	23.399200	0	10	0	0	yes				
23	86.860000	86.861000	0	62	0	0	yes				
22	11.820000	11.831900	0	10	0	0	yes				
21	30.870000	30.872700	0	88	0	0	yes				
20	174.470000	174.490500	0	26	0	0	yes				
19	266.030000	266.033400	0	76	0	0	yes				
18	10.620000	10.620900	0	52	0	0	yes				
17	383.620000	383.620900	0	48	0	0	yes				
16	0.390000	0.391400	0	2	0	0	yes				
15	31.280000	31.321100	0	17	0	0	yes				
14	20.410000	20.412700	0	54	0	0	yes				
13	11.320000	11.321300	0	16	0	0	yes				
12	450.270000	450.280500	0	37	0	0	yes				
11	41.680000	41.776200	0	11	0	0	yes				
10	24.470000	24.474200	0	31	0	0	yes				
9	17.840000	17.842600	0	31	0	0	yes				
8	281.540000	281.539600	0	30	0	0	no				
7	245.050000	245.050700	0	30	0	0	yes				
6	640.940000	640.942600	0	40	0	0	yes				
5	197.190000	197.193400	0	66	0	0	yes				
4	224.890000	287.819900	0	4	0	0	yes				
3	54.280000	54.285300	0	69	0	0	yes				
2	34.370000	34.834700	0	7	0	0	yes				
1	21.970000	22.021200	0	10	0	0	yes				

Ejemplo del archivo con el valor no alcanzado para la ejecución 1. Nota: los encabezamientos de columnas y la información no siempre estarán alineadas debido al requerimiento de la separación por tabulador entre los encabezamientos. Esta inconveniencia visual se puede solucionar importando el resultado a un programa de hoja de cálculo como MS Excel.

5.3.4.1 Conservation Feature (Objeto de Conservación)

Descripción: El ID único del objeto de conservación (vea Sección 3.2.2.1).

5.3.4.2 Feature Name (Nombre del Objeto)

Descripción: Nombre alfabético opcional del objeto de conservación (vea Sección 3.2.2.7). Si no se ha especificado nombre alguno, entonces no aparecerá nada en esa columna.

5.3.4.3 Target (Meta Cuantitativa)

Descripción: El nivel de meta de representación (si hay alguno) para es objeto de conservación (vea Sección 3.2.2.3).

5.3.4.4 Amount Held (Cantidad Incluida)

Descripción: Cantidad de ese objeto de conservación incluida en el sistema de reservas. Se incluyen únicamente las cantidades en agrupamientos válidos (vea Sección 3.2.2.5).

5.3.4.5 Occurrence Target (Meta Cuantitativa de Incidencia)

Descripción: Meta cuantitativa para las incidencias de ese objeto de conservación en el sistema de reservas (vea Sección 3.2.2.6).

5.3.4.6 Occurrence Held (Incidencias Incluidas)

Descripción: La cantidad de incidencias del objeto de conservación incluidas en la solución. Una vez más, se incluirán únicamente las incidencias en agrupamientos válidos.

5.3.4.7 Separation Target (Meta Cuantitativa de Separación)

Descripción: Cantidad de incidencias, separadas mutuamente y de manera adecuada, de ese objeto de conservación requerido en el sistema de reservas (vea Secciones 3.2.2.8 y 3.2.2.9).

5.3.4.8 Separation Achieved (Separación Alcanzada)

Descripción: La cantidad reportada aquí será la más baja, o bien de la cantidad de incidencias separadas que se lograron realmente en el sistema de reservas; o de la meta de incidencias separadas. El conteo de separación (vea Apéndice B-1.3.1)

nunca excede la meta de separación para ese objeto. Esta es una convención que acelera la ejecución del programa pero significa que no se ha suministrado información alguna sobre en cuánto se ha sido excedido esa meta.

5.3.4.9 Target Met (Meta Cuantitativa Cumplida)

Descripción: Variable alfabética que retorna 'si' si se cumplen todos los objetivos para ese objeto, de lo contrario retorna 'no'.

5.3.5 Información sobre el valor no alcanzado en la mejor ejecución

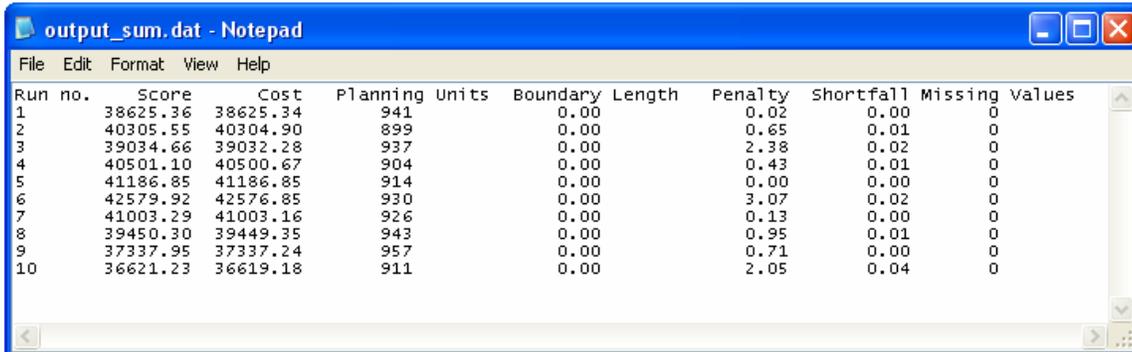
File name: escenario_mvbest.dat

Descripción: Exactamente como el anterior excepto que se reporta sólo para la ejecución que generó la solución con el mejor valor objetivo.

5.3.6 Información de Resumen

File name: escenario_sum.dat

Descripción: Ese archivo contiene la información de resumen para cada repetición de ejecución. Es exactamente la misma información suministrada en la salida 'en pantalla', que se describe en la Sección 5.2.1.



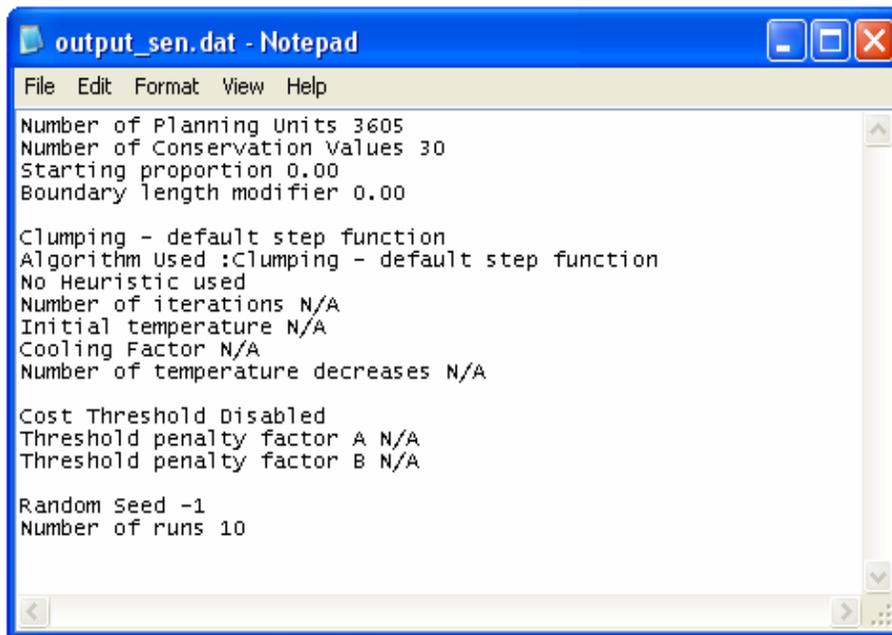
Run no.	Score	Cost	Planning Units	Boundary Length	Penalty	Shortfall	Missing	Values
1	38625.36	38625.34	941	0.00	0.02	0.00	0	
2	40305.55	40304.90	899	0.00	0.65	0.01	0	
3	39034.66	39032.28	937	0.00	2.38	0.02	0	
4	40501.10	40500.67	904	0.00	0.43	0.01	0	
5	41186.85	41186.85	914	0.00	0.00	0.00	0	
6	42579.92	42576.85	930	0.00	3.07	0.02	0	
7	41003.29	41003.16	926	0.00	0.13	0.00	0	
8	39450.30	39449.35	943	0.00	0.95	0.01	0	
9	37337.95	37337.24	957	0.00	0.71	0.00	0	
10	36621.23	36619.18	911	0.00	2.05	0.04	0	

Ejemplo del archivo de salida de resumen.

5.3.7 Detalles de Escenario

File name: scenario_sen.dat

Descripción: Este archivo contiene un listado argumentado de todos los valores de parámetros más importantes para ese escenario. Este archivo es de mucha utilidad



```
output_sen.dat - Notepad
File Edit Format View Help
Number of Planning Units 3605
Number of Conservation Values 30
Starting proportion 0.00
Boundary length modifier 0.00

Clumping - default step function
Algorithm Used :Clumping - default step function
No Heuristic used
Number of iterations N/A
Initial temperature N/A
Cooling Factor N/A
Number of temperature decreases N/A

Cost Threshold Disabled
Threshold penalty factor A N/A
Threshold penalty factor B N/A

Random Seed -1
Number of runs 10
```

para registrar los parámetros que producen determinados resultados, especialmente, cuando se ejecutan escenarios múltiples. Esa información es necesaria para ayudar a seleccionar los valores apropiados para parámetros comúnmente modificados.

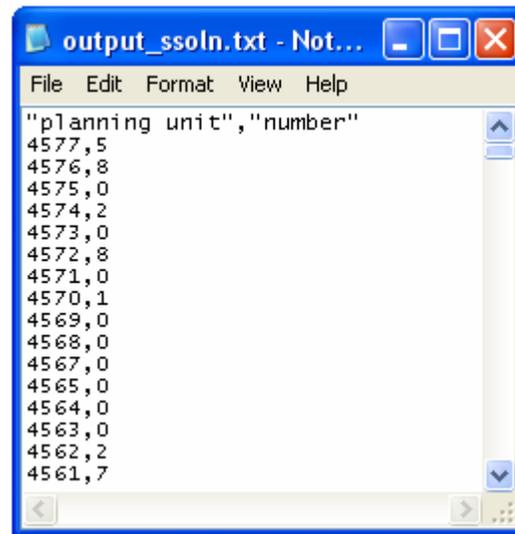
Ejemplo del archivo de resultado de detalles del escenario

5.3.8 Solución Sumada

File name: escenario_ssoln.dat

Descripción: La solución sumada ofrece la frecuencia de selección de cada unidad de planificación en todas las ejecuciones. Cada línea tiene el ID de una unidad de planificación (vea Sección 3.2.3.1) y las veces que esa unidad de planificación fue seleccionada en la solución final entre todas las repeticiones de ejecuciones.⁹ De los

archivos de salida de Marxan este es quizás el más utilizado y con seguridad el que más se muestra. Brinda una medida de la utilidad de cada unidad de planificación para la creación de un sistema de reservas eficiente. Más que como un mapa de resultado de solución sumada, puede verse como un mapa de prioridad de conservación. Esas unidades de planificación que comúnmente se seleccionan en la solución de reserva final (ej. > 70%) tiene probabilidades de ser requeridas para un sistema de reservas eficiente. Las unidades de planificación que se seleccionan con poca frecuencia son de menor urgencia, ya que los objetos de conservación que contienen pueden ser adquiridos en otras locaciones. Sin embargo, note que incluso si una unidad de planificación es seleccionada en casi todas las soluciones, no significa que no podrá obtener los mismos objetos en otros lugares. Esto sólo significa que esa unidad de planificación ayuda a proporcionar soluciones eficientes. En realidad, esto puede no guardar mucha relación con los objetos de conservación existentes en el sitio y sí estar más relacionado con el costo o la ubicación de la unidad de planificación.



Ejemplo del archivo de solución sumada – ArcView formato.

⁹ Esa medida se denominaba 'irreemplazabilidad' aunque 'Frecuencia de Selección' es una descripción mas adecuada. Este concepto se inspira en la noción de Bob Pressey de la irreemplazabilidad, aunque difiere de dicha noción (Pressey et al. 1994). Vea Carwardine et al. (2006) para una comparación entre irreemplazabilidad y frecuencia de selección.

5.3.9 Archivo de Registro en Pantalla

File name: scenario_log.dat

Descripción: Es un archivo de texto que contiene exactamente lo que Marxan mostró como resultado en pantalla para ese escenario (vea Sección 5.1). Esto puede ser de utilidad en la depuración de errores, ó, si por ejemplo, se quiere regresar a las ejecuciones para examinar la cantidad de trabajo realizado en la fase de templado en comparación con el mejoramiento iterativo (vea Sección 5.2.2).

5.3.10 Archivo de muestra instantánea

File name: scenario_snap_r00001t01000.dat (para la muestra instantánea número 1000 en la 1ra ejecución)

Descripción: Los archivos de salidas de muestras instantáneas presentan el progreso de la solución en las etapas del procedimiento de optimización. La solución actual se guarda en un número predeterminado de iteraciones del sistema o cambios del sistema (vea Sección 3.2.1.5.2). Se guarda en el mismo formato de la solución final para cada ejecución. Esos archivos permiten que el usuario analice el progreso de un método de solución. Se necesita sólo en análisis avanzados para examinar el procedimiento de templado bajo diferentes valores de parámetros. Por lo general, no se recomienda.

(Esta página está en blanco intencionalmente)

6. Obtención de Buenos Resultados

En este manual se debe ofrecer toda la información que usted necesita para hacer un uso correcto y eficaz de Marxan. Asegurar que los análisis hechos por Marxan sean precisos y bien argumentados no es parte del contenido de este documento. Esta información se recoge en el MGPH. Una vez que usted esté familiarizado con el funcionamiento del programa, lo exhortamos a leer el folleto sobre las buenas prácticas antes de acometer análisis más complejos. En esta sección sólo hemos mencionado algunos de los aspectos que se deben conocer en aras de garantizar que Marxan genere resultados irrefutables con calidad.

6.1 Experimentación

Lo más importante a tener en cuenta es que Marxan no ofrece soluciones de reserva 'expeditas'. Como se resalta en la Sección 3 de este manual, muchos de los parámetros requerirán de mucha experimentación antes de que Marxan sea capaz de generar soluciones razonables (ej. que los objetivos de conservación se cumplan, y que cada uno de los restantes objetivos, como el nivel de agrupamiento, se cumplan hasta un nivel aceptable). Para que cada parámetro se establezca de manera ideal, debe hacerse paso a paso y de forma sistemática. No obstante, esto pudiera resultar difícil ya que los parámetros no son independientes entre ellos. Los parámetros pueden interactuar con frecuencia de maneras inesperadas, por ejemplo, el BML óptimo puede variar de manera notable si se modifica el SPF. Asimismo, los cambios hechos a la información de entrada pueden provocar una re-calibración de los valores de los parámetros. Se han recomendado variados métodos para ayudar a seleccionar los valores adecuados de parámetros de Marxan (algunos de ellos son mencionados en este manual). Sin embargo, no hay nada mejor que la mera posibilidad de explorar tantos escenarios como lo permitan el tiempo de su proyecto y el presupuesto, y la seguridad de destinar el tiempo adecuado para esos experimentos.

6.2 Inspección Visual

La planificación de la conservación es una disciplina espacial y su medio natural es cartográfico. Aunque la muestra visual es quizás el más básico y con frecuencia el más subjetivo de todos los procedimientos de post-procesamiento, no se debe subestimar el poder del ojo humano para visualizar tendencias, y detectar detalles que pueden escaparse a las estadísticas espaciales sofisticadas. Como se mencionó en el primer capítulo de este manual, la información disponible no siempre es la

ideal, y hay muchos detalles de la selección de reservas que no se pueden incorporar a Marxan. El conocimiento de su región de planificación lo ayudará a impedir que ocurran errores obvios en la ubicación de las reservas. El conocimiento de los problemas debe utilizarse para actualizar los escenarios futuros ejecutados en Marxan. Modificar la solución al final de un análisis puede conllevar tanto a deficiencias como insuficiencias en la solución. Para contribuir con el proceso de inspección pudiera ser recomendable comparar las soluciones de manera visual con sus capas de información. Por ejemplo, debe haber notado que las soluciones resultan, en primer lugar, a partir de la distribución de los costos y no de los objetos de conservación, o que la sola distribución de unos pocos objetos de conservación, explica en gran medida la forma espacial de las soluciones. Aunque estos no son exactamente problemas, es importante conocerlos ya que pueden influir en la próxima iteración de selección de los valores para los parámetros.

6.3 Análisis de sensibilidad

Incluso cuando se está satisfecho con la calidad de los sistemas de reserva generados por Marxan, es importante analizar como serían modificadas las soluciones si se cambian algunos de los detalles del escenario. Si, por ejemplo, pequeños cambios efectuados en la información de costo conllevarían a grandes cambios en el sistema óptimo de reserva. Por esta razón, se deberá garantizar que el uso de una estructura de costo específica esté bien argumentado.

Se debe analizar a profundidad la capacidad que tienen sus soluciones de permanecer invariables ante pequeños cambios en los escenarios. Esto se hará mediante un análisis formal de sensibilidad donde los resultados de parámetros de modificación, las limitantes y la información sean comparados tanto cualitativa como cuantitativamente. Así, debido al gran volumen de variables y objetos de conservación que inciden en un análisis determinado, la mayoría de los análisis de sensibilidad no pueden examinar todos los aspectos, por tanto, sólo se examinan aquellos considerados atributos fundamentales. Los análisis de sensibilidad deben incluir detalles de escenarios que normalmente no están sujetos a experimentación, como el tamaño y forma de las unidades de planificación, las metas cuantitativas de conservación, la extensión de la región de planificación y diferentes tipos de información ecológica y de costo. Existen diversos enfoques para comparar la similitud de los sistemas de reserva siguiendo los cambios de escenarios, como la estadística Kappa (vea Richardson et al. 2006) y el análisis de agrupamiento (vea Airame 2005). Determinar si el resultado proveniente de los diferentes escenarios es similar, ayudará a evaluar la sensibilidad de sus soluciones. Informar la sensibilidad

de las soluciones ante los diferentes factores, puede resultar de gran utilidad para destacar el impacto que tienen las limitaciones sociales o políticas en las soluciones, o contribuir en las inversiones directas para la recopilación de información.

6.4 Hágase un experto

¡Haga todo lo explicado con mucha frecuencia! Y, por favor, comparta sus conclusiones con otros. Existe un servicio de lista de Marxan. Envíe un mensaje electrónico a 'marxan-owner@sib.uq.edu.au', solicitando alistarse en la membresía, o descargue Marxan desde su sitio web e indique en la planilla que puede descargar allí, que quisiera ser miembro de la lista. Si encuentra alguna falla en Marxan, por favor, infórmelo a Hugh Possingham mediante un mensaje electrónico a 'h.possingham@uq.edu.au', o al listado de Marxan mediante un mensaje electrónico a 'marxan@sib.uq.edu.au'. De igual manera, envíe cualquier notificación o publicación, informes y oportunidades de financiamiento relacionadas con Marxan.

(Esta página está en blanco intencionalmente)

Glosario

Agrupamiento: Cantidad mínima de un objeto de conservación requerida en unidades de planificación adyacentes antes de que se considere que ese ‘agrupamiento’ contribuye de manera efectiva con la meta de representación para ese objeto de conservación (Vea Sección 3.2.2.5 y Apéndice B-1.3.1) Asimismo, se puede asignar una cantidad de agrupamientos únicos para un objeto de conservación (Vea *distancia de separación*).

Algoritmo: Un proceso matemático que soluciona un problema sistemáticamente con el empleo de reglas o procesos bien definidos. Marxan puede utilizar varios algoritmos de optimización (algoritmo exacto, algoritmo heurístico, templado simulado y mejoramiento iterativo) para identificar soluciones para el diseño de reservas con un costo mínimo, sujeto a la limitante de alcanzar los objetivos planteados.

Algoritmo Heurístico: Una clase general de algoritmos sub-óptimos que emplean estrategias de ahorro de tiempo, o “reglas generales” para solucionar problemas. Si se utiliza en Marxan, las unidades de planificación se adicionan hasta que se cumplan las metas cuantitativas de biodiversidad (Vea Apéndice B-2.3).

Análisis de sensibilidad: El proceso de modificación de los parámetros de entrada, las limitaciones y la información, para evaluar de manera cuantitativa la influencia de diferentes variables en la solución final, es decir, cuan “sensibles” son los resultados ante las variaciones de esos parámetros.

Compacidad: Una medida del agrupamiento de unidades de planificación en una solución de reserva. Se calcula como una proporción de la longitud total de frontera con relación al área total de un sistema de reservas. Stewart y Possingham (2005) describen este concepto con más profundidad.

Costo de frontera: Conocido también como *longitud de frontera*, es el costo de frontera entre dos unidades de planificación adyacentes. Cuando una de las dos unidades está incluida en el sistema el costo de frontera es una medida relativa sobre la importancia de incluir la otra unidad de planificación y viceversa. A pesar de que la relación entre las dos unidades de planificación es, por lo general, la longitud de la frontera compartida, los costos de frontera se pueden especificar también entre unidades de planificación que no son adyacentes pero reflejan factores ecológicos o económicos.

Costo: El costo de incluir una unidad de planificación en un sistema de reservas. Este costo debe reflejar las consecuencias socio-políticas de reservar la unidad de planificación para acciones de conservación. Puede ser: área total, costo de adquisición o cualquier otra medida relativa de costo social, económico o ecológico. A cada *unidad de planificación* se le asigna un costo, aunque se pueden combinar diferentes medidas para crear un costo métrico. (Vea Sección 3.2.3.2 y Apéndice B-1.1).

Diseño del Sistema de reservas: Método utilizado para diseñar una red de áreas que, de manera colectiva, se centran en el objetivo del problema de conservación.

Distancia de Separación: Define la distancia mínima de separación a la que deben ubicarse los diferentes agrupamientos de un objeto de conservación para que sean considerados representaciones separadas. Puede verse como una forma de diseminación de riesgo. (Vea Sección 3.2.2.9).

Eficiencia: La capacidad de un sistema de reservas de cumplir todas las metas cuantitativas de conservación (ej. Ecosistemas, hábitats, especies) a un costo y con una compacidad aceptables.

Estadística Kappa: Un índice que compara la coincidencia/similitud espacial de un sistema de reservas contra aquel que se espera que ocurra solo por casualidad.

Factor de Penalidad de Especie (SPF): Un multiplicador definido para la penalidad aplicada a la función objetivo cuando una meta cuantitativa de objeto de conservación no se cumple en el escenario de reserva actual (Vea Apéndice B-1.3).

Factor de penalidad de los objetos de conservación: Vea *factor de penalidad de las especies*.

Frecuencia de selección: También conocido comúnmente como *irreemplazabilidad*. Es la frecuencia con la cual se selecciona una unidad de planificación determinada dentro de un sistema de reservas en una serie de soluciones de Marxan. Este valor se reporta en el archivo de salida "Soluciones Sumadas".

Función Objetivo: Una ecuación asociada con un problema de optimización que determina la eficacia de una solución ante un problema. En Marxan, el valor de la ecuación es una función de costos de unidad de planificación, costos de frontera y

penalizaciones. A cada solución del diseño de reserva se le asigna un valor de función objetivo; una solución con un valor bajo es más óptima que una con un valor alto. (Vea Sección 1.5).

Interfase de Usuario: Los medios mediante los cuales las personas interactúan con una aplicación de software específica. Una Interfase Gráfica de Usuario (GIU, por sus siglas en inglés) presenta la información, de manera asequible, utilizando gráficos, menús e íconos.

Irremplazabilidad: Vea Frecuencia de Selección.

Mejoramiento Iterativo: Método heurístico simple donde el algoritmo va a considerar un cambio aleatorio para determinar si el valor de la función objetivo mejora al realizar dicho cambio. En Marxan, el mejoramiento iterativo se puede utilizar para desechar de las soluciones aquellas unidades de planificación redundantes (Vea Apéndice B-2.2).

Meta Cuantitativa/ Meta Cuantitativa de Representación: Las metas cuantitativas son los valores cuantitativos (cantidad) de cada objeto de conservación que debe ser incluida en la solución final de reserva.

MGPH: Guía para las Mejores Prácticas con Marxan, un documento complementario a este Manual para el Usuario de Marxan.

Mínimo local/Óptimo local: Un mínimo local ocurre en el punto donde la mera adición de una *unidad de planificación* favorable o la eliminación de una *unidad de planificación* desfavorable del sistema de reservas, no bastará para mejorar el valor de la *función objetivo*. Esto significa, en esencia, que el sistema de reservas no puede ser mejorado sin cambiar su estructura de manera considerable.

Modificador de longitud de frontera (BLM): Una variable que indica el énfasis que debe hacerse para minimizar la longitud de frontera total del sistema de reservas con relación a su costo . Los mayores valores de BLM generarán sistemas de reserva más compactos. (Vea Sección 3.2.1.1.2 y Apéndice B-1.2).

Objeto de conservación: Un elemento de la biodiversidad seleccionado como el centro para la acción o planificación de la conservación. Puede incluir clasificaciones ecológicas, tipos de hábitat, especies, características físicas, procesos o cualquier elemento que pueda medirse en una *unidad de planificación*.

Planificación de la conservación de la Biodiversidad: Planeamiento amplio enfocado a alcanzar los requerimientos de representación y persistencia de diferentes rasgos de la biodiversidad. Esta planificación se dirige a sistemas completos, en lugar de sólo a partes de los sistemas y puede ocurrir a diferentes escalas. El esfuerzo debe concentrarse en las condiciones futuras deseadas, el manejo mejorado de la biodiversidad, y dirigirse a problemas y oportunidades. El resultado de la planificación de la conservación de la biodiversidad permitirá a los grupos interesados locales proteger sus comunidades y a los ecosistemas y biodiversidad de los que ellos dependen.

Planificación Sistemática de la Conservación: Método formal para identificar las zonas potenciales para el manejo de la conservación que van a cumplir con mayor efectividad un conjunto de objetivos específicos, por lo general, una representación mínima de la biodiversidad. El proceso incluye tanto la conservación marina como la terrestre. La efectividad de la planificación sistemática proviene de su capacidad para hacer un mejor uso de recursos fiscales limitados en aras de lograr los objetivos de conservación de una manera bien argumentada, contable y que reconozca con transparencia los requerimientos de los usuarios de los diferentes recursos.

Problema de cobertura máxima: El objetivo del problema de cobertura máxima es maximizar la protección de objetos de conservación, sujetos a la limitante de que los recursos empleados no sobrepasen un costo fijo. Marxan puede aproximarse a este problema de cobertura máxima con el uso de la función de Umbral de Costo, sin embargo, el resultado podrá ser sub-óptimo.

Problema de Conjunto Mínimo: El objetivo del problema de conjunto mínimo es minimizar los recursos consumidos, sujeto a la limitante de que todos los objetos cumplan sus objetivos de conservación. Marxan fue diseñado para solucionar este tipo de problema de conservación.

Sistema de Información Geográfica (SIG): Un sistema computarizado consistente de un hardware y un software necesarios para la obtención, el almacenaje, el manejo, el análisis y la presentación de información geográfica (espacial).

Software de apoyo en la toma de decisiones: Una aplicación computarizada que utiliza información sobre posibles acciones y limitaciones de esas acciones, para ayudar en el proceso de la toma de decisiones en aras de alcanzar un objetivo fijado.

Templado de Esquema Adaptable: Una función opcional de Marxan donde la planificación del *templado simulado* se hace automáticamente. Marxan muestra el problema y establece la temperatura inicial y las tasas de descenso de temperatura. (Vea también *Templado simulado* y *Templado de esquema Fijo*) (Vea Apéndice B-2.1.1)

Templado de esquema fijo: Una función opcional en Marxan donde la planificación del templado simulado la configura el propio usuario. Si se utiliza el esquema fijo, el esquema de templado (incluida la temperatura inicial y la tasa de descenso de la temperatura) se debe establecer por el usuario antes de comenzar a correr el algoritmo. (Vea también *Templado simulado* y *Templado de esquema adaptable*). (Vea Apéndice B-2.1.2).

Templado Simulado: Un método de optimización (algoritmo) con base en el mejoramiento iterativo, pero con aceptación aleatoria de movimientos errados a inicios del proceso para evitar quedar atascados en el valor de función objetivo de mínimo local. (Vea Apéndice B-2.1).

Unidades de Planificación: Las unidades de planificación son los componentes básicos de un sistema de reservas. Un área de estudio se divide en unidades de planificación que son parcelas geográficas más pequeñas de formas regulares o irregulares. Los ejemplos incluyen cuadrados, hexágonos, parcelas catastrales y unidades hidrológicas. (Vea Sección 1.7.1).

Verbosidad: La cantidad de información que se muestra en pantalla cuando Marxan se ejecuta. (Vea Sección 3.2.1.5.1).

(Esta página está en blanco intencionalmente)

Referencias Principales

Referencias fundamentales de Marxan

Ball, I.R. and H.P. Possingham. (2000). *Marxan (V1.8.2): Marine Reserve Design Using Spatially Explicit Annealing, a Manual*.

Possingham, H.P., I.R. Ball and S. Andelman. (2000). Mathematical methods for identifying representative reserve networks. In: S. Ferson and M. Burgman (Eds.), *Quantitative methods for conservation biology* (pp. 291-305). New York: Springer-Verlag.

Ardron, J. and C. Klein (Eds.). (2008). *Marxan good practices handbook*. St. Lucia, Queensland, Australia: University of Queensland, and Vancouver, British Columbia, Canada: Pacific Marine Analysis and Research Association.

Referencias seleccionadas que utilizan Marxan y describen conceptos

Airame, S. (2005). Channel Islands National Marine Sanctuary: Advancing the science and policy of marine protected areas. In: A. Scholz and D. Wright (Eds.), *Place Matters: Geospatial Tools for Marine Science, Conservation, and Management in the Pacific Northwest* (pp 91-124). Corvallis, OR: Oregon State University Press.

Ando, A., J. Camm, S. Polasky and A. Solow. (1998). Species distributions, land values, and efficient conservation. *Science*, 279: 2126-2128.

Araújo, M.B. (2004). Matching species with reserves - uncertainties from using data at different resolutions. *Biological Conservation*, 118: 533-538.

Armsworth, P.R., G.C. Daily, P. Kareiva and J.N. Sanchirico. (2006). Land market feedbacks can undermine biodiversity conservation. *Proceedings of the National Academy of Sciences (PNAS)*, 103: 5403-5408.

Ball, I.R. (2000). Mathematical applications for conservation ecology: The dynamics of tree hollows and the design of nature reserves. PhD Thesis, The University of Adelaide.

Balmford, A., K.J. Gaston, A.S.L. Rodrigues and A. James. (2000). Integrating costs of conservation into international priority setting. *Conservation Biology*, 14: 1-9.

- Ban, N. (in review). Beyond marine reserves: Exploring the approach of selecting permitted fishing areas.
- Beck, M.W. and M. Odaya. (2001). Ecoregional planning in marine environments: Identifying priority sites for conservation in the northern Gulf of Mexico. *Aquatic Conservation*, 11: 235-242.
- Burgman, M.A., H.P. Possingham, A.J.J. Lynch, D.A. Keith, M.A. McCarthy, S.D. Hopper, W.L. Drury, J.A. Passioura and R.J. Devries. (2001). A method for setting the size of plant conservation target areas. *Conservation Biology*, 15: 603-616.
- Cabeza, M. and A. Moilanen. (2001). Design of reserve networks and the persistence of biodiversity. *Trends in Ecology and Evolution*, 16: 242-248.
- Cabeza, M. (2003). Habitat loss and connectivity of reserve networks in probability approaches to reserve design. *Ecology Letters*, 6: 665-672.
- Carwardine, J., W. Rochester, K. Richardson, K. Williams, R. Pressey and H. Possingham. (2006). Conservation planning with irreplaceability: Does the method matter? *Biodiversity and Conservation*, 16: 1-14.
- Carwardine, J., K. Wilson, M. Watts, and H.P. Possingham. (2006). Where do we act to get the biggest conservation bang for our buck? A systematic spatial prioritisation approach for Australia. In: European Congress for Conservation Biology conference Proceedings. Eger, Hungary.
- Chan, K.M.A., M.R. Shaw, D.R. Cameron, E.C. Underwood and G.C. Daily. (2006). Conservation planning for ecosystem services. *PLoS Biology*, 4: e379.
- Church, R.L., D.M. Stoms, and F.W. Davis. (1996). Reserve selection as a maximal covering location problem. *Biological Conservation*, 76: 105-112.
- Cocks, K.D. and I.A. Baird. (1989). Using mathematical programming to address the multiple reserve selection problem: An example from the Eyre Peninsula, South Australia, *Biological Conservation*, 49: 113-130.
- Cook, R.R. and P.J. Auster. (2005). Use of simulated annealing for identifying essential fish habitat in a multispecies context. *Conservation Biology*, 19(3): 876-886.

Cowling, R.M. and R.L. Pressey. (2003). Introduction to systematic conservation planning in the Cape Floristic Region. *Biological Conservation*, 112:1-13.

Cowling, R.M., R.L. Pressey, M. Rouget and A.T. Lombard. (2003). A conservation plan for a global biodiversity hotspot - the Cape Floristic Region, South Africa. *Biological Conservation*, 112:191-216.

Cowling, R.M., R.L. Pressey, R. Sims-Castley, A. le Roux, E. Baard, C.J. Burgers and G. Palmer G. (2003). The expert or the algorithm? - Comparison of priority conservation areas in the Cape Floristic Region identified by park managers and reserve selection software. *Biological Conservation*, 112: 147-167.

Cowling, R.M., A.T. Knight, D.P. Faith, S. Ferrier, A.T. Lombard, A. Driver, M. Rouget, K. Maze and P.G. Desmet. (2004). Nature conservation requires more than a passion for species. *Conservation Biology*, 18: 1674-1676.

Csuti, B., S. Polasky, P.H. Williams, R.L. Pressey, J.D. Camm, M. Kershaw, A.R. Kiestler, B. Downs, R. Hamilton, M. Huso and K. Sahr. (1997). A comparison of reserve selection algorithms using data on terrestrial vertebrates in Oregon. *Biological Conservation*, 80: 83- 97.

Desmet, P. and R.M. Cowling. (2004). Using the species-area relationship to set baseline targets for conservation. *Ecology and Society*, 9(2): 11.

Erasmus, B.F.N., S. Freitag, K.J. Gaston, B.H. Erasmus and A.S. van Jaarsveld. (1999). Scale and conservation planning in the real world. *Proceedings of the Royal Society of London Series B-Biological Sciences*, 266: 315-319.

Ferdaña, Z. (2005). Nearshore marine conservation planning in the Pacific Northwest: Exploring the use of a siting algorithm for representing marine biodiversity, in D.J. Wright and A.J. Scholz (Eds.), *Place Matters: Geospatial Tools, for Marine Science, Conservation, and Management in the Pacific Northwest*. Corvallis, OR: Oregon State University Press.

Fernandes, L., J. Day, A. Lewis, S. Slegers, B. Kerrigan, D. Breen, D. Cameron, B. Jago, J. Hall, D. Lowe, J. Innes, J. Tanzer, V. Chadwick, L. Thompson, K. Gorman, M. Simmons, B. Barnett, K. Sampson, G. De'ath, B. Mapstone, H. Marsh, H. Possingham, I. Ball, T. Ward, K. Dobbs, J. Aumend, D. Slater and K. Stapleton. (2005). Establishing representative no-take areas in the Great Barrier Reef: Large-

scale implementation of theory on marine protected areas. *Conservation Biology*, 19: 1733-1744.

Ferrier, S. (2002). Mapping spatial pattern in biodiversity for regional conservation planning: Where to from here? *Systematic Biology*, 51(2): 331-363.

Ferrier, S., G. Watson, J. Pearce, M. Drielsma. (2002). Extended statistical approaches to modelling spatial pattern in biodiversity in northeast New South Wales. I. Species-level modelling. *Biodiversity and Conservation*, 11: 2275-2307.

Ferrier, S., M. Drielsma, G. Manion, G. Watson. (2002). Extended statistical approaches to modelling spatial pattern in biodiversity in northeast New South Wales. II. Community-level modeling. *Biodiversity and Conservation*, 11: 2309-2338.

Leslie, H., M. Ruckelshaus, I.R. Ball, S. Andelman and H.P. Possingham. (2003). Using siting algorithms in the design of marine reserve networks. *Ecological Applications*, 13: S185-S198.

Higgins, J.V., M.T. Bryer, M.L. Khoury, and T.W. Fitzhugh. (2005). A freshwater classification approach for biodiversity conservation planning. *Conservation Biology* 19(2): 432-445.

Kirkpatrick, J.B. (1983). An iterative method for establishing priorities for selection of nature reserves: An example from Tasmania. *Biological Conservation*, 25: 127-134.

Kirkpatrick J.B. and M.J. Brown. (1994). A comparison of direct and environmental domain approaches to planning reservation of forest higher plant communities in Tasmania. *Conservation Biology*, 8: 217-224.

Knight, A.T., R.M. Cowling and B.M. Campbell. (2006). An operational model for implementing conservation action. *Conservation Biology*, 20: 408-419.

Leslie, H., M. Ruckelshaus, I.R. Ball, S. Andelman and H.P. Possingham. (2003). Using siting algorithms in the design of marine reserve networks. *Ecological Applications*, 13(1): S185-S198.

Lombard A.T., R.M. Cowling. R.L. Pressey and A.G. Rebelo. (2003). Effectiveness of land classes as surrogates for species in conservation planning for the Cape Floristic Region. *Biological Conservation*, 112: 45-62.

McDonald, R., M. McKnight, D. Weiss, E. Selig, M. O'Connor, C. Violin. and A Moody. (2005). Species compositional similarity and ecoregions: Do ecoregion boundaries represent zones of high species turnover? *Biological Conservation*, 126: 24-40.

McDonnell, M.D., H.P. Possingham, I.R. Ball and E.A. Cousins. (2002). Mathematical methods for spatially cohesive reserve design. *Environmental Modeling and Assessment*, 7: 107-114.

Margules, C.R. and R.L Pressey. (2000). Systematic conservation planning. *Nature*, 405: 243-253.

Meir, E., S. Andelman and H.P. Possingham. (2004). Does conservation planning matter in a dynamic and uncertain world? *Ecology Letters*, 7: 615-622.

Naidoo, R. and W.L. Adamowicz. (2006). Modeling opportunity costs of conservation in transitional landscapes. *Conservation Biology*, 20: 490-500.

Naidoo, R., A. Balmford, P.J. Ferraro, S. Polasky, T.H. Ricketts and M. Rouget. (2006). Integrating economic costs into conservation planning. *Trends in Ecology and Evolution*, 21: 681-687.

Nicholls, A.O. and C.R. Margules. (1993). An upgraded reserve selection algorithm. *Biological Conservation*, 64: 165-169.

Nicholson, E., and H.P. Possingham. (2006). Objectives for multiple species conservation planning. *Conservation Biology*, 20: 871-881.

Noss, R.F. (2004). Conservation targets and information needs for regional conservation planning. *Natural Areas Journal*, 24: 223-231.

Pierce S.M., R.M. Cowling, A.T. Knight, A.T. Lombard, M.Rouget and T. Wolf. (2005). Systematic conservation planning products for land-use planning: Interpretation for implementation. *Biological Conservation*, 125: 441-458.

Possingham, H.P., I.R. Ball and S. Andelman. (2000). Mathematical methods for identifying representative reserve networks. In: S. Ferson and M. Burgman (Eds.), *Quantitative methods for conservation biology* (pp. 291-305). New York: Springer-Verlag.

Possingham, H.P., J.R. Day, M. Goldfinch and F. Salzborn. (1993). The mathematics of designing a network of protected areas for conservation. In: D.J. Sutton, C.E.M. Pearce and E.A. Cousins (Eds.), *Decision Sciences: Tools for Today*. Proceedings of 12th National ASOR Conference. (pp. 536-545). Adelaide: ASOR.

Possingham, H.P., J. Franklin, K.A. Wilson and T.J. Regan. (2005). The roles of spatial heterogeneity and ecological processes in conservation planning. In: G.M. Lovett, C.G. Jones, M.G. Turner and K.C. Weathers (Eds). *Ecosystem function in heterogeneous landscapes* (pp 389-406). New York: Springer-Verlag.

Pressey, R.L. (2002). The first reserve selection algorithm. *Progress in Physical Geography*, 26(3): 434-441.

Pressey, R.L. (2004). Conservation planning and biodiversity: Assembling the best data for the job. *Conservation Biology*, 18(6): 1677-1681.

Pressey, R.L., M. Cabeza, M.E. Watts, R.M. Cowling and K.A. Wilson. (in press) Conservation planning in a changing world. *Trends in Ecology and Evolution*.

Pressey, R.L., R.M. Cowling and M. Rouget. (2003). Formulating conservation targets for biodiversity pattern and process in the Cape Floristic Region, South Africa. *Biological Conservation*, 112: 99-127.

Pressey, R.L., I.R. Johnson and P.D. Wilson. (1994). Shades of irreplaceability: Towards a measure of the contribution of sites to a reservation goal. *Biodiversity and Conservation*, 3: 242-262.

Pressey, R.L. and V.S. Logan. (1994). Level of geographical subdivision and its effects on assessments of reserve coverage: A review of regional studies. *Conservation Biology*, 8: 1037-1046.

Pressey, R.L. and V.S. Logan. (1998). Size of selection units for future reserves and its influence on actual vs targeted representation of features: A case study in western New South Wales. *Biological Conservation*, 85: 305-319.

Pressey, R.L., P.H. Possingham, and J.R. Day. (1997). Effectiveness of alternative heuristic algorithms for identifying indicative minimum requirements for conservation reserves. *Biological Conservation*, 80: 207-219.

Pressey, R.L. and K.H. Taffs. (2001). Scheduling conservation action in production landscapes: Priority areas in western New South Wales defined by irreplaceability and vulnerability to vegetation loss. *Biological Conservation*, 100: 355-376.

Pressey, R.L., M.E. Watts, and T.W. Barrett. (2004). Is maximizing protection the same as minimizing loss? Efficiency and retention as alternative measures of the effectiveness of proposed reserves. *Ecology Letters*, 7: 1035-1046.

Pressey, R.L., G.L. Whish, T.W. Barrett and M.E. Watts. (2002). Effectiveness of protected areas in north-eastern New South Wales: Recent trends in six measures. *Biological Conservation*, 106: 57-69.

Rebelo, A.G. and W.R. Siegfried. (1992). Where should nature reserves be located in the Cape Floristic Region, South Africa? Models for the spatial configuration of a reserve network aimed at maximizing the protection of floral diversity. *Conservation Biology*, 6(2): 243-252.

Richardson, E.A., M.J. Kaiser, G. Edwards-Jones, and H.P. Possingham. (2006). Sensitivity of marine-reserve design to the spatial resolution of socioeconomic data. *Conservation Biology*, 20 (4): 1191–1202.

Rodrigues, A.S.L. and T.M. Brooks. (2007). Shortcuts for biodiversity conservation planning: The effectiveness of surrogates. *Annual Review of Ecology, Evolution, and Systematics*, 38: 713-737.

Rondinini, C., K. Wilson, L. Boitani, H. Grantham, H. Possingham. (2006). Tradeoffs of different types of species occurrence data for use in systematic conservation planning. *Ecology Letters*, 9: 1136-1145.

Sarkar, S., J. Justus, R. Fuller, C. Kelley, J. Garson and M. Mayfield. (2005). Effectiveness of environmental surrogates for the selection of conservation area networks. *Conservation Biology*, 19: 815-825.

Sarkar S., R.L. Pressey, D.P. Faith, C.R. Margules, T. Fuller, D.M. Stoms, A. Moffett, K. Wilson, K.J. Williams, P.H. Williams and S. Andelman. (2006). Biodiversity conservation planning tools: Present status and challenges for the future. *Annual Review of Environment and Resources*, 31: 123-159.

Smith, R., P. Goodman and W. Matthews. (2006). Systematic conservation planning: A review of perceived limitations and an illustration of the benefits using a case study from Maputaland, South Africa. *Oryx*, 40: 400-410.

Stewart, R.R. (2003). Opportunity cost of ad hoc marine reserve design decisions: An example from South Australia. *Marine Ecology Progress Series*, 253: 25-38.

Stewart, R.R., T. Noyce and H.P. Possingham. (2003). The opportunity cost of ad-hoc marine reserve design decisions - An example from South Australia. *Marine Ecology Progress Series*, 253: 25-38.

Stewart, R.R. and H.P. Possingham. (2005). Efficiency, costs and trade-offs in marine reserve system design. *Environmental Modeling and Assessment*, 10: 203-213.

Stewart, R.R. and H.P. Possingham. (2003). A framework for systematic marine reserve design in South Australia: A case study. In *Proceedings of the Inaugural World Congress on Aquatic Protected Areas*, Cairns - August 2002.

Underhill, L.G. (1994). Optimal and suboptimal reserve selection algorithms. *Biological Conservation*, 70: 85-87.

Warman, L.D., A.R.E. Sinclair, G.G.E. Scudder, B. Klinkenberg, and R.L. Pressey. (2004). Sensitivity of systematic reserve selection to decisions about scale, biological data, and targets: Case study from southern British Columbia. *Conservation Biology*, 18: 655-666.

Wilson, K.A., R.L. Pressey, A.N. Newton, M.A. Burgman, H.P. Possingham and C.J. Weston. (2005). Measuring and incorporating vulnerability into conservation planning. *Environmental Management*, 35: 527-543

Wilson, K.A., A.N. Newton, C. Echeverría, C.J. Weston and M.A. Burgman. (2005). A vulnerability analysis of the temperate forests of south central Chile. *Biological Conservation*, 122: 9-21.

Wilson, K.A., M.I. Westphal, H.P. Possingham and J. Elith. (2005). Sensitivity of conservation planning to different approaches to using predicted species distribution data. *Biological Conservation*, 122: 99-112.

Wilson, K.A., M. McBride, M. Bode, and H.P. Possingham. (2006). Prioritising global conservation efforts. *Nature*, 440: 337-340.

Winter, S.J., K.J. Esler and M. Kidd. (2005). An index to measure the conservation attitudes of landowners towards Overberg Coastal Renosterveld, a critically endangered vegetation type in the Cape Floral Kingdom, South Africa. *Biological Conservation*, 126: 383-394.

Zacharias, M.A. and J.C. Roff. (2000). A hierarchical ecological approach to conserving marine biodiversity. *Conservation Biology*, 14(5): 1327-1334.

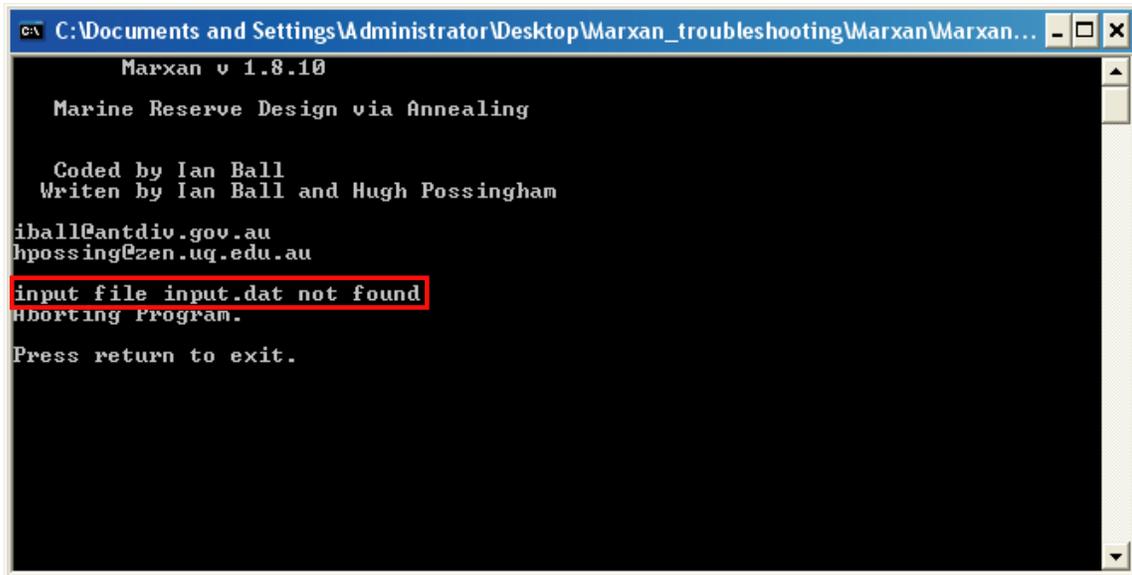
Zacharias, M.A. and J.C. Roff. (2001). Use of focal species in marine conservation and management: A review and critique. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 11: 59-76.

(Esta página está en blanco intencionalmente)

Apéndice A – Solución de problemas

En esta sección, ofrecemos ejemplos de mensajes de errores comunes que se encuentran durante la ejecución de Marxan, y los errores que pueden provocar que se generen esos mensajes

A-1. Marxan se detiene porque no se ha encontrado algún archivo o parámetro de entrada

A screenshot of a Windows command prompt window. The title bar reads "C:\Documents and Settings\Administrator\Desktop\Marxan_troubleshooting\Marxan\Marxan...". The terminal text is as follows:

```
Marxan v 1.8.10
Marine Reserve Design via Annealing

Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au
input file input.dat not found
Aborting Program.

Press return to exit.
```

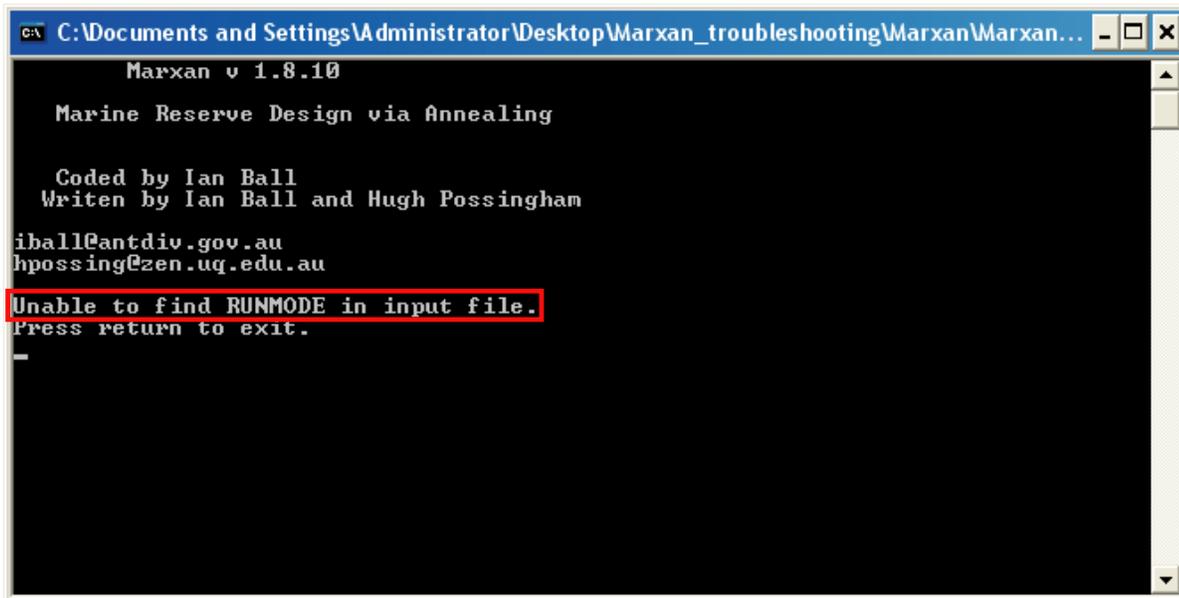
The line "input file input.dat not found" is highlighted with a red rectangular box.

Marxan se detendrá con este mensaje (arriba) si no es capaz de localizar el **Archivo de Entrada de Parámetros** (input.dat).

¿Qué hacer?

1. Compruebe que el **Archivo de Entrada de Parámetros** y el ejecutable (Marxan.exe) de Marxan se encuentren en el mismo directorio.
2. Compruebe que el nombre y la extensión del **Archivo de Entrada de Parámetros** exactamente leen – “input.dat”.

Marxan se detendrá con un mensaje similar de error (debajo) si no puede encontrar el importante parámetro 'RUNMODE'.



The image shows a terminal window titled "C:\Documents and Settings\Administrator\Desktop\Marxan_troubleshooting\Marxan\Marxan...". The terminal content is as follows:

```
Marxan v 1.8.10
Marine Reserve Design via Annealing

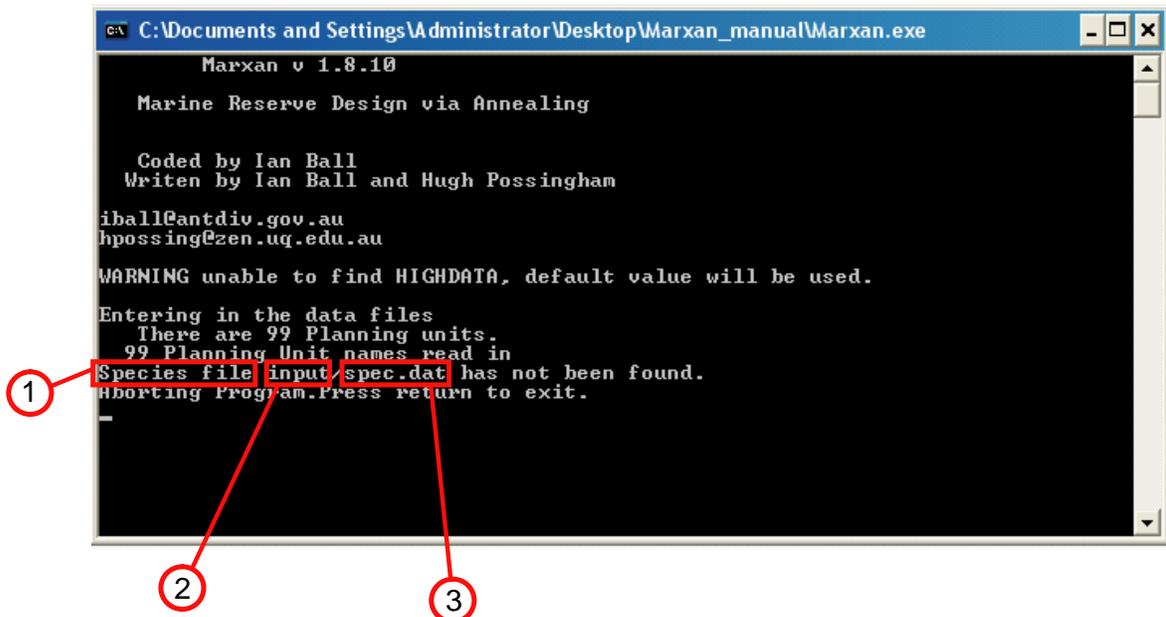
Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au
Unable to find RUNMODE in input file.
Press return to exit.
```

¿Qué hacer?

1. Compruebe que hay un valor (que sea válido) para ese parámetro en el **Archivo de Entrada de Parámetros**. No hay un valor predeterminado para 'RUNMODE'.
2. Compruebe que el parámetro tiene el nombre correcto, debe estar en letra mayúscula sin espacios o caracteres extras. (si se utiliza **Inedit** esto se hace automáticamente).

Marxan se detendrá con este mensaje (abajo) si encuentra que el **Archivo de Entrada de Parámetros** no puede localizar uno de los otros tres archivos de entrada requeridos



```
C:\Documents and Settings\Administrator\Desktop\Marxan_manual\Marxan.exe
Marxan v 1.8.10

Marine Reserve Design via Annealing

Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au

WARNING unable to find HIGHDATA, default value will be used.

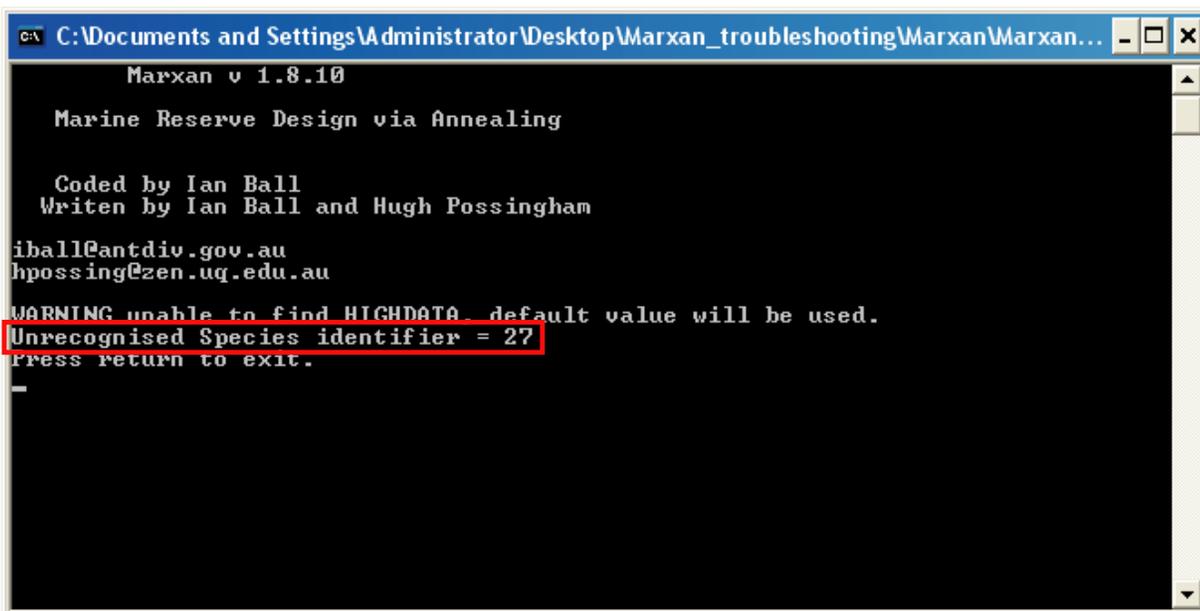
Entering in the data files
There are 99 Planning units.
99 Planning Unit names read in
Species file input/spec.dat has not been found.
Aborting Program.Press return to exit.
```

1. Tipo de archivo de entrada que no se alcanza (en este ejemplo, el **Archivo de Entrada de Parámetros**, también conocido como Archivo de Especies)
2. Nombre del directorio que Marxan está buscando en los archivos de entrada. Es el mismo que se especifica en el **Archivo de Entrada de Parámetros**
3. Nombre del archivo que Marxan está buscando. Es el nombre especificado en el **Archivo de Entrada de Parámetros**

¿Qué hacer?

1. Compruebe que el nombre en el directorio que contiene los archivos de entrada sea el mismo que el nombre dado en el Archivo de Entrada de Parámetros bajo la variable Directorio de Entrada ('INPUTDIR').
2. Compruebe que el directorio que contiene los archivos de entrada se encuentra dentro del mismo directorio que el ejecutable de Marxan (Marxan.exe) o que el lugar donde está ubicado se haya especificado correctamente en el **Archivo de Entrada de Parámetros**
3. Compruebe que el nombre del archivo dado en el **Archivo de Entrada de Parámetros** concuerda exactamente con el nombre en el directorio de archivo de entrada.

A-2. Marxan se detiene a consecuencia de un identificador no reconocido



```
Marxan v 1.8.10
Marine Reserve Design via Annealing

Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au

WARNING unable to find HIGHDATA - default value will be used.
Unrecognised Species identifier = 27
Press return to exit.
```

En este ejemplo (arriba) el error se debe a una inconsistencia en el Objeto de Conservación (conocido como Especies). Ese error puede también recibir el término “Unidad de Planificación” en lugar de “Especies” .

Este error ocurrirá si hay alguna inconsistencia en las Identificaciones del Objeto de Conservación o las Identificaciones de Unidad de Planificación entre archivos. En otras palabras, las identificaciones listadas en el **Archivo de Objeto de Conservación** o en el **Archivo de Unidad de Planificación** no son las mismas representadas en el **Archivo de Unidad de Planificación versus Archivo de Objeto de Conservación** (y viceversa)

¿Qué hacer?

1. Compruebe que las Identificaciones de Objeto de Conservación listadas en el **Archivo de Unidades de Planificación versus Objetos de Conservación** concuerdan exactamente con las identificaciones listadas en el **Archivo de Objetos de Conservación**.
2. Compruebe que las identificaciones de las unidades de planificación listadas en el **Archivo de Unidades de Planificación versus Objetos de Conservación** concuerda exactamente con las identificaciones en el **Archivo de Unidades de Planificación**.

A-3. Marxan comienza la primera corrida pero se detiene porque no puede salvar los resultados requeridos

```
Marxan v 1.8.10
Marine Reserve Design via Annealing

Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au

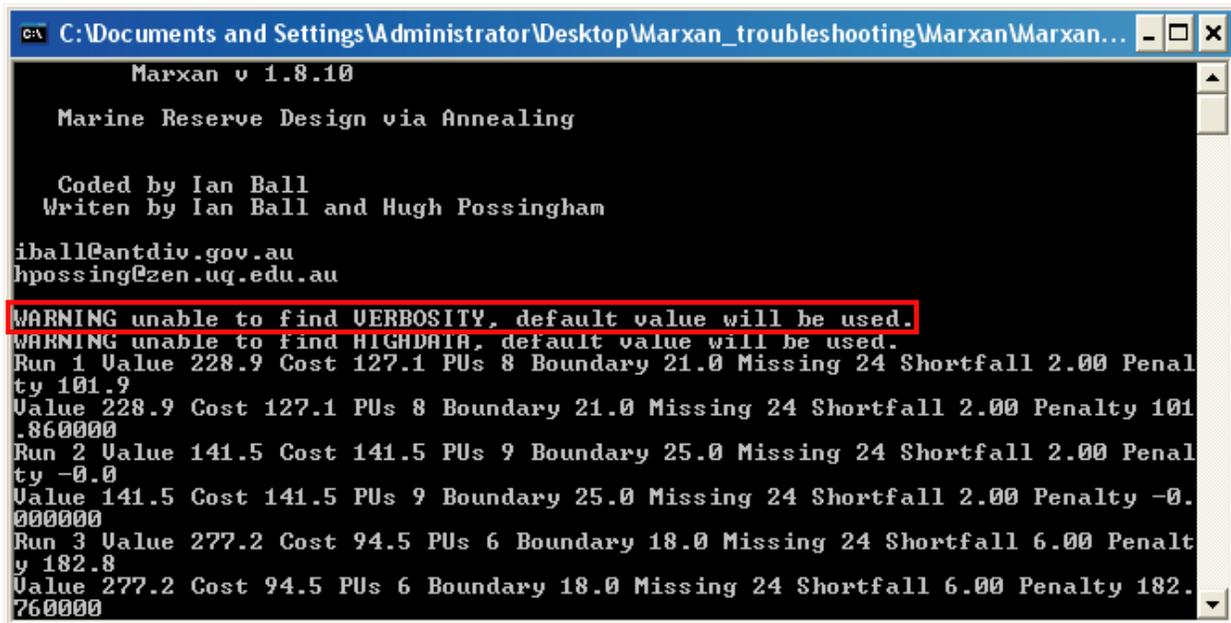
WARNING unable to find HIGHDATA, default value will be used.
Run 1 Value 173.7 Cost 144.1 PUs 9 Boundary 29.0 Missing 24 Shortfall 2.00 Penalty 29.6
Value 173.7 Cost 144.1 PUs 9 Boundary 29.0 Missing 24 Shortfall 2.00 Penalty 29.620000
Cannot save output to out put/trouble_shoot_r00001.dat
Press return to exit.
```

1. Mensaje identificando que Marxan no puede salvar los resultados deseados.
2. Nombre del directorio donde Marxan está tratando de salvar los archivos de resultados (en el ejemplo, el error se debe a un espacio entre las palabras 'out' y 'put').
3. Nombre del archivo que Marxan trata de salvar cuando identifica el error.

¿Qué hacer?

1. Compruebe que el nombre del directorio que se ha establecido para salvar los archivos de resultados sea el mismo indicado en el Archivo de Entrada de Parámetros, bajo la variable Output Directory ('OUTPUTDIR') (directorio de salida).
2. Compruebe que ese directorio se encuentra ubicado en el mismo directorio del ejecutable de Marxan (Marxan.exe) o la ubicación se ha especificado correctamente en el **Archivo de Entrada de Parámetro** (Esto se hace fácilmente utilizando **Inedit**).
3. Compruebe que el nombre de archivo indicado en el **Archivo de Entrada de Parámetros** concuerda exactamente con el nombre que aparece en el directorio de archivo de entrada.

A-4. Marxan se ejecuta pero advierte que no puede localizar una variable específica



```
Marxan v 1.8.10

Marine Reserve Design via Annealing

Coded by Ian Ball
Written by Ian Ball and Hugh Possingham

iball@antdiv.gov.au
hpossing@zen.uq.edu.au

WARNING unable to find UERBOSITY, default value will be used.
WARNING unable to find HIGHDATA, default value will be used.
Run 1 Value 228.9 Cost 127.1 PUs 8 Boundary 21.0 Missing 24 Shortfall 2.00 Penalty 101.9
Value 228.9 Cost 127.1 PUs 8 Boundary 21.0 Missing 24 Shortfall 2.00 Penalty 101.9
Run 2 Value 141.5 Cost 141.5 PUs 9 Boundary 25.0 Missing 24 Shortfall 2.00 Penalty -0.0
Value 141.5 Cost 141.5 PUs 9 Boundary 25.0 Missing 24 Shortfall 2.00 Penalty -0.0
Run 3 Value 277.2 Cost 94.5 PUs 6 Boundary 18.0 Missing 24 Shortfall 6.00 Penalty 182.8
Value 277.2 Cost 94.5 PUs 6 Boundary 18.0 Missing 24 Shortfall 6.00 Penalty 182.8
```

Marxan mostrará esa advertencia cuando no pueda leer el valor de algún parámetro en el **Archivo de Entrada de Parámetros**. Esto se debe a que el valor no fue especificado o hubo un error de formato. Marxan comenzará a correr de manera normal pero utilizará el valor predeterminado para el parámetro.

¿Qué hacer?

1. Si no se estableció un valor intencionalmente y usted está satisfecho con el valor predeterminado, no haga nada. Marxan correrá de manera normal.
2. Si no desea utilizar el valor predeterminado, verifique lo siguiente:
 - o Que el **Archivo de Entrada de Parámetros** (input.dat) contiene un valor para el parámetro que falta
 - o Que el parámetro está correctamente identificado en el **Archivo de Entrada de Parámetros**. Debe ser en letra mayúscula sin espacio entre ellas o caracteres adicionales.
 - o Compruebe que hay un retorno de párrafo al final del **Archivo de Entrada de Parámetros**

Nota: NO se preocupe por el mensaje “WARNING unable to find HIGHDATA (advertencia, no se puede encontrar HIGHDATA)”, este refiere a un parámetro en Marxan que ya está caduco. Marxan correrá correctamente incluso cuando se muestre esa advertencia. Ese error de mensaje no aparecerá en las versiones futuras de Marxan.

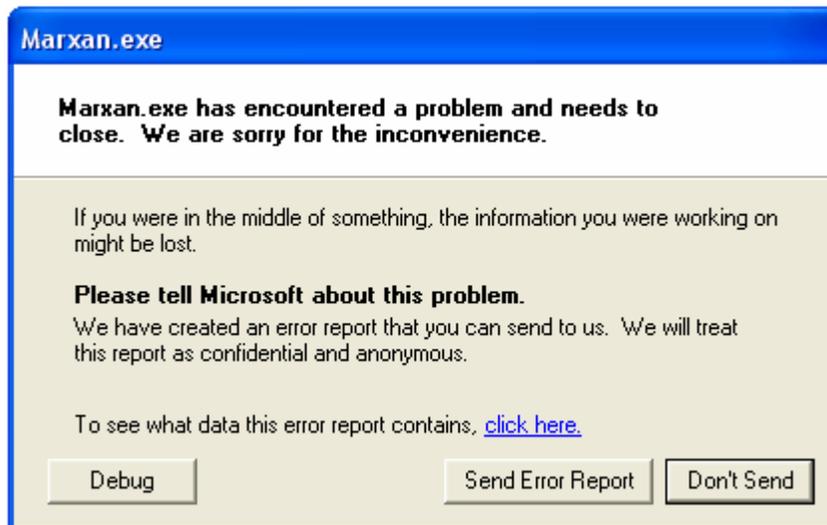
A-5. No se están guardando los resultados en el directorio de salida

¿Qué hacer?

1. Compruebe que se ha especificado un directorio de salida en el **Archivo de Entrada de Parámetros**. Si no se especifica ningún directorio no se salvarán los resultados.
2. Compruebe que ha seleccionado los resultados que se deben salvar. Eso se puede hacer analizando directamente el **Archivo de Entrada de Parámetros**; el valor '1' debe aparecer al lado de todos los archivos que se van salvar. El valor '0' debe aparecer al lado de aquellos resultados que no se deseen salvar.

A-6. Marxan muestra una falla cuando se ejecuta y la pantalla de Marxan se cierra

Esto puede suceder por diferentes razones, pero por lo general, ocurre cuando falta un valor de los archivos de información de entrada. Si se está corriendo Marxan en una computadora que utilice Windows, el mensaje puede estar acompañado del siguiente mensaje de error:



Si aparece este mensaje, por favor hágaselo saber a Microsoft. Sin embargo, dudamos que ellos hagan algo al respecto. Marxan no obtiene muchos fondos de ellos

¿Qué hacer?

1. Compruebe que hay valores para todas las variables en cada uno de los archivos de información de entrada. Esto incluye los tres archivos requeridos (**Archivo de Objeto de Conservación**, **Archivo de Unidades de Planificación**, y el **Archivo de Objetos de Conservación versus Unidades de Planificación**) y los dos archivos opcionales (el **Archivo de Longitud de Frontera** y el **Archivo de Definición de Bloque**) Para cada uno de esos archivos debe haber un conjunto completo de valores para todas las variables requeridas y cualquier variable no requerida que esté siendo utilizada.
2. Compruebe que hay un único retorno de párrafo al final de cada uno de los archivos de entrada.

(Esta página está en blanco intencionalmente)

Apéndice B – Información Técnica de Marxan

Este apéndice contiene detalles técnicos sobre la forma en que Marxan se ejecuta. Aunque esa información no es necesaria para realizar las corridas básicas, conocer como se ejecuta el programa lo ayudará a comprender como los cambios que se realicen en los diferentes parámetros tendrán un impacto en los resultados.

B-1. La Función Objetivo

El “corazón” matemático de Marxan es la *función objetivo* que evalúa y compara los sistemas de reservas potenciales. En la Sección 1.5, se dio una breve introducción referente a esta función. En esta sección se ofrece información sobre el modo de calcular cada uno de los componentes en la función objetivo. La función objetivo en Marxan está diseñada de manera que mientras menor sea el valor, mejor el resultado. Potencialmente puede incluir los siguientes componentes:

$$\sum_{PUs} Cost + BLM \sum_{PUs} Boundary + \sum_{ConValue} SPF \times Penalty + CostThresholdPenalty(t)$$

B-1.1 Costo

El componente ‘Cost’ de la función objetivo es la suma del costo dado a cada unidad de planificación incluida en el sistema de reservas. La información referente al costo se encuentra en el **Archivo de Unidades de Planificación** (vea Sección 3.2.3). Ese costo puede ser el costo económico real de la compra de la unidad de planificación, o puede reflejar un concepto más abstracto como el costo de la posibilidad de poner la unidad de planificación bajo protección. Vea la Sección 3.2.3.2 para información sobre el costo de unidad de planificación.

B-1.2 Frontera y Modificador de Longitud de Frontera (BLM)

‘Boundary’ (frontera) es la longitud de la frontera que rodea a un sistema de reservas. Al incluir un término de longitud de frontera en la función objetivo, podemos controlar el nivel de fragmentación en el sistema de reservas. El componente ‘Boundary’ (frontera) de la función objetivo se calcula primeramente sumando las longitudes de todas las fronteras de las unidades de planificación existentes en la reserva y de aquellas que se ubican fuera de la reserva. Las fronteras entre dos unidades de planificación que se encuentren dentro del mismo sistema de reservas no se cuentan. La información referente a las longitudes de las fronteras existentes entre unidades de planificaciones se archiva en el **Archivo de Longitud de Frontera** (vea

Sección 3.3.1). El valor dado a una frontera compartida puede reflejar tanto la longitud geográfica real como alguna otra asociación entre las unidades de planificación, por ejemplo, las fronteras que son deseables o no deseadas. En la Sección 3.3.1.2 se describen los diferentes usos posibles del valor o costo de frontera.

Como es muy probable que el valor de longitud de frontera se exprese en unidades que son diferentes a la medida del costo de unidad de planificación, ambas medidas no se pueden simplemente sumarse juntas. Se utiliza un factor de multiplicación para permitir que la longitud de frontera se adicione a la medida de costo. Este es el 'Modificador de Longitud de Frontera' (BLM). Al calcular el valor de función objetivo, la suma de las fronteras se multiplica por el BLM. La inclusión de este modificador no sólo permite la compatibilidad entre las diferentes métricas, sino que ofrece además un método para controlar la importancia de la compacidad de la reserva, con relación al costo de reserva. Un planificador de conservación puede explorar este tema modificando el BLM. Si el valor dado al BLM es cero, entonces la longitud de frontera no será incluida en la función objetivo. Si se le da un valor alto al BLM, entonces obtener un sistema compacto superará a todas las demás consideraciones.

B-1.3 Penalidad y Factor de Penalidad de Especies (SPF)

El componente de Penalidad de la función objetivo de Marxan es la penalidad aplicada a un sistema de reservas por no representar adecuadamente los objetos de conservación. Se basa en el principio de que si un objeto de conservación está por debajo de su meta de representación, entonces la penalidad debe ser una aproximación del costo por incrementar el objeto de conservación hasta su meta de representación. Por ejemplo, si el requerimiento era representar cada objeto de conservación, con al menos un caso, la penalidad por no representar un objeto de conservación determinado, sería el costo de la unidad de planificación menos costosa que incluya un caso de ese objeto de conservación. Si no se logran representar varios objetos de conservación, puede entonces generar un sistema de reservas totalmente representativo adicionando las unidades de planificación menos costosas que contienen los objetos de conservación no representados.

Marxan emplea un algoritmo 'ávido' para identificar la manera menos costosa en que cada objeto de conservación puede ser representado y esto forma la penalidad base para ese objeto de conservación. Para esto, Marxan adiciona las unidades de planificación menos costosas que lograrán la meta de representación. Este método se describe en el siguiente pseudo-código.

- I. Para cada unidad calcule el valor de 'costo por hectárea'.
 - A. Determine que por ciento de la meta cuantitativa para ese objeto de conservación fue aportado por esa unidad de planificación.
 - B. Determine el costo económico de la unidad de planificación.
 - C. Determine la longitud de frontera de la unidad de planificación.
 - D. El costo general es costo económico + longitud de frontera x BLM (Multiplicador de Longitud de Frontera)
 - E. El costo-por-hectárea es entonces el valor del objeto de conservación dividido por el costo general.

- II. Seleccione la unidad de planificación con el menor costo por hectárea. Adicione su costo al costo total de la ejecución y el nivel de representación del objeto de conservación con respecto de la representación total.

- III. Continúe adicionando esos totales hasta que encuentre una colección de unidades de planificación que representen adecuadamente al objeto de conservación.

- IV. La penalidad base para el objeto de conservación es el costo total (incluido la longitud de frontera multiplicado por el modificador de longitud) de esas unidades de planificación.

De esta forma, si una unidad de conservación no estuvo representada en su totalidad, entonces la penalidad sería la misma que el costo de sumar al sistema el conjunto de unidades de planificación, escogido utilizando código anterior, asumiendo que se encuentran separadas entre ellas, por razones de longitud de frontera. Este valor se calcula rápidamente pero tiende a ser mayor que el valor óptimo. Pueden existir formas más efectivas para representar un objeto de conservación que la determinada por un algoritmo 'ávido'.¹⁰ Analice el siguiente ejemplo.

El Objeto de Conservación A está presente en varias unidades de planificación, las mejores son:

Unidad de Planificación	Costo	Cantidad de Objeto A
1	\$2	3
2	\$4	5
3	\$5	5
4	\$8	6

¹⁰ Vea la Sección B-2.3.1 de este apéndice para una descripción de la heurística 'ávida'.

La meta cuantitativa para el objeto A es 10 unidades. Si utilizamos el algoritmo 'ávido' lo representaríamos con las unidades de planificación 1,2, y 3 (seleccionadas en ese orden) por un costo total de \$11. Obviamente, si se escogen las unidades de planificación 2 y 3, aún se podría representar adecuadamente el objeto A, pero el costo sería sólo \$9. En este ejemplo se muestra sencillamente un caso donde el algoritmo 'ávido' no genera los mejores resultados. El algoritmo 'ávido' es rápido y genera resultados razonables.

El programa tendrá la tendencia a sobreestimar y nunca subestimar las penalidades cuando utiliza un algoritmo 'ávido'. Sin embargo, no se desea tener un valor de penalidad muy bajo porque entonces la función objetivo pudiera no mejorar representando en su totalidad a todos los objetos de conservación. No representa un problema tener penalidades mayores de lo que realmente deben ser, en ocasiones hasta es conveniente. El costo de frontera de una unidad de planificación en el pseudo-código anterior es la suma de todas sus fronteras. A diferencia del componente de frontera de la función objetivo, este asume que la unidad de planificación no tiene fronteras comunes con el resto de la reserva y, una vez más, sobreestimaré el costo de la unidad de planificación y la penalidad.

Sería ideal hacer un nuevo cálculo de las penalidades después de cada cambio en el sistema de reservas. Sin embargo, eso requeriría de mucho tiempo y resulta ser más efectivo trabajar con las penalidades que sólo cambian en su forma más simple, desde un punto del algoritmo al próximo punto. La penalidad se calcula y se fija en la fase inicial del algoritmo. Se aplica de una manera lineal sencilla, si el objeto de conservación ha logrado la mitad de su meta cuantitativa, entonces recibe la mitad de su penalidad. El problema con esto es que se puede enfrentar una situación donde sólo se necesite una pequeña cantidad para cumplir una meta de objeto de conservación, sin embargo no hay forma de hacerlo de manera que se reduzca el valor objetivo. Si tomamos el ejemplo utilizado anteriormente, entonces la penalidad para el objeto de conservación A es 11. Si las unidades de planificación 1 y 4 ya se encuentran en el sistema de reservas, tiene entonces 9 unidades de objeto de conservación A y la penalidad por una representación por debajo (recuerde que la meta cuantitativa es 10) se calculará como $11 \times (10-9)/10 = 1.1$. Así, el objeto genera una penalidad de 1.1 unidades y necesita sólo una unidad de abundancia para cumplir su meta. Al no haber unidades de planificación con un costo tan bajo, la adición de cualquiera de las unidades de planificación restantes incrementaría el costo del sistema de reservas mucho más que la ganancia en la reducción de la penalidad.

El problema se puede solucionar estableciendo un SPF o Factor de Penalidad de Especies (también conocido como Factor de Penalidad de Objeto de Conservación, vea Sección 3.2.2.4) El SPF se puede ver como una forma de distinguir el valor relativo de diferentes objetos de conservación y de conocer la importancia de que estén totalmente representados. Objetos con alto valor de conservación (pueden ser las especies amenazadas o aquellas de gran importancia social o económica) deben tener valores SPF mayores que los objetos de menor importancia. Esto significa que se tiene menor interés en comprometerlos en la red de reservas.

Para calcular el valor de la función objetivo, Marxan calculará primeramente la penalidad para aquellos objetos con una representación por debajo de la meta, multiplicará esas penalidades por el valor SPF apropiado para cada objeto, y luego sumará esos valores en todos los objetos. Todos los objetos cuyas metas cuantitativas de representación se cumplan satisfactoriamente, tendrán una penalidad de cero y por consiguiente, no incrementarán el valor de la función objetivo.

B-1.3.1 Penalidades de configuración espacial

Marxan permite que los usuarios establezcan dos limitantes espaciales en la incidencia de objetos en los posibles sistemas de reserva. Estos son, un tamaño mínimo de agrupamiento requerido antes de que las incidencias de ese objeto contribuyan a cumplir con la meta general (Vea Sección 3.2.2.5), y una distancia de separación mínima requerida entre las incidencias múltiples del mismo objeto (vea Sección 3.2.2.9). Ambas pueden especificarse en el **Archivo de Objetos de Conservación** (vea Sección 3.2.2).

Sin embargo, antes de utilizar esas características espaciales adicionales, sugerimos que los otros parámetros de Marxan hayan sido comprobados y ajustados, ya que ellas van a hacer considerablemente más lento al algoritmo.

Al calcular la penalidad inicial para un objeto de conservación que tiene requerimientos espaciales, Marxan emplea un método diferente al aplicado para la penalidad básica del objeto de conservación. El método 'ávido' ha sido reemplazado por un método de mejoramiento iterativo. Un objeto de conservación, que tiene una regla espacial de agrupamiento, tiene un segundo valor de meta que especifica el tamaño de agrupamiento más pequeño que se tomará en cuenta para cumplir la meta principal (vea Sección 3.2.2.5). Si un grupo de unidades de planificación contigua contiene un objeto de conservación, pero no tan representado como el

tamaño mínimo de agrupamiento, entonces el sistema de reservas es penalizado para ese objeto de conservación como si ninguna de esas unidades de planificación contuviera dicho objeto de conservación. Pueden también aplicarse penalidades más avanzadas para agrupamientos de menor tamaño (vea Sección3.2.1.7.3). Por ejemplo, en lugar de no tomar en cuenta, el agrupamiento para determinar la cantidad de los objetos de conservación, ese agrupamiento puede contar con la mitad de su valor. Otra alternativa se basa en el hecho de que el factor de escala para la cantidad con la que contribuye el agrupamiento es igual a la proporción del tamaño mínimo de agrupamiento que se cumple. En este caso, si el tamaño de agrupamiento es la mitad del agrupamiento mínimo, entonces la cantidad contribuye a los factores de conservación con la mitad de su valor. En todos los casos, si el agrupamiento es mayor que el tamaño mínimo de agrupamiento, no hay penalidad y la cantidad contribuye directamente a la cantidad total de ese objeto de conservación.

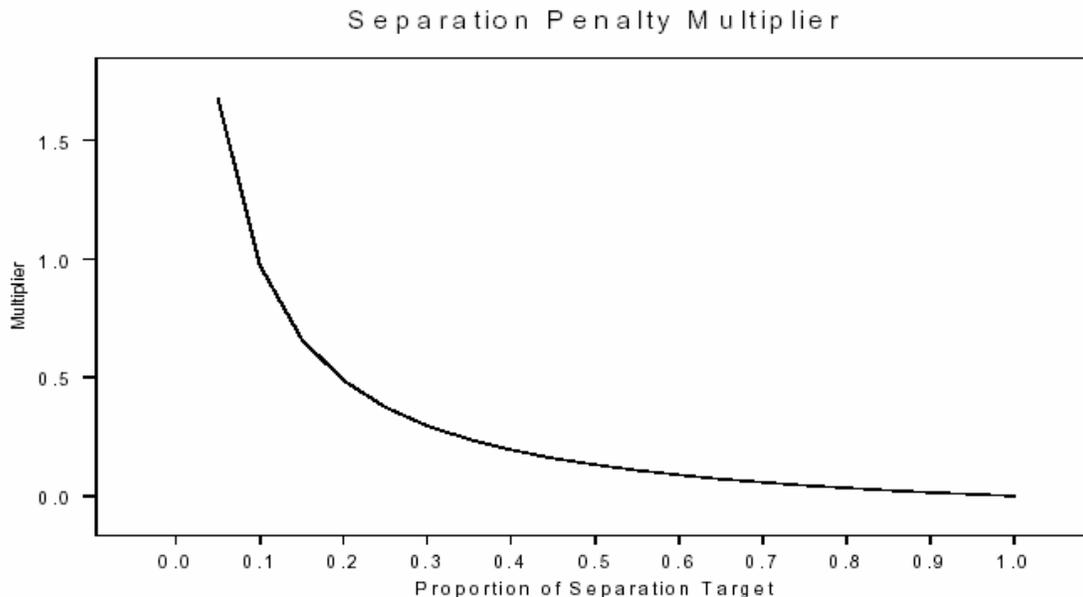
Esas dos reglas alternativas pueden ser útiles para favorecer el agrupamiento de determinados sitios, pero son inconsistentes con el concepto de tamaño mínimo de agrupamiento relacionado directamente con el tamaño mínimo de población viable. Esto se debe a que ambas tienden a cumplir algunas de las metas cualitativas de un objeto de conservación, con fragmentos de ese objeto de conservación (posiblemente incluidos en agrupamientos de otros objetos de conservación de gran tamaño). Pueden ser utilizados donde se hace necesario el agrupamiento pero el tamaño mínimo de agrupamiento no es tan rígido como el tamaño mínimo de población viable.

La regla de separación funciona de forma que el algoritmo busque el objeto de conservación dado en todas las unidades de planificación y determine si hay suficientes de ellos, a la distancia de separación requerida. La distancia de separación mínima debe ser especificada para cada objeto de conservación que tiene una meta para la cantidad mínima de incidencias individuales (vea Sección3.2.2.9). Marxan determina la cantidad de unidades de planificación, que contienen ese objeto, que se separan, mutuamente, al menos, por la distancia especificada de línea recta. Ese número es denominado el conteo de separación. Si un objeto de conservación tiene un conteo de separación menor que la cifra meta de incidencias individuales, entonces se adiciona una penalidad. Esa penalidad se multiplica por la penalidad base para el objeto de conservación (la penalidad se aplica donde el objeto de conservación no está suficientemente representado dentro del sistema de reservas) y luego se adiciona a la penalidad total del objeto de conservación.

Esa función de penalidad de separación es:

$$penalty = \frac{1}{7 * C_p + 0.2} - \frac{1}{7.2} .$$

Aquí C_p es el conteo de separación para el objeto de conservación como una fracción de la meta de incidencias individuales (meta/conteo de separación). Si el conteo de separación es cero, entonces se calcula la penalidad basada en un conteo de separación de 1/conteo de separación meta. Los valores 7 y 0.2 se escogieron después de una fase de experimentación para ofrecer un multiplicador de penalidad de separación que fuese aplicable bajo diferentes condiciones. A medida que aumenta el conteo de separación desde 1/conteo de separación meta hasta 1, esta penalidad quedaría como:



Multiplicador de penalidad de separación como una función de la proporción de la meta de separación cumplida. Note que las metas de separación normalmente son enteros de bajo valor, así que sólo pueden tomarse algunos valores del multiplicador.

B-1.4 Penalidad de Umbral de Costo

Marxan por lo general se utiliza para encontrar un sistema de reservas de costo mínimo. La penalidad de umbral de costo se ha incluido en la función objetivo para que esta pueda analizar una versión inversa del problema, por ejemplo, encontrar el sistema de reservas con la mejor representación para todos los objetos de conservación dado un costo máximo del sistema de reservas. A esto por lo general, se le ve como un problema de cobertura máxima. Este es un problema muy diferente

desde el punto de vista filosófico, por tanto, la penalidad de umbral de costo es un intento de tratar de resolver este problema dentro del marco existente del conjunto mínimo establecido. La mera adición de un umbral de costo a la función objetivo, no logrará que Marxan solucione de manera óptima un problema de cobertura máxima.

La penalidad de umbral de costo funciona aplicando una penalidad a la función objetivo si el costo total del sistema ha aumentado por encima del umbral deseado. El umbral se basa en el costo del sistema únicamente y no incluye los costos asociados a la longitud de frontera. El valor adicionado a la función objetivo, se calcula como la cantidad con que se ha excedido el umbral, multiplicado por la penalidad de umbral de costo. La penalidad depende de la fase del algoritmo templado (ej, ¿cuándo el sistema se da como una proporción dentro del proceso de templado?) y se calcula de la siguiente manera:

$$\text{Cost Threshold Penalty} = (\text{amount over threshold}) \times (Ae^{bt} - A)$$

Aquí t es el tiempo durante la corrida que debe estar entre 0 (inicio de la corrida) y 1 (fin de la corrida). A y b son parámetros de control. El coeficiente b controla la forma de la curva de penalidad, en otras palabras, con que frecuencia se aplica la penalidad (si es alta, la penalidad variará muy poco hasta avanzada la ejecución). El coeficiente A controla el tamaño de la penalidad. Si es alta, recibirá una penalidad muy alta por exceder el umbral. Un valor de A más bajo puede permitir que se exceda ligeramente el umbral. La penalidad siempre comienza con 0 cuando t es cero. El valor tanto para A como b requerirá de alguna experimentación para que sean establecidos adecuadamente y ambos pueden ser modificados en el tabulador 'Cost Threshold' de **Inedit** (vea Sección 3.2.1.6). En **Inedit**, A y b se listan como 'Penalty Factor A' and 'Penalty Factor B' (factor de penalidad A y factor de penalidad b). En el archivo 'input.dat' aparecen como 'THRESHPEN1' y 'THRESHPEN2'. El uso apropiado de esa opción requiere de experimentación y sólo se recomienda su uso para usuarios avanzados.

La manera en que la Penalidad de Umbral de Costo influye en la corrida de Marxan, depende del método de optimización utilizado. Si se emplea el Templado Simulado, el sistema podrá sobrepasar el umbral de costo y la penalidad lo disminuirá de nuevo al final. La aplicación de la penalidad va a variar según los parámetros descritos anteriormente. Esto no limitará muchas opciones permitiendo así que Marxan aún encuentre soluciones efectivas razonables. Si por otra parte, se utiliza un algoritmo heurístico o un mejoramiento iterativo, Marxan simplemente dejará de adicionar

unidades de planificación en el sistema cuando se alcance el umbral. Si se usa el mejoramiento iterativo seguido del templado simulado, Marxan va a eliminar las unidades de planificación hasta que se alcance el umbral de costo. Ambos escenarios tiene posibilidades de resultar en soluciones sub-óptimas.

B-2 Métodos de Optimización

Como se describe en este manual, Marxan puede utilizar diferentes algoritmos para tratar de mejorar el valor de la función objetivo de los sistemas de reservas potenciales. El principal y más poderoso de esos algoritmos es el Templado Simulado. Marxan ofrece también la opción de utilizar heurísticos comunes, menos sofisticados e incluso las técnicas básicas del mejoramiento iterativo. Es posible también utilizar esos métodos en combinación con otros, por ejemplo, se recomienda generalmente el templado simulado seguido del mejoramiento iterativo. En la Sección 3.2.1.2.2 puede encontrar más información referente a los métodos de optimización que se pudieran utilizar bajo diferentes condiciones. En esta sección del apéndice se describen los detalles técnicos sobre el funcionamiento de cada uno de los métodos y los parámetros que se pueden emplear para controlar su funcionamiento.

B-2.1 Templado Simulado

El templado simulado se basa en el mejoramiento iterativo pero con una aceptación estocástica (aleatoria) de movimientos errados para evitar quedar atascados prematuramente en un mínimo local del valor de la función objetivo. Un mínimo local ocurre en el punto donde la simple adición al sistema de una unidad de planificación favorable, o la eliminación de una unidad de planificación no favorable, no resulta suficiente para mejorar el valor de la función objetivo. Ese mínimo local puede ocurrir en un valor de función objetivo muy diferente del óptimo verdadero.

El templado simulado deriva su nombre de una técnica metalúrgica que implica el calentamiento y enfriamiento controlado de un material para reducir sus defectos (templado). Inicialmente las altas temperaturas hacen que los átomos estén libres y tengan movimientos aleatorios. El enfriamiento lento, entonces, incrementa las posibilidades de los átomos de encontrar configuraciones con menos defectos. Por analogía, la eficiencia se logra en una red de áreas de conservación al tolerar, a inicios del proceso de selección, cambios que implican un costo adicional en la red. Sin embargo, a medida que el proceso avanza, la temperatura se va enfriando y sólo se aceptan los cambios positivos o efectivos en el diseño. Esto permite que el sistema escape a los mínimos locales a inicios de las rondas de muestreo y el

mejoramiento progresivo hacia las soluciones efectivas en rondas de muestreo posteriores. Otra analogía útil es imaginar las soluciones como una zona montañosa donde la punta de la montaña más alta es la solución óptima. Si se comienza a escalar hacia el pico más alto desde la base de la montaña, pero sólo se permite subir, rápidamente quedará atrapado en la cima de una de las estribaciones (equivalente al mínimo local). Si, por el contrario, se le permite bajar de vez en cuando para poder cruzar los valles, al final podrá alcanzar los picos más altos.

En Marxan, el procedimiento de templado simulado se ejecutará para una cantidad de iteraciones definidas por el usuario. En cada iteración, una unidad de planificación se escoge al azar y puede o no estar ya incluida en el sistema de reservas. Se evalúa el cambio en el valor de la función objetivo del sistema de reservas, que ocurrirá si esa unidad de planificación fuese adicionada o eliminada del sistema. Ese cambio se combina con un parámetro denominado la 'temperatura' y luego se compara con un número aleatorio uniforme. La unidad de planificación puede entonces ser adicionada o eliminada del sistema en dependencia de esa comparación.

La temperatura comienza con un valor alto y va disminuyendo durante el algoritmo. Cuando la temperatura es alta, al principio del procedimiento, tanto los cambios acertados como los inadecuados pueden ser aceptados o rechazados. A medida que la temperatura disminuye, se reducen las posibilidades de aceptar un cambio errado, hasta que, al final, sólo se aceptan los cambios acertados. Para hacerlo más simple, el algoritmo debe concluir su corrida antes de que pueda aceptar únicamente cambios acertados y se debe proceder al mejoramiento iterativo, ya que en ese punto el algoritmo de templado simulado se comporta como un algoritmo eficiente de mejoramiento iterativo.

Existen dos tipos de templado simulado que pueden utilizarse en Marxan. Uno es el 'templado de esquema fijo' donde el esquema de templado (incluida la temperatura inicial y la tasa de disminución de temperatura) se fija antes de comenzar el algoritmo. El otro es el "Esquema de Templado Adaptable", en el que Marxan muestrea el problema y establece la temperatura inicial y la tasa de disminución de temperatura basado en su muestreo.

B-2.1.1 Templado de Esquema Adaptable

La opción de templado adaptable ofrece una forma fácil y relativamente rápida de establecer los parámetros necesarios para ejecutar el templado simulado y demanda

poco análisis previo por parte del usuario. Por tal razón, debe ser favorecido por los usuarios principiantes o aquellos que han recibido una rápida indicación de las posibles soluciones. Incluso para los usuarios de experiencia, el templado simulado será de mayor utilidad para investigaciones más profundas, comprobaciones y ensayos en el sistema que requerirían de un uso muy cuidadoso y detallado de un algoritmo con esquema fijo de templado

El algoritmo de templado de esquema adaptable comienza con un muestreo del sistema repetido varias veces (la cifra establecida de iteraciones/100). Luego establece la temperatura final de meta como el cambio mínimo positivo (ej. la menos mala) que tuvo lugar durante el periodo de muestreo. La temperatura inicial (y máxima) se establece siguiendo la siguiente fórmula:

$$T_{initial} = MinChange + 0.1 \times (MaxChange - MinChange)$$

Esto se basa en el esquema adaptable (Conolly 1990). Aquí, *T_{initial}* es la temperatura inicial. Los cambios (Mín y Máx) son los cambios errados, mínimos y máximos, que tuvieron lugar. En nuestro caso un cambio negativo es aquel que incrementa el valor de función objetivo (ej. un valor positivo). El Templado de Esquema Adaptable puede ser aplicado marcando la casilla 'Adaptive Annealing' en la ficha 'Annealing' del programa **Inedit** (vea Sección 3.2.1.3), o llevando la variable 'STARTTEMP', a cualquier valor negativo en el archivo 'input.dat' file.

B-2.1.2 Templado de esquema fijo

Con el templado de esquema fijo los parámetros que controlan el esquema de templado son fijados por el usuario para cada implementación del algoritmo. Esto se hace generalmente probando el algoritmo con diferentes parámetros. Las pruebas deben incluir el análisis de los resultados finales y el seguimiento del progreso en las corridas individuales. Los esquemas de templado que surgen de procesos de esquema fijo bien comprobados, por lo general son superiores al Templado de Esquema Adaptable, y el procesamiento será más rápido ya que hay mucho menos en el camino de las corridas iniciales. No obstante, se requiere de alguna habilidad para establecer los esquemas. Por esta razón, se examina aquí en detalles.

B-2.1.3 Configurar un esquema fijo de templado

Establezca primero la "verbosidad" del algoritmo como "Progreso Detallado" (vea Sección 3.2.1.5.1) para que pueda ver como funciona el templado simulado.

Cuando establecemos un esquema fijo de templado, los dos parámetros que necesitan ser establecidos son la temperatura inicial y final. La temperatura final se

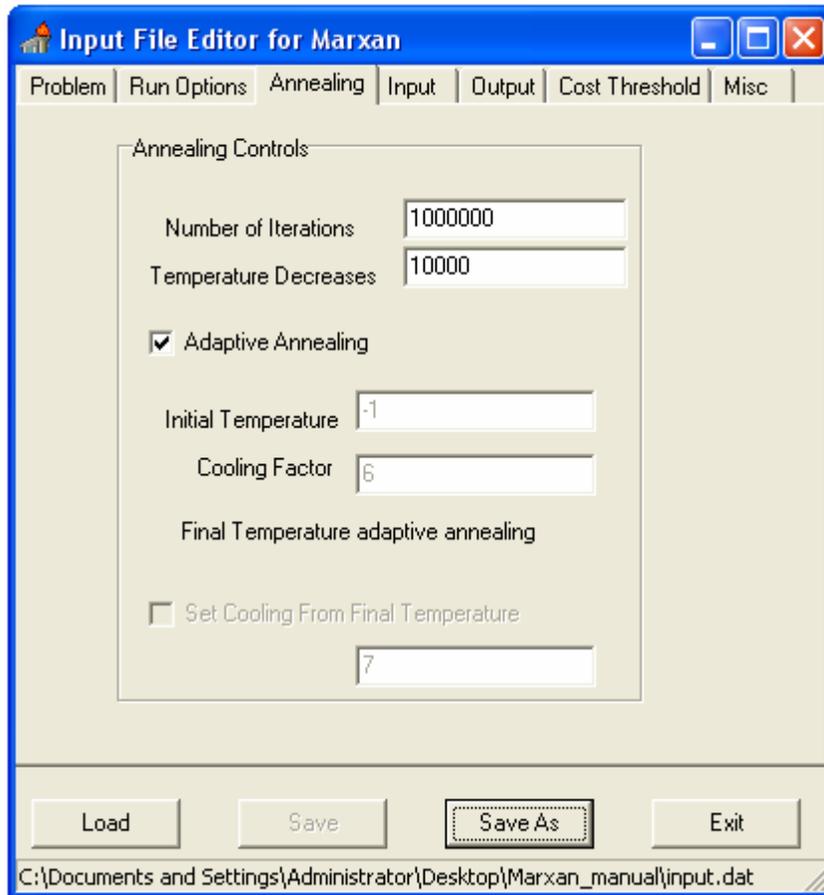
establece escogiendo un valor apropiado para la función de enfriamiento. Si la temperatura final es muy baja, entonces el algoritmo pasará mucho tiempo “estancado” en un mínimo local que no es capaz de mejorar el sistema ni continuar intentando. Si la temperatura final es muy alta, entonces una parte importante del sistema de refinamiento no se completará y el sistema de reservas se obtendrá mediante un esquema de mejoramiento iterativo si sigue a un templado simulado. Como el mejoramiento iterativo sólo encontrará mínimos locales, no parece ser una buena solución. Si la temperatura inicial es muy alta entonces el sistema consumirá mucho tiempo en las altas temperaturas, aceptando movimientos errados, y menos tiempo para realizar la mayor parte del trabajo de templado.

La mejor manera de tener una idea general de lo que deben ser los dos parámetros, es correr el algoritmo con muchos valores diferentes. Analice el valor del sistema actual regularmente, para detectar cuando se logra el equilibrio a diferentes temperaturas, cuales son y cuándo no habrá más cambios o mejoras del sistema. Esto facilita establecer una temperatura final provisional y proporciona además un estimado de lo que pudieran ser temperaturas iniciales razonables.

Las pruebas se pueden ejecutar analizando el resultado final de múltiples corridas y diferentes parámetros, pero con menos cantidad de iteraciones. Sin embargo, esas ejecuciones cortas sólo pueden mostrar como comienzan las cosas. Para ver el final (o “cola”), se requerirá de la cantidad total de iteraciones. Las pruebas se pueden ejecutar analizando el resultado final de múltiples y diferentes ejecuciones. Una vez que se han encontrado valores pequeños con pocas iteraciones, se necesita incrementarlos para una cantidad mayor de iteraciones. Esto se debe a que el tiempo consumido en las temperaturas más bajas o críticas es importante y conllevará a la búsqueda de buenos parámetros. Al aumentar la longitud del algoritmo, se consumirá mayor tiempo que el deseado a esas temperaturas. La mejor manera de proceder es mantener invariable la temperatura final e incrementar el nivel de la temperatura inicial de manera que destine el mismo tiempo en valores más bajos pero le permita encontrar el espacio de solución a una extensión mayor. Para una corrida corta, generalmente es mejor ejecutar el sistema a una temperatura crítica por todo el tiempo que sea posible. Para una corrida más larga (más iteraciones) es conveniente incrementar el rango de temperaturas utilizado. Sin embargo, a la larga, la mayoría de los usuarios utilizarán a conveniencia tanto la temperatura inicial como la final para ajustarse mejor al número total de iteraciones.

Si se desea utilizar un esquema fijo de templado, tanto la temperatura inicial como el factor de enfriamiento (que controla las temperaturas finales) serán establecidos en

el tabulador de **Inedit** denominado 'Annealing' (vea abajo). De manera alternativa, pueden establecerse directamente en el archivo 'input.dat' utilizando los nombres de variables 'STARTTEMP' y 'COOLFAC' (vea Sección 3.2.1.3.1).



La ficha 'Annealing' del programa **Inedit**. Aquí se pueden establecer los diferentes parámetros que controlan el esquema de templado.

B-2.2 Mejoramiento iterativo

El mejoramiento iterativo es un método simple de optimización. No es muy poderoso y no tiene mucho sentido ejecutar el mejoramiento iterativo por separado, pero sí puede ser utilizado de manera muy provechosa para ayudar en los resultados del templado simulado. Existen tres tipos básicos de mejoramiento iterativo que pueden ser utilizados por Marxan. Ellos difieren en el establecimiento de los posibles cambios considerados en cada paso. Cada uno comienza con una solución 'semilla'. Puede ser cualquier sistema de reservas que incluya algunas, todas o ninguna unidad de planificación. Resulta útil tomar el resultado final de otro algoritmo como el templado simulado, como solución inicial para el mejoramiento iterativo. En ese caso, el algoritmo de mejoramiento iterativo se utiliza únicamente para asegurar que no hay más posibilidades de otros mejoramientos simples.

En cada iteración, el algoritmo analizará un cambio aleatorio para ver si el valor de la función objetivo mejoraría si se realizara el cambio. Si el cambio sí mejora el sistema, entonces se realiza, de lo contrario, otro cambio, no comprobado hasta el momento, se prueba de forma aleatoria. Este proceso continúa hasta que se han analizado todos los posibles cambios y ninguno mejorará el sistema. El sistema de reservas resultante, está por consiguiente, en un óptimo local (que puede o no ser particularmente bueno en su totalidad, dependiendo de lo que precedió al mejoramiento iterativo).

Los tres tipos básicos de mejoramiento iterativo difieren en los tipos de cambios que se van a analizar. El más simple se denomina 'mejoramiento iterativo normal' y los únicos cambios que se consideran son la adición o eliminación de cada unidad de planificación del sistema de reservas. Este es el mismo 'conjunto de movimientos' considerado por el algoritmo "ávido" y el templado simulado.

El segundo tipo de mejoramiento iterativo es denominado 'de reemplazo' y va a seleccionar unidades de planificación al azar, si el sistema mejora con la adición o eliminación de la unidad de planificación seleccionada, entonces se realiza dicha acción, en caso contrario, se analiza un intercambio. Si la unidad de planificación seleccionada ya se encuentra en el sistema de reservas, entonces los cambios valorados son la eliminación de esa unidad de planificación pero la adición de otra en otro lugar. Si la unidad de planificación seleccionada no se encuentra en el sistema de reservas, el cambio sería agregarla al sistema de reservas pero eliminar una que ya está dentro del sistema. Los posibles 'reemplazos' se contemplan en orden aleatorio, hasta que se encuentre uno que mejore el sistema. Este proceso continúa

hasta que se hayan analizado todos los reemplazos posibles. Esta es una opción mucho más lenta ya que se pudieran analizar gran cantidad de posibilidades.

El tercer tipo es el denominado 'de dos pasos', en este método, además de comprobar cada unidad de planificación (en orden aleatorio) para analizar si se debe agregar o eliminar del sistema, se consideran cada una de las posibles combinaciones de dos pasos. Esos cambios incluyen, la adición o eliminación de la unidad de planificación escogida junto con la adición o eliminación de unidades de planificaciones alternas. La cantidad de movimientos es incluso mayor que en el método de 'reemplazo', por lo que también requiere de mucho tiempo. Este método sólo es conveniente para pequeñas cantidades de unidades de planificación.

Hay una cuarta opción que es correr primeramente el método normal, obtener un buen óptimo local y luego correr el método de 'dos pasos'. Esto resulta más rápido que la ejecución independiente del método de 'dos pasos', ya que la cantidad de mejoramientos que este método encuentra, debe ser menor después de haber corrido un algoritmo de mejoramiento sobre la solución 'semilla'.

Para poner en práctica cualquiera de las cuatro opciones de mejoramiento iterativo en una ejecución de Marxan, se pueden seleccionar en la ficha 'Run Options' (opciones de ejecución) de **Inedit** (vea Sección 3.2.1.2), o especificarlas directamente en el archivo 'input.dat', empleando la variable 'ITIMTYPE' (vea Sección 3.2.1.2.2).

B-2.3 Otros Algoritmos Heurísticos

Heurística es el término utilizado para nombrar una clase de algoritmos que históricamente se ha utilizado para el problema de la selección de reservas (Pressey et al. 1993). Ellos surgen de un intento de automatizar el proceso de selección de reservas copiando la manera en que una persona lo escogería 'a mano'.

Existen algunos métodos heurísticos fundamentales a partir de los cuales se han hecho variaciones. Ellos son el heurístico 'ávido', el heurístico de rareza y el heurístico de irremplazabilidad. Todos los heurísticos agregan unidades de planificación al sistema de reservas de manera secuencial. Comienzan con un sistema de reservas vacío y adicionan unidades de planificación al sistema hasta que se alcance un criterio de detención. Los criterios de detención siempre sugieren que ningún sitio sin reservar mejorará el sistema. Como se verá, la definición tiene dos significados ligeramente diferentes. Los heurísticos pueden estar seguidos de un

algoritmo de mejoramiento iterativo para asegurar que ninguna de las unidades de planificación adicionales queden obsoletas a consecuencia de futuras adiciones.

El heurístico 'ávido' puro es relativamente fácil de comprender por usuarios no técnicos y generalmente es bueno utilizarlo como ejemplo de punto de partida. Aunque no es el mejor método heurístico, es fiable y opera con problemas de gran complejidad basándose en el mero concepto de que adicionando de manera iterativa las unidades de planificación, se mejorará al máximo la función objetivo. Se recomienda el uso de los demás heurísticos para la generación de soluciones rápidas, en aras de explorar ideas de diseño rápidamente aunque no con efectividad. Por ejemplo, las diferentes soluciones 'ávidas' (que son fáciles de comprender), pueden ser comparadas posteriormente con las soluciones derivadas mediante el templado simulado, que por lo general será superior (aunque más difícil de comprender).

B-2.3.1 El heurístico 'Ávido'

Los heurísticos 'Ávidos' son aquellos que intentan mejorar un sistema de reservas tan rápido como sea posible. El heurístico agrega cualquier sitio donde el objeto de conservación esté menos representado. Este heurístico, por lo general, se denomina el heurístico de riqueza y el sitio con el objeto de conservación menos representado es el sitio de mayor riqueza. Tiene como ventaja que se realizan avances inmediatos encaminados a la representación efectiva de todos los objetos de conservación (o al menos a mejorar el resultado objetivo) y opera razonablemente bien.

El resultado de un heurístico 'Ávido' no sólo incluye una lista de las unidades de planificación que conforman un sistema de reservas, sino también un orden de prioridad para esas unidades de planificación. Esta prioridad se basa en el orden en que las unidades de planificación fueron adicionadas a la solución. Esto pudiera resultar útil si no hay suficientes recursos para obtener o rechazar el sistema de reservas completo en un momento determinado. En esos casos, se pueden conservar las zonas por orden de prioridad y el sistema de reservas resultante será todavía bueno, con relación a su costo. Analizados desde esta perspectiva pueden ser muy útiles.

Los heurísticos pueden ser divididos posteriormente según la función objetivo que utilicen. Los dos utilizados por Marxan se denominan el Heurístico de Riqueza y el Heurístico Ávido Puro. Esos heurísticos se describen a continuación.

B-2.3.1.1 Riqueza

Al utilizar este heurístico cada unidad de planificación recibe dos resultados; el valor de conservación de la unidad de planificación y el costo de la unidad de planificación. El costo siempre es el costo especificado para esa unidad de planificación más el cambio potencial en la longitud de frontera modificada. El valor de conservación es la suma de la falta de representatividad de los objetos de conservación presentes en una unidad de planificación. La falta de representatividad de un objeto es sencillamente como estaría mejor representado un objeto de conservación si esa unidad de planificación se adicionara al sistema. Si un objeto de conservación ha cumplido su meta cuantitativa, no contribuye a esa sumatoria. La riqueza de la unidad de planificación es la contribución que hace en la representación de todos los objetos de conservación divididos por su costo.

B-2.3.1.2 Ávido Puro

El heurístico ávido puro valora las unidades de planificación según la manera en que ellas modifiquen la función objetivo de Marxan. Esto es similar, pero no igual, al heurístico de riqueza. Al utilizar el sistema de penalidad de objeto de conservación, que se utiliza con el templado simulado, el heurístico ávido puro difiere un poco del algoritmo de riqueza. Pudiera no continuar hasta que estén representados todos los objetos de conservación. Pudiera suceder que el beneficio de elevar un objeto de conservación hasta su meta cuantitativa sea sobrepasado por el costo de la unidad de planificación que tendría que adicionar. El algoritmo ávido puro emplea la función objetivo usual de Marxan, que le permite analizar las longitudes de frontera del sistema de reservas así como otras consideraciones avanzadas, como aquellas relacionadas con un agrupamiento de objeto de conservación y las reglas de separación mínima.

B-2.3.2 Algoritmo de Rareza

Las unidades de planificación escogidas por la heurística ávida, estarán guiadas, por lo general, por la presencia de objetos de conservación relativamente comunes. Las primeras unidades de planificación adicionadas son aquellas con gran cantidad de objetos de conservación, resultando, con frecuencia, en la selección de objetos de conservación que son bastante comunes en el conjunto de información. Los algoritmos de rareza funcionan con el concepto de que un sistema de reservas debe ser diseñado para asegurar que los objetos de conservación relativamente raros se reserven primero, antes de centrarse en los demás objetos de conservación más comunes. Al desarrollar Marxan, se estudiaron muchos algoritmos de rareza, aunque

ellos tienden a funcionar de manera similar. Aquellos disponibles para Marxan se han denominado; Rareza Máxima, Mejor rareza, Rareza Promedio, y Rareza Sumada.

La rareza de un objeto de conservación es su cantidad total en todas las unidades de planificación. Por ejemplo, puede ser la cantidad total de un tipo de vegetación determinado disponible en hectáreas. Aquí hay un problema potencial, ya que las rarezas para diferentes categorías de conservación pudieran ser de un orden de magnitud diferente. Eso se puede sortear en la mayoría de los algoritmos siguientes utilizando la proporción: abundancia de un objeto de conservación en una unidad de planificación dividido por la rareza total (o cantidad total) del objeto de conservación. Como la abundancia y la rareza de un objeto de conservación son de la misma unidad, esto genera un valor adimensional.

B-2.3.2.1 Rareza Máxima

Este método da un valor a cada unidad de planificación según la formula:

$$\frac{\text{Abundancia Efectiva}}{(\text{Rareza} \times \text{Costo de la UP})}$$

Esto se basa en el objeto de planificación dentro de la unidad de planificación que tiene la menor rareza. La abundancia es qué cantidad de ese objeto de conservación se encuentra en la unidad de planificación limitada por la meta cuantitativa del objeto de conservación. Por ejemplo, suponga que la especie de más rareza de la unidad de planificación incide en más de unas 1000 hectáreas por todo el sistema. Este tiene una meta cuantitativa de 150 hectáreas de las cuales se ha cumplido con 100. En la unidad de planificación hay 60 hectáreas del objeto de conservación. El costo de la unidad de planificación es 1 (incluido tanto el costo expresado como la longitud de frontera multiplicado por el modificador de longitud de frontera). La abundancia efectiva es entonces de 50 hectáreas (los 10 extras no cuentan contra la meta cuantitativa) y la medida para esa unidad de planificación es $50 / (1000 \times 1) = 0.05$. Note que la rareza máxima se basa en las especies más raras de la unidad de planificación y que la rareza en sí es un valor dimensionado. Por esa razón, se espera que el algoritmo tenga un funcionamiento pobre cuando se encuentra más de un tipo de objeto de conservación en el conjunto de información (ej. tipos de vegetación y especies de la fauna). Después de calcular la rareza máxima para cada unidad de planificación, se adiciona al sistema de reservas aquella con mayor valor, eliminando de manera aleatoria las limitaciones.

B-2.3.2.2 Mejor Rareza

La Mejor rareza es muy similar al heurístico de rareza máxima descrito anteriormente. Se utiliza la misma fórmula:

$$\frac{\text{Abundancia Efectiva}}{(\text{Rareza} \times \text{Costo de la UP})}$$

La diferencia entre las dos es que cuando se utiliza mejor rareza, el objeto de conservación en el que se basa el valor es aquel con la mejor proporción (Abundancia Efectiva/Rareza) y no aquel con el resultado de mejor rareza. Esto evita el problema de dimensiones pero, por otra parte, funciona de manera similar.

B-2.3.2.3 Rareza Sumada

La rareza sumada toma la suma de la relación Abundancia Efectiva/Rareza para cada objeto de conservación en la unidad de planificación y luego divide esa suma por el costo de la unidad de planificación. Por tanto, en esta medida hay un elemento tanto de riqueza como de rareza. Aquí es posible que una unidad de planificación con muchos objetos de conservación obtengan resultados mayores a una unidad con un objeto de conservación único, aunque raro. La fórmula empleada es:

$$\frac{\sum_{\text{Obj. Cons.}} \frac{\text{Abundancia efectiva}}{\text{Rareza}}}{\text{Costo de la UP}}$$

B-2.3.2.4 Rareza Promedio

La rareza promedio es la misma que la rareza sumada exceptuando que la suma se divide en el costo multiplicado por la cantidad de objetos de conservación representados en la unidad de planificación. Al dividirlo por esta cifra, el heurístico tenderá a dar más peso a los objetos de conservación raros. La fórmula empleada es:

$$\frac{\sum_{\text{Obj. Cons.}} \frac{\text{Abundancia efectiva}}{\text{Rareza}}}{\text{Costo} \times \text{Número Obj. Cons.}}$$

B-2.3.3 Irremplazabilidad

La irremplazabilidad captura algunas de las ideas de los heurísticos de rareza y ávido. La irremplazabilidad funciona analizando cuán importante es cada unidad de planificación para lograr la meta cuantitativa para un objeto de conservación determinado. Esto se basa en la idea de una memoria intermedia de objetos de conservación. La memoria es la cantidad total de un objeto de conservación menos la meta cuantitativa para ese objeto de conservación. Si la meta es tan grande como la cantidad total, tiene entonces una memoria de cero y es necesaria cada unidad de planificación que contiene ese objeto de conservación. La irremplazabilidad de una unidad de planificación para un objeto de conservación específico es la siguiente:

$$Irreplaceability \begin{cases} \frac{\text{Buffer} - \text{Effective Abundance}}{\text{Buffer}}, & \text{Buffer} > 0 \\ 0, & \text{Buffer} = 0 \end{cases}$$

Note que si la memoria es cero entonces la irremplazabilidad se hace cero. Si una unidad de planificación es esencial, tendrá también valor de irremplazabilidad de cero. Un valor cercano a 1 indica que la unidad de planificación no es realmente necesaria. Hay dos formas en que se utiliza esta medida: La irremplazabilidad del producto y la irremplazabilidad sumada (vea abajo).

B-2.3.3.1 Producto

La irremplazabilidad para cada objeto de conservación se multiplica para dar un valor a la unidad de planificación entre 0 y 1, con 0 significa que la unidad de planificación es esencial para uno o más objetos de conservación. El número se sustrae de 1, de manera que un valor alto es mejor que uno bajo. El valor de ese producto no puede superar al valor de un objeto de conservación individual. Es similar al heurístico de rareza ya que va a tender a seleccionar las unidades de planificación basándose en su contenido de objetos de conservación difíciles de representar.

B-2.3.3.2 La Irremplazabilidad Sumada

Al utilizar este heurístico, la irremplazabilidad se sustrae de 1 para generar un valor entre 0 y 1, donde se necesita una unidad de planificación de alto valor para objetivos de conservación y no una de bajo valor. Esos valores se adicionan en todos los objetos de conservación. De esta forma, es menos sensible a elementos externos o la información débil. Esta sumatoria significa que la cantidad de objetos de conservación es importante y se relaciona con el heurístico de irremplazabilidad producto, de la misma manera que el heurístico de rareza sumada se relaciona con el heurístico de mejor rareza.

Apéndice C – Asesoría sobre el desarrollo de archivos de entrada de Marxan y muestra de los resultados en SIG

Hay muchas maneras de crear los archivos necesarios para la corrida de Marxan. El método que se escoja dependerá de sus habilidades, del software disponible y la preferencia personal. Algunos software son gratis y otros relativamente caros. Nuestro objetivo aquí es proveerlos con herramientas y recursos útiles, más que ofrecer asesoría detallada paso a paso. Casi todos los consejos se basan en ESRI ArcView 3 y ArcGIS 8 y 9, y las interfaces fáciles de Marxan; Zonación del Uso de la Tierra para la Conservación (CLUZ) y P.A.N.D.A.

C-1 Recursos

C-1.1 Software

Se necesitará como mínimo un programa de hoja de cálculo o de tipo Textpad para desarrollar, leer y manipular los archivos de entrada de texto. El sistema operativo de Windows contiene el editor de texto Notepad. Hay muchos editores de texto gratis disponibles en Internet que pueden ser reemplazados. Los programas con hojas de cálculo como Microsoft Excel y OpenOffice (una fuente alternativa gratis) son útiles para la aplicación de fórmulas y la fácil manipulación de la información. Otro programa útil para manipular los archivos de entrada es el editor de tabla C-Plan, que está disponible cuando se descarga desde <http://www.uq.edu.au/~uqmwatts/cplan.html>.

Para el análisis espacial se requiere de un software de Sistema de Información Geográfica (SIG). Los más populares son la familia de Software ESRI (www.esri.com). Ellos incluyen ArcView 3 y ArcGIS 8 o 9. Debido a la popularidad de esos programas, hay muchas extensiones que se pueden descargar para extender su funcionalidad y automatizar las funciones para una manipulación más fácil de la información. Recomendamos algunas al describir posteriormente los métodos para la creación de Marxan. Note que hay varios paquetes de software SIG disponibles para ser descargados gratuitamente como MapWindow (www.mapwindow.com), Quantum GIS (www.qgis.org), y GRASS(<http://grass.itc.it/>). Otra alternativa gratis es DIVA (<http://www.diva-gis.org/>). En la actualidad no tiene la misma funcionalidad que los ESRI, pero diversos programadores están trabajando para incrementar sus funcionalidades incluyendo las herramientas necesarias para desarrollar los archivos

de entrada de Marxan . Versiones futuras de este programa incluirán actualizaciones sobre esta situación.

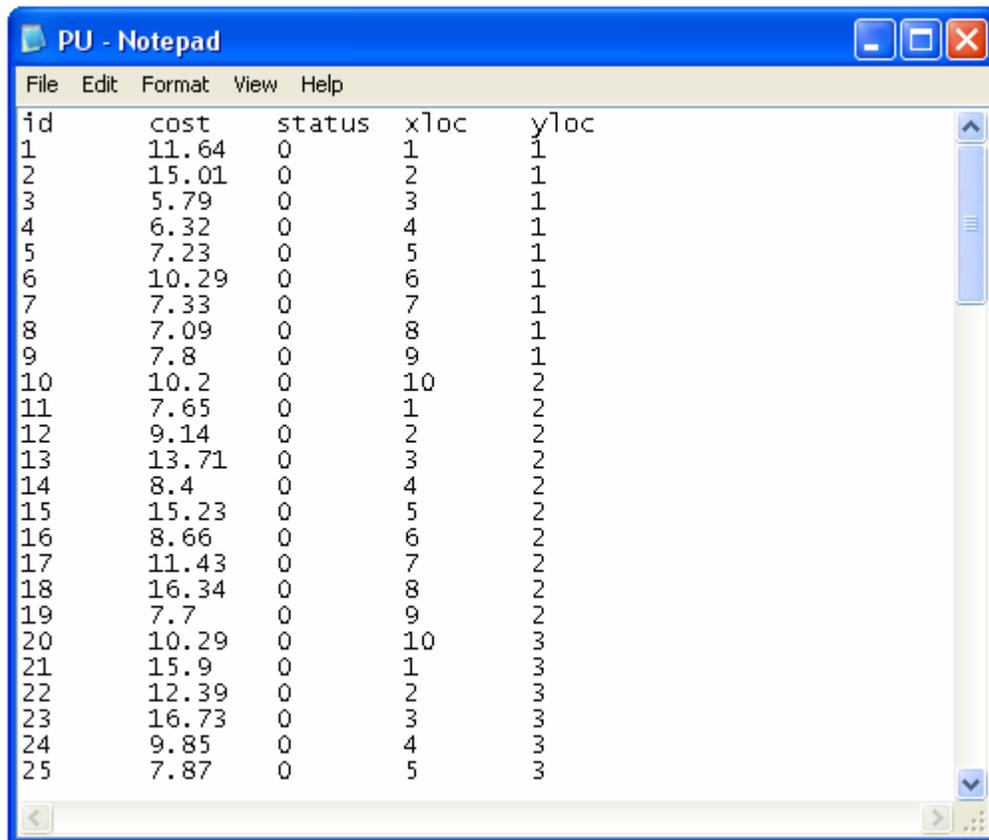
Existen también algunas herramientas de software gratis que simplifican la ejecución de Marxan y la creación de los archivos de entrada de Marxan. Tres de ellos son: CLUZ que opera mediante ArcView 3 (<http://www.mosaic-conservation.org/cluz/index.html>), C-Plan, que también opera mediante ArcView 3 (<http://www.uq.edu.au/~uqmwatts/cplan.html>) y P.A.N.D.A, que requiere ArcGIS 8 y 9 (http://www.mappamondogis.it/panda_en.htm). Hay planes futuros para que softwares similares funcionen mediante software SIG. The Nature Conservancy ha desarrollado “Sistema de Apoyo a la Toma de Decisiones GAP de Áreas Protegidas” , que facilita el análisis de Marxan, disponible desde ftp://cerp.cerppassword@ftp.tnc.org/CDSS/Protected_Area_Gap_DSS.zip. Por último, Vista, que opera mediante ArcGIS 9, no es gratis pero para la demostración gratis, vea <http://www.natureserve.org/prodServices/vista/overview.jsp>.

C-1.2 Cursos y Guías Didácticas

La Universidad de Queensland ofrece cursos sobre el uso de Marxan y la planificación sistemática de la conservación. Puede encontrar los detalles en el sitio web de Marxan <http://www.ecology.uq.edu.au/index.html?page=27710> vea también <http://www.aeda.edu.au/events>

Si no puede participar, los materiales del curso, incluida las guías didácticas (tutoriales) están disponibles para su descarga. Esto incluye asesoría detallada sobre el desarrollo de programas de entrada de Marxan. Con frecuencia, se encuentra también un curso corto sobre planificación de la conservación y software como Marxan; como parte de la Conferencia Anual de la Sociedad para la Biología de la Conservación <http://www.conbio.org/>. Para información referente a los cursos cortos anteriores (incluidas las presentaciones), las demostraciones de software (incluida la información e instrucciones descargables sobre el uso de Marxan, vea <http://www.aeda.edu.au/events>. Una instrucción útil sobre el uso de la interfase de CLUZ puede ser descargado desde <http://www.mosaic-conservation.org/cluz/tutorial.html> y una guía por pasos para la ejecución de las evaluaciones de conservación utilizando CLUZ y Marxan está disponible en <http://www.mosaic-conservation.org/cluz/steps.html>.

C-2 Crear el archivo de unidad de planificación



The screenshot shows a Notepad window with a blue title bar and standard Windows window controls. The text inside is a table with five columns: 'id', 'cost', 'status', 'xloc', and 'yloc'. The data consists of 25 rows, each representing a planning unit. The 'status' column for all units is '0'. The 'xloc' and 'yloc' columns show coordinates for each unit, ranging from 1 to 10 on the x-axis and 1 to 3 on the y-axis.

id	cost	status	xloc	yloc
1	11.64	0	1	1
2	15.01	0	2	1
3	5.79	0	3	1
4	6.32	0	4	1
5	7.23	0	5	1
6	10.29	0	6	1
7	7.33	0	7	1
8	7.09	0	8	1
9	7.8	0	9	1
10	10.2	0	10	2
11	7.65	0	1	2
12	9.14	0	2	2
13	13.71	0	3	2
14	8.4	0	4	2
15	15.23	0	5	2
16	8.66	0	6	2
17	11.43	0	7	2
18	16.34	0	8	2
19	7.7	0	9	2
20	10.29	0	10	3
21	15.9	0	1	3
22	12.39	0	2	3
23	16.73	0	3	3
24	9.85	0	4	3
25	7.87	0	5	3

Un ejemplo del **Archivo de Unidad de Planificación**.

Al generar los archivos de entrada, es necesario tener una capa espacial de unidades de planificación. Las unidades de planificación, no obstante, no necesariamente tienen que ser de forma cuadradas, cualquier división puede ser aceptada. Otros ejemplos incluyen hexágonos, parcelas catastrales y unidades hidrológicas. Se debe analizar como determinar el tipo y tamaño más apropiado de la unidad de planificación para cada aplicación específica (vea la sección de referencias). Para crear una unidad de conservación que represente entidades como fronteras catastrales o cuencas, sencillamente utilice la capa espacial existente que origina las fronteras de esos polígonos, de lo contrario será necesario crear sus propios polígonos.

Para usuarios de ArcView 3:

1. Utilice unidades de planificación existentes o desarrolle nuevas unidades. Una extensión de gran utilidad para el desarrollo de nuevas unidades es <http://arcscripts.esri.com/details.asp?dbid=14329>. Esta extensión ofrece una selección de formas que se repiten sobre una zona especificada por el usuario. Esas formas pueden ser hexágonos, cuadrículas, triángulos, círculos o puntos, y pueden generarse con cualquier orientación direccional. Como una alternativa, ET GeoWizards tiene una función para crear rejillas. www.ian-ko.com/
2. *Utilice las capas temáticas de áreas protegidas que ya existen y agregue un campo denominado "ID" a la tabla de atributos de la capa temática de áreas protegidas. Dé un número único como identificador a cada área protegida. Se puede :
 - Utilizar el "update polygon theme" (actualización de temas de polígonos) en Xtools <http://arcscripts.esri.com/details.asp?dbid=11526>
 - Utilice la función "erase" (borrar) en ET geowizards www.ian-ko.com/ para borrar las unidades de planificación de las áreas donde se ubican zonas protegidas. Luego "una" (conecte) la capa de zonas protegidas con las unidades de planificación que fueron borradas.
 - Use la opción de ArcViews "union" en el Geoprocessing Wizard para combinar las unidades de planificación y las áreas protegidas en un tema nuevo. El nuevo tema dividirá cada área protegida en varios polígonos regulares enteros y algunos fragmentos. Utilice la opción "dissolve", (disolver) en el Geoprocessing Wizard para fusionar todas esas piezas en un polígono para cada área protegida. En el cuadro de diálogo, de la casilla de entrada que indica 'Seleccione un atributo para disolver escoja el campo " ID".
3. Asegúrese de que cada unidad de planificación tenga un identificador único que se pueda utilizar para el campo "id".
4. Agregue un campo "status" y utilice la capa de áreas protegidas para "seleccionar por temas" ("select by theme") las unidades de planificación que están protegidas. Después de seleccionar las unidades de planificación ya protegidas, en la misma tabla, adicione "2" en el campo "status" para esas unidades de planificación. Ese proceso se puede repetir para unidades de planificación si se desea que se excluyan determinadas zonas. En la tabla de tema adicione "3" en el campo "status".
5. Para calcular el campo de costo se necesita determinar una superficie de costo espacial.

6. El próximo paso es calcular el costo de cada unidad de planificación.
 - o Configure las propiedades de vista. El alcance del análisis debe establecerse en su capa de unidades de planificación, y el tamaño de la unidad de análisis debe establecerse lo suficientemente pequeña para que permita resultados precisos, pero al vez, debe ser lo suficientemente grande para que pueda ajustarse a la ejecución (ej. 10m o 100m, en dependencia del tamaño de las regiones)
 - o Si es una capa vectorial conviértala a ráster**
 - o Utilice 'resumen por zonas' ('summarise zones') con las capas de costo y unidades de planificación. Una la tabla de salida con la capa temática de unidades de planificación. Se pueden aplicar también estadísticas zonales***utilizando diversas extensiones creadas por los usuarios como "Mila Grid Utilities" <http://www.mila.ucl.ac.be/logistique/sig/sig-tools/milagrid/index.html>
7. Cree los campos X y Y, Xtool****
<http://arcscripts.esri.com/details.asp?dbid=14329> cuenta con una función para hacerlo.
8. Su tabla de tema debe contener todos los campos necesarios para crear el **Archivo de Unidades de Planificación**. Exporte la tabla y ábrala en un programa de notepad u hoja de cálculo para su limpieza, verifique y sálvela como un archivo de texto con la extensión ".dat".

Los usuarios de CLUZ deben seguir los pasos 1-2 mostrados arriba para crear el tema de unidad de planificación, pero los demás pasos se pueden seguir utilizando las funciones "Crear el tema de unidades desde shapefile", "Calcular el % de unidad que se incluye dentro de las unidades de planificación" y "Crear unit.dat".

Para usuarios de ArcMap 8 y 9:

1. Utilice las unidades de planificación existentes si están disponibles o desarrolle nuevas unidades. Una extensión de gran utilidad para el desarrollo de nuevas unidades es ET Geowizards <http://www.ian-ko.com/>
2. *Utilice las zonas protegidas existentes y agregue un campo denominado "ID" a la tabla de temas de zonas protegida. Asigne un número único de identificación a cada zona protegida.
 - o Utilice la función de la "capa de actualización del polígono" en Xtools Pro **** <http://www.xtoolspro.com/index.html>
 - o Utilice la función "erase" en ET geowizards www.ian-ko.com/ para borrar unidades de planificación donde existan zonas protegidas, luego

- “una” la capa de zona protegida con las unidades de planificación que fueron borradas.
- Utilice la función ArcMap “Union” para combinar las unidades de planificación y las zonas protegidas en un nuevo tema. El nuevo tema dividirá cada zona protegida en varios polígonos regulares enteros y algunos fragmentos. Utilice la función “dissolve” para fusionar todas esas partes en un polígono para cada zona protegida. Disuelva basándose en el campo “ID”.
3. Asegure que cada unidad de planificación tenga un identificador único que se pueda utilizar como campo “id”.
 4. Agregue un campo de “status” al tema de la unidad de planificación. Utilice las capas de zonas protegidas para “select by location” (seleccionar por ubicación) a las unidades de planificación que están protegidas. En la tabla de tema adicione “2” en el campo de “status” para esas unidades de planificación. Ese proceso se puede repetir para las unidades de planificación que se desea excluir. En la misma tabla adicione “3” en el campo “status”.
 5. Para calcular el campo de costo se necesita una superficie de costo espacial mínimo.
 - Si es una capa de vector convierta a ráster**
 - Utilice “estadísticas zonales” *** con la capa de costo y las unidades de planificación. Junte la capa de costo con la capa de unidad de planificación.
 6. Crear los campos X y Y, Xtool Pro**** <http://www.xtoolspro.com/index.html> cuenta con una función para hacerlo. De lo contrario, i) convierta los objetos a puntos, ii) adicione las coordenadas de XY y iii) vuélvalos a unir.
 7. La tabla de tema debe incluir todos los campos necesarios para crear el **Archivo de Unidad de Planificación** de Marxan. Exporte la tabla y abra un programa de Notepad u hoja de cálculo para la limpieza, verifique y salve como un archivo de texto con la extensión “.dat”.

*Algunas personas no incluyen las zonas protegidas como unidades de planificación completas sino que clasifican sus unidades de planificación como protegidas o no, basándose en la proporción de la unidad de planificación que ya está protegida.

**Note que se requiere de la extensión del análisis espacial para que ArcMap y ArcView 3 generen y utilicen archivos ráster.

***Con frecuencia hay una frontera computacional para el uso de las estadísticas por zonas

****Estas funciones se pueden utilizar gratuitamente, otras no.

C-3 Crear el archivo de unidad de planificación versus objeto de conservación

pu	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	15	0	0	1	0	12	0	0	0
2	0	0	1	60	0	0	1	0	4	0	0	0
3	0	0	1	17	0	0	1	0	13	0	0	0
4	0	0	1	19	0	0	1	0	11	0	0	0
5	0	0	1	0	0	0	1	0	15	0	0	0
6	0	0	1	0	0	0	1	0	19	0	0	0
7	0	0	1	0	0	0	1	0	24	0	0	0
8	0	0	1	0	0	0	1	0	15	0	0	0
9	0	0	1	0	0	0	1	0	17	0	0	0
10	0	0	0	0	0	0	1	0	21	0	0	0
11	0	0	0	0	0	0	1	0	23	0	0	0
12	0	0	0	0	0	0	1	0	23	0	0	0
13	0	0	0	0	0	0	1	3	11	0	0	0
14	0	0	0	0	0	0	1	5	6	0	0	0
15	0	6	0	0	0	0	1	2	4	0	0	0
16	0	1	0	0	0	0	1	3	9	0	0	0
17	0	0	0	0	0	0	1	4	1	0	0	0
18	0	0	0	0	0	0	1	2	5	0	0	0
19	0	0	0	0	0	0	1	1	3	0	0	0
20	0	0	0	0	0	0	1	3	2	0	0	0
21	0	0	0	0	0	0	1	1	11	0	0	0
22	0	0	0	0	0	0	1	1	2	0	0	0
23	0	0	0	0	0	0	1	0	2	0	0	0
24	0	0	0	0	0	0	1	0	0	0	0	0
25	0	0	0	0	0	0	1	0	0	0	0	0
26	0	0	0	0	0	0	1	0	0	0	0	0

Un ejemplo del **archivo de unidad de planificación versus el archivo de objeto de conservación** en forma de tabla.

species	pu	amount
1	29	1
1	31	1
1	45	1
1	51	1
1	68	1
1	69	1
1	71	1
1	83	1
1	91	4
1	95	4
2	44	1
2	52	1
2	54	1
2	64	3
2	70	7
2	84	15
2	98	21
3	20	1
3	24	1
3	38	1
3	45	1
3	48	1
3	58	1
3	64	1
3	73	1
3	99	1
4	2	1

Un ejemplo de la forma relacional del **Archivo de Unidad de Planificación versus objeto de conservación**.

Hay dos formatos del **Archivo de Unidades de Planificación versus Objetos de Conservación**: tabular y relacional. El formato relacional es necesario para las versiones más recientes de Marxan y permite cálculos más rápidos. Si se utiliza el relacional, recomendamos primero crear una versión tabular del archivo y luego convertir ese archivo a relacional utilizando el programa que está incluido cuando se descargan las versiones más recientes de Marxan desde: <http://www.ecology.uq.edu.au/index.html?page=27710> o el Editor de Tabla: <http://www.uq.edu.au/~uqmwatts/cplan.html> que permite que los archivos se puedan importar (.csv o .dbf pudieran requerir una conversión) y luego exportarlos a relacional utilizando el comando “save as Marxan file” (guardar como archivo de Marxan). Esos pasos deberán ser aplicados constantemente a cada objeto de conservación utilizado en el análisis.

Para Usuarios de ArcView 3

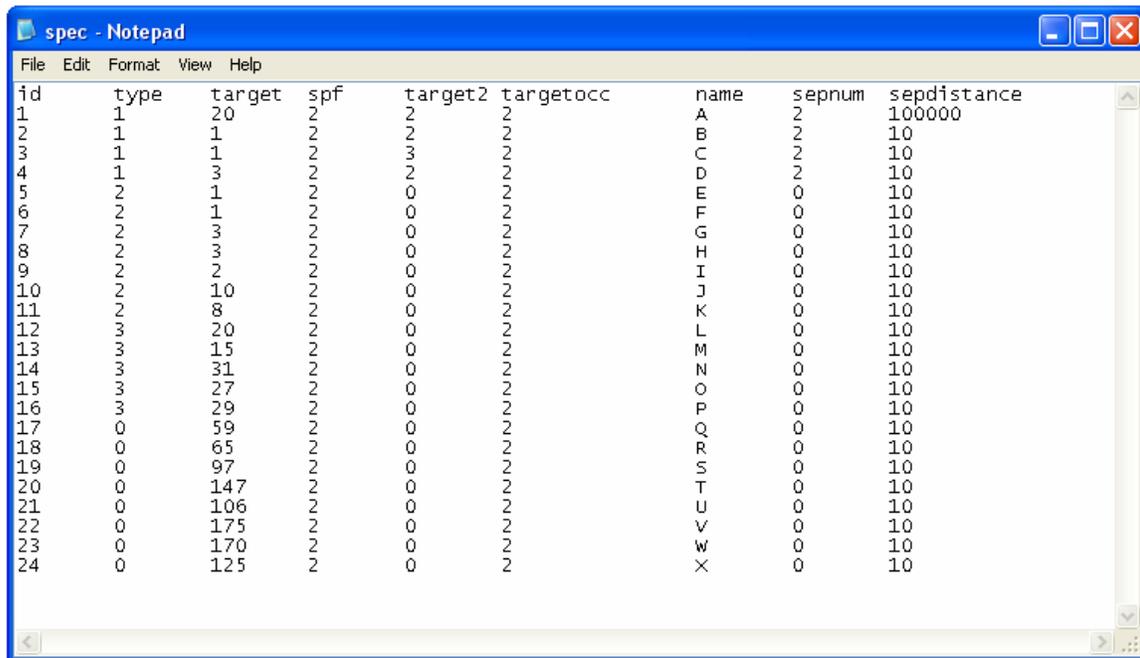
Existen funciones para que los usuarios de CLUZ puedan crear una nueva tabla de unidad de planificación versus objeto de conservación, importar información desde las tablas o archivos existentes y exportar esa información en el formato correcto de Marxan, de lo contrario:

1. Haga una copia de la capa de unidades de planificación y elimine todos los campos excepto ID. No se quieren tabular objetos que ya se tienen.
2. Establezca propiedades de vista. La extensión del análisis debe establecerse a la capa temática de unidad de planificación, y el tamaño del análisis de coordenadas debe establecerse a un valor lo suficientemente bajo para lograr resultados precisos y, a su vez, ser lo suficientemente grande para que se ajuste a la corrida. (por ejemplo, 10m o 100m en dependencia del tamaño de las regiones).
3. Si es una capa de vector convertida a ráster.
4. Use la función de “tabulate areas” (tabular zonas) para calcular el total de cada objeto en cada unidad de planificación. Una la tabla que se prepara con la capa copiada de unidades de planificación. Repita ese paso para todos los objetos.
5. Una vez que se haya repetido para todos los objetos de conservación, abra la tabla de atributo y expórtela. Ábrala en un programa de Notepad u hoja de cálculo para que sea verificada. Por ejemplo, asegúrese de que el encabezamiento incluya “pu” y note que el nombre de cada objeto tiene que ser un número. Guárdelo como archivo de texto con la extensión “.dat”.

Para usuarios de ArcMap

1. Haga una copia de la capa de unidades de planificación y elimine todos los campos excepto ID. No se quieren tabular objetos que ya se tienen. Repita los siguientes pasos para todos los objetos.
2. Use la función de “tabulate areas” (tabular zonas) para calcular el total de cada objeto en cada unidad de planificación. Una la tabla que se prepara a la capa temática copiada de unidades de planificación. Repita ese paso para todos los objetos.
3. Una vez que se haya repetido para todos los objetos de conservación, abra la tabla de atributo y expórtela. Ábrala en un programa como Notepad o una hoja de cálculo para que sea verificada. Por ejemplo, asegúrese de que el encabezamiento incluya “pu” y note que el nombre de cada objeto tiene que ser un número. Guárdelo como archivo de texto con la extensión “.dat”.
4. Remítase al manual del usuario de PANDA para ver detalles referentes al uso del **Archivo de Unidad de Planificación** con PANDA.

C-4 Archivo de Objeto de Conservación



The screenshot shows a Notepad window with a table of data. The table has 9 columns: id, type, target, spf, target2, targetocc, name, sepnum, and sepdistance. The data is as follows:

id	type	target	spf	target2	targetocc	name	sepnum	sepdistance
1	1	20	2	2	2	A	2	100000
2	1	1	2	2	2	B	2	10
3	1	1	2	3	2	C	2	10
4	1	3	2	2	2	D	2	10
5	2	1	2	0	2	E	0	10
6	2	1	2	0	2	F	0	10
7	2	3	2	0	2	G	0	10
8	2	3	2	0	2	H	0	10
9	2	2	2	0	2	I	0	10
10	2	10	2	0	2	J	0	10
11	2	8	2	0	2	K	0	10
12	3	20	2	0	2	L	0	10
13	3	15	2	0	2	M	0	10
14	3	31	2	0	2	N	0	10
15	3	27	2	0	2	O	0	10
16	3	29	2	0	2	P	0	10
17	0	59	2	0	2	Q	0	10
18	0	65	2	0	2	R	0	10
19	0	97	2	0	2	S	0	10
20	0	147	2	0	2	T	0	10
21	0	106	2	0	2	U	0	10
22	0	175	2	0	2	V	0	10
23	0	170	2	0	2	W	0	10
24	0	125	2	0	2	X	0	10

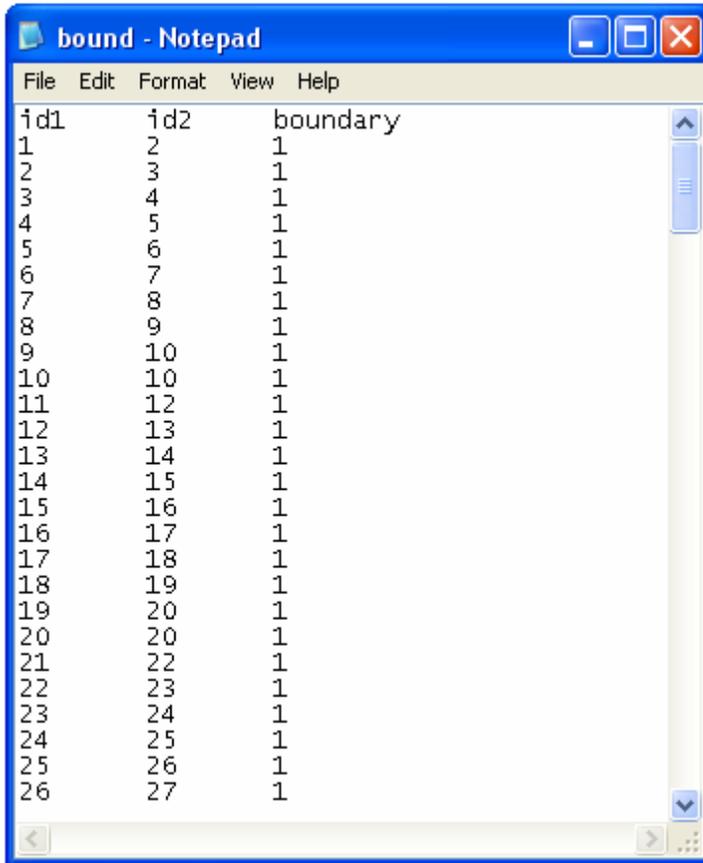
Un ejemplo del **Archivo de Objeto de Conservación** (spec.dat) utilizado en Marxan.

El **Archivo de Objeto de Conservación** contiene información referente a cada objeto que se toma en cuenta en el análisis. El grueso de la información requerida para la creación de este archivo incluirá la historia natural de cada objeto, los procesos que representan amenazas y los requerimientos espaciales para su persistencia. Posiblemente, requiera de una investigación extensa de literatura y de gran suministro de información. Es muy probable que el archivo se desarrolle en un programa de hoja de cálculo para permitir que se apliquen las fórmulas y se salven como un archivo de texto de extensión ".dat".

Si se van a establecer metas cuantitativas similares para objetos determinados, recomendamos el uso de un **Archivo de Definición de Bloques** (Vea Sección 3.3.2).

Para los usuarios de CLUZ hay múltiples funciones que crean una nueva tabla de metas cuantitativas y permiten la entrada y exportación de esa información en el formato correcto de Marxan. Para los usuarios de P.A.N.D.A., remítanse al manual de usuario de PANDA

C-5 Crear el Archivo de Longitud de Frontera



id1	id2	boundary
1	2	1
2	3	1
3	4	1
4	5	1
5	6	1
6	7	1
7	8	1
8	9	1
9	10	1
10	10	1
11	12	1
12	13	1
13	14	1
14	15	1
15	16	1
16	17	1
17	18	1
18	19	1
19	20	1
20	20	1
21	22	1
22	23	1
23	24	1
24	25	1
25	26	1
26	27	1

Un ejemplo del **Archivo de Longitud de Frontera** (bound.dat)

El **Archivo de Longitud de Frontera de Marxan** contiene los costos de las fronteras entre las unidades de planificación. Las dos extensiones siguientes generan un **Archivo de Longitud de Frontera** donde el costo es la longitud de la frontera compartida entre las unidades de planificación adyacentes (la opción más común), sin embargo, para otras técnicas remitimos a los usuarios más avanzados al MGPH. Note que la capa temática de unidad de planificación debe ser correcta desde el punto de vista topológico. Es decir, no deben existir vacíos ni coincidencias entre las unidades de planificación vecinas, ni curvas pendientes. Las herramientas de topología dentro de ArcGIS u otros software pueden utilizarse para examinar y corregir esos errores. Con frecuencia, es posible crear un **Archivo de Longitud de Frontera** a partir de una capa temática de unidad de planificación con errores topológicos. Sin embargo, la información de frontera de una capa de unidad de planificación, va a incluir esos errores, lo que provocará que Marxan genere resultados extraños (ej. hacer que las unidades de planificación vecinas no aparezcan como adyacentes).

Para los usuarios de ArcView 3 hay una extensión disponible desde <http://www.mosaic-conservation.org/gis/boundary.html>. Esa extensión se incluye también en la extensión CLUZ.

Para ArcGIS, las extensiones JNCC ArcGIS están disponibles desde <http://www.ecology.uq.edu.au/index.html?page=30009&pid=29778>

PANDA también brinda una herramienta para crear el **Archivo de Longitud de Frontera** con el menú “Marxan Advanced” (Marxan avanzado).

C-6 Conectar Archivos de Salida con ArcGIS

Es muy útil mostrar los archivos de salida de Marxan para la interpretación visual de los resultados, la publicación y el apoyo a la toma de decisiones. Existen interfases fáciles de utilizar, que permiten el análisis y la muestra instantánea de los resultados en el SIG. Entre ellas se incluyen CLUZ, P.A.N.D.A. y C-Plan. Si no se utilizan ninguna de esas interfases, puede conectar los archivos de salida, de manera manual, a ArcView 3 y ArcGIS 8 y 9 de la siguiente manera:

1. Abra el archivo en un programa de hoja de cálculo, auto-ajuste las columnas y salve como .dbf or .archivo de texto.
2. Abra en SIG y una la tabla con las capas espaciales de las unidades de planificación.
3. Resulta de utilidad abrir copias del archivo de unidad de planificación de manera que se puedan ver y comparar los resultados.

(Esta página está en blanco intencionalmente)