



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA



ecotrust



Natural Heritage Trust
Helping Communities Help Australia
An Australian Government Initiative

αεδα

Applied Environmental Decision Analysis Commonwealth
Environmental Research Facility

MARXAN CON ZONAS (V1.0.1)

Conservación por zonas mediante calibración espacial explícita

Manual preparado para Ecotrust, la Organización de Patrimonio Nacional y el Centro de Análisis Aplicado para la toma de decisiones Medioambientales

Matthew Watts
Carissa Klein
Romola Stewart
Ian Ball
Hugh Possingham

Noviembre 2008

Traducido al Español por: Liliana Solano
Revisado por: Maria Jose Martinez-Harms
Marzo 2013

Citación sugerida: Watts, M. E., C. K. Klein, R. R. Stewart, I. R. Ball, and H. P. Possingham. 2008. Marxan with Zones (V1.0.1): Conservation Zoning using Spatially Explicit Annealing, a Manual.



Introducción

Marxan con Zonas (también llamado Marxan Z) es una herramienta de apoyo para la toma de decisiones en conservación, desarrollado por el Centro de Ecología de la Universidad de Queensland. Sin embargo, el desarrollo del programa Marxan Z ha sido posible gracias al apoyo de un número considerable de instituciones. Queremos agradecer el apoyo económico de Ecotrust, cuyo interés en el desarrollo de este programa surgió de la necesidad de apoyar el diseño de áreas protegidas marinas a lo largo de la costa de California, como parte de la Ley de Protección de la diversidad marina de California. Así mismo, agradecemos el financiamiento recibido de la Organización de Patrimonio Nacional y el Centro de Análisis Aplicado para la toma de decisiones Medioambientales.

Marxan Z es una extensión del programa Marxan, desarrollado por Ian Ball y Hugh Possingham. Marxan Z tiene la misma funcionalidad de Marxan, pero incluyendo la capacidad de asignar unidades de planificación a múltiples zonas (por ejemplo: áreas marinas protegidas bajo varios niveles de protección) e incorporar múltiples costos en un esquema de planificación sistemática. El propósito de Marxan Z es el de asignar a cada unidad de planificación en un área de estudio una zona particular, con el fin de alcanzar un número dado de objetivos ecológicos, sociales y económicos, a un mínimo costo total.

Marxan es una herramienta que ayuda en el diseño de redes de áreas protegidas o sistemas de reserva. Marxan Z abarca el rango de problemas que el programa puede resolver para conseguir la asignación de recursos más cercana al óptimo para la configuración de zonas-múltiples.

Este manual está dividido en dos secciones: 1) Uso de Marxan Z: Archivo de entrada de datos y archivo de resultados, y 2) Función objetiva y algoritmos usados en Marxan Z. La primera sección describe la información básica necesaria para entender cómo funciona y como se usa el programa Marxan. La segunda sección del manual provee información adicional relacionada con las características técnicas de Marxan Z. Se incluye una lista de publicaciones en las que se ha usado o discutido Marxan. Estas publicaciones pueden ser muy útiles en el proceso de aplicar Marxan Z.

Este manual se diseñó usando como guía el manual del programa Marxan (v1.8.2), pero ha sido adaptado para incluir la funcionalidad adicional de Marxan Z y excluir las características de Marxan que no son utilizadas cuando se trabaja con Marxan Z.

Tabla de contenido

<u>1.0 Usando Marxan Z: Archivos de entrada de datos y archivos de resultados</u>	3
<u>1.1 ARCHIVOS DE ENTRADA DE DATOS</u>	3
<u>Archivo de unidad de planificación</u>	3
<u>Archivo de características</u>	4
<u>Unidad de planificación versus características</u>	5
<u>Zonas</u>	6
<u>Costos</u>	6
<u>Costos de zona</u>	7
<u>Longitud de frontera</u>	7
<u>Costo del límite de la zona</u>	8
<u>Planear la unidad de zona</u>	9
<u>Planear el límite de la unidad</u>	9
<u>Zona objetivo y zona objetivo 2</u>	10
<u>Contribución de la zona y contribución de la zona 2</u>	12
<u>Archivo de entrada de parámetros</u>	13
<u>1.2 ARCHIVOS DE RESULTADOS</u>	18
<u>Soluciones para cada repetición del análisis</u>	18
<u>Mejor solución para todas las repeticiones del análisis</u>	18
<u>Valores faltantes en cada repetición del análisis</u>	18
<u>Información de resumen</u>	19
<u>Detalles del escenario</u>	20
<u>Solución sumada</u>	20
<u>Archivo de memoria de pantalla</u>	20
<u>1.3 PANTALLA DE RESULTADOS</u>	22
<u>2.0 Función objetiva y Algoritmo usado</u>	24
<u>2.1 LA FUNCION OBJETIVO</u>	24
<u>Costo de configuración de la Zona</u>	25
<u>Longitud de la frontera y Fragmentación</u>	25
<u>Requerimientos de representación de características</u>	25
<u>Penalización de las características</u>	25
<u>Factor de penalización de las características</u>	28
<u>Penalidad del valor de umbral de costo</u>	28
<u>2.2 METODOS DE ALGORITMOS</u>	29
<u>Calibración simulada</u>	29
<u>Mejoramiento iterativo</u>	31
<u>3.0 AGRADECIMIENTOS</u>	33
<u>4.0 REFERENCIAS BIBLIOGRAFICAS CLAVES</u>	34
<u>Apéndice A. Ejemplo del archivo de entrada de parámetros</u>	38

NOTA: Para facilitar la identificación de los términos y parámetros predeterminados en el programa Marxan Z, y su relación con cada explicación ofrecida en este manual en español, todas las siglas se han dejado intencionalmente en inglés.

1.0 Usando Marxan Z: Archivos de entrada de datos y archivos de resultados

1.1 ARCHIVOS DE ENTRADA DE DATOS

Marxan Z usa siete archivos fundamentales de entrada de datos. También hay archivos opcionales de entrada de datos que facilitan funcionalidades adicionales en Marxan Z.

Archivos fundamentales de entrada de datos	Archivos opcionales de entrada de datos
Unidad de planificación	Longitud de frontera
Características	Costo de la zona de frontera
Unidad de planificación versus características	Planificación de la unidad de zona
Zonas	Planificación del límite de la unidad
Costos	Zona objetivo o zona objetivo 2
Costos de la zona	Contribución de la zona o Contribución de la zona 2
Parámetros de entrada	

Esta sección describe la estructura básica de cada archivo. A excepción del archivo de entrada de parámetros, todos los archivos constan de una línea de título y el contenido principal. La línea de título es una lista de nombres específicos que describen el contenido de cada columna del contenido principal. Cada archivo contiene una serie de nombres de columna obligatorios y otros opcionales para que Marxan Z funcione. Los títulos en cada archivo deben estar escritos solo en minúsculas, sin signos de puntuación, sin espacios o numerales (excepto cuando se indique lo contrario). Todos los archivos de entrada se pueden crear en cualquier programa de hojas de cálculo o edición de texto. Las variables que se encuentren en la misma línea, se pueden separar usando caracteres como espacios, tabulaciones, o comas. Un ejemplo de este formato se presenta en la sección que describe el archivo de unidad de planificación.

Archivo de unidad de planificación.

Nombre predeterminado: “pu.dat”

Descripción: Este archivo se requiere para que Marxan Z funcione. Contiene una lista de todas las unidades de planificación en la región de estudio, e incluye los costos asociados con la asignación de una unidad de planificación a una zona en particular. Costos

múltiples se pueden indicar para cada unidad de planificación. La importancia de cada costo en referencia con los demás costos se puede especificar en el archivo de unidad de planificación o en el archivo de costo de la zona (que se describirá más adelante en este documento). Si el archivo de costo de zona se usa para asignar el grado de importancia de cada costo, los valores de cada costo deberían ser normalizados a la misma escala a la que se encuentran los datos del archivo de unidad de planificación.

Titulo	Requerido/Opcional	Descripción
id	requerido	Identificador numérico para la unidad de planificación.
costname	Opcional, usara un valor predeterminado de 1 si no se especifica algo diferente.	Costo individual de cada unidad de planificación. Campos de costos múltiples con nombres diferentes se pueden incluir en el archivo de unidad de planificación de Marxan Z. El título 'costname' se puede reemplazar con el nombre del costo pero no debe incluir caracteres delimitadores (espacios, tabuladores, etc.).

Ejemplo de una parte del archivo pu.dat

```
id, salmonfishing, squidfishing
1, 34, 22
2, 4, 0
3, 40, 12
```

Archivo de características

Nombre predeterminado: “feat.dat”

Descripción: Este archivo es necesario para el funcionamiento de Marxan Z. Cada característica se debe listar en este archivo y se le debe asignar a cada una un número único de identificación. El rango de características espaciales aceptadas incluye características ecológicas, económicas y sociales. En este archivo se debe especificar la extensión del área objetivo de cada característica (o la extensión objetivo proporcional) a ser capturada por las zonas nominadas. Si se especifica un objetivo general en este archivo, usando las opciones de target, prop, targetocc o propocc, esto requerirá la inclusión de un archivo de contribución de zona y/o un archivo de contribución 2, en el cual se especifique cuales zonas contribuyen a estos objetivos, y el índice al que las zonas contribuyen hacia los objetivos. El objetivo general especificado en el archivo de características y los índices de contribución especificados en el archivo de contribución de zona *trabajan en tandem* para especificar objetivos generales para las características e

índices de contribución diferencial de las zonas a estos objetivos. Ver una descripción de estos archivos en la sección subsecuente que trata el tema de contribución de zonas.

Titulo	Requerido/Opcional	Descripción
id	Requerido	El identificador numérico para esta característica. Debe ser un número entero positivo.
target	Opcional cuando se use la opción prop	El tamaño del objetivo (en unidades del archivo puvfeat.dat) de la característica que se va a incluir a lo largo de todas las zonas protegidas (por ejemplo: objetivo general).
prop	Opcional cuando se use la opción target	Es una alternativa al objetivo. Es la proporción de la cantidad total de la característica que tiene que ser incluida en las zonas protegidas. Un valor de 0.3 indicaría que el 30% de la característica debe ser protegido.
targetocc	opcional	El número de ocurrencias requerido de la característica. Se considera una ocurrencia cuando la característica ocurre en una unidad de planificación, independientemente de su valor. Esto se puede usar en conjunto o en lugar de 'target'.
propocc	opcional	Porcentaje requerido de ocurrencia de la característica. Se puede usar en conjunto o en lugar de 'prop'.
fpf	opcional	Factor de penalización para la característica.
name	opcional	Indica el nombre de la característica. No debe incluir espacios, o caracteres no-alfanuméricos en el nombre.

fpf

El factor de penalización de la característica es un factor multiplicativo que puede ser único para cada característica. Es una medida de cuán importante es alcanzar el objetivo para la determinada característica. Si este es menor de 1, el algoritmo probablemente se resista a incluir una unidad de planificación, para incluir la característica particular sino hay otras características en esa unidad de planificación. Si la solución no alcanza a cumplir el objetivo para una característica, el fpf se debería incrementar a un valor superior a 1.

Ejemplo de una parte del archivo feat.dat

id, prop, fpf, name 1, 0.3, 1, RockyReef 2, 0.9, 1, KelpForest
--

3, 0.5, 1, FishingGrounds

Unidad de planificación versus característica

Nombre predeterminado: “puvfeat.dat”

Descripción: Este archivo es requerido para el funcionamiento de Marxan Z. Contiene la información sobre la distribución de las características a lo largo de las unidades de planificación. El rango de características que pueden ser aceptadas incluye ecológicas, económicas e información social. El archivo debe contener tres columnas que deben estar en el siguiente orden: - featureid, puid, amount - organizadas siguiendo el orden de los id de cada unidad de planificación, para que Marxan Z pueda ejecutarlo correctamente.

Titulo	Requerido/Opcional	Descripción
featureid	requerido	El identificador de la característica se debe especificar en el archivo de características.
puid	requerido	El identificador de la unidad de planificación se debe indicar en el archivo de unidad de planificación.
amount	requerido	Cantidad de la característica en la unidad de planificación. La unidad de medida puede variar entre características. Sin embargo, las unidades dentro de una característica debe ser consistente

Ejemplo de una parte del archivo puvfeat.dat

```
featureid, puid, amount
1, 1, 350
1, 2, 50
1, 3, 100
2, 3, 160
3, 1, 300
3, 2, 200
```

Zonas

Nombre predeterminado: “zones.dat”

Descripción: Este archivo se requiere para el correcto funcionamiento de Marxan Z. Contiene una lista de los nombres e identificadores numéricos de todas las posibles zonas. El archivo debe tener dos espacios en el siguiente orden –zoneid, zonename - organizados de menor a mayor en el campo de identificador de zona: zoneid.

Titulo	Requerido/Opcional	Descripción
zoneid	requerido	El identificador numérico de la zona. El zoneid



		debe ser un número entero positivo y el archivo debe estar organizado de menor a mayor siguiendo el campo de zoneid.
zonename	requerido	Indica el nombre de la zona. El nombre solo debe contener caracteres alfanuméricos y no se debe incluir espacios.

Ejemplo de una parte del archivo zones.dat

zoneid, zonename 1, Available 2, MarineReserve 3, MarinePark 4, FishingZone

Costos

Nombre predeterminado: “costs.dat”

Descripción: El propósito de este archivo es asignar un identificador único a cada uno de los nombres de costo contenidos en el archivo de unidad de planificación. Si este archivo no es incluido en Marxan Z, los valores de costo indicados en el archivo de unidad de planificación se consideraran inválidos y un valor predeterminado de 1 será asignado a todas las unidades de planificación. Este archivo contiene una lista de los nombres e identificadores para todos los posibles costos. Los valores de costo son incluidos en el archivo de unidades de planificación. Este archivo debe tener dos campos organizados en el siguiente orden: costid, costname - organizados de menor a mayor siguiendo el campo de costid.

Título	Requerido/Opcional	Descripción
costid	requerido	El identificador numérico para el costo. El costid debe ser un número entero positivo y el archivo debe ser organizado de menor a mayor siguiendo el campo de costid.
costname	requerido	Indica el nombre del costo. El nombre no debe incluir espacios y solo debe contener caracteres alfanuméricos.

Ejemplo de una parte del archivo costs.dat

costid, costname 1, area 2, salmonfishing 3, squidfishing
--

Costo de Zona

Nombre predeterminado: “zonecost.dat”

Descripción: Este archivo se requiere para el funcionamiento de Marxan Z. Incluye el factor de ponderación asignado a cada costo en cada zona. En el caso que algunas combinaciones de zona y costo no se presenten, al costo se le asignara un factor de ponderación de 0 y no se incluirá al formular una solución. El archivo debe contener tres campos en el siguiente orden: – zoneid, costid, multiplier - organizados por costid de menor a mayor.

Título	Requerido/Opcional	Descripción
zoneid	requerido	El identificador de zona - debe ser compatible con el archivo zones.dat
costid	requerido	El identificador de costo - debe ser compatible con el archivo costs.dat
multiplier	requerido	Este número puede ser un número entero o una fracción. En una zona dada, será multiplicado por el costo específico. Todos los costos en una zona determinada serán multiplicados por el multiplicador especificado y luego se adicionara para dar el costo total para cada unidad de planificación. Por ejemplo, si hay 3 costos en una zona, el costo total para esa zona seria calculado usando la siguiente ecuación: Total C = (C1 * M1) + (C2 * M2) + (C3*M3) Donde C = costo, M = Multiplicador

Ejemplo de una parte del archivo zonecost.dat

```
zoneid, costid, multiplier
2, 1, 0.2
2, 2, 0.4
2, 3, 0.4
```

Longitud de límite (frontera)

Nombre predeterminado: “bound.dat”

Descripción: Este archivo es opcional en Marxan Z. Se requiere cuando el modificador de la longitud de frontera (BLM) es mayor que 0 (cero) en el archivo input.dat. Este archivo contiene información relacionada con el "costo" de las fronteras de dos unidades de planificación. Las fronteras que estén duplicadas en este archivo, serán sumadas una con otra. Cuando una unidad de planificación se encuentra en el borde de una región de



estudio, se le identifica como frontera consigo misma (por ejemplo, el identificador de la unidad de planificación es igual para id1 e id2). El archivo debe tener tres campos en el siguiente orden id1, id2, boundary.

Titulo	Requerido/Opcional	Descripción
id1	requerido	Identificador de la unidad de planificación.
id2	requerido	Identificador de la unidad de planificación vecina a id1 o el mismo identificador de id1 en el caso de fronteras irremovibles.
boundary	requerido	Debería ser mayor o igual a cero. Puede ser la misma longitud de la frontera o puede ser modificado de tal manera que refleje el "costo" de separar dos unidades de planificación.

Ejemplo de una parte del archivo bound.dat

```
id1, id2, boundary
1, 1, 1
1, 2, 1
1, 3, 1
2, 3, 1
```

Costo de la frontera de la Zona

Nombre predeterminado: “zoneboundcost.dat”

Descripción: Este es un archivo opcional en Marxan Z. Este archivo se usa cuando se quiere dar diferentes costos a las fronteras entre diferentes zonas (por ejemplo, si usted quiere que dos zonas estén ubicadas una junto a la otra, usted puede incrementar el costo de frontera entre esas zonas). Si este archivo no existe, todas las fronteras entre zonas recibirán un costo de 1. Así mismo, si el costo de una frontera zona-zona no se indica en este archivo, se le asignará el valor de 1. Este archivo es similar al archivo bound.dat pero se relaciona con el "costo" de frontera de las zonas y no con las unidades de planificación. El archivo debe tener tres campos en el siguiente orden – zoneid1, zoneid2, cost - organizados de menor a mayor, primero por zoneid1, y luego por zoneid2.

Titulo	Requerido/Opcional	Descripción
zoneid1	requerido	Identificador de la Zona - debe ser indicado en el archivo zones.dat
zoneid2	requerido	Identificador de la Zona, diferente de zoneid1 - debe ser indicado en el archivo zones.dat
cost	requerido	El costo entre zoneid1 y zoneid2. Si no se indica el costo, se le asignará un costo de 1.



Ejemplo de una parte del archivo zoneboundcost.dat

```
zoneid1,zoneid2,cost
1,1,0
2,2,0
3,3,0
1,2,0.1
1,3,0.1
2,3,1
```

Unidad de Planificación de Zona

Nombre predeterminado: "puzone.dat"

Descripción: Este archivo es opcional en Marxan Z. Este archivo se usa cuando se quiere restringir ciertas unidades de planificación a dos o más de sus zonas. Este archivo debe tener dos campos en el siguiente orden– puid, zoneid - organizados de menor a mayor por puid, y luego por zoneid.

No use este archivo para restringir unidades de planificación a una sola zona; por el contrario, para este propósito use el archivo de "Restricción de unidad de Planificación". En este archivo No ingrese información relacionada con identificadores de unidades de planificación que no estén en el archivo "Unidad de Planificación".

Título	Requerido/Opcional	Descripción
puid	requerido	Identificador de unidad de planificación que es restringida a una zona específica. La misma unidad de planificación se puede incluir más de una vez para indicar la restricción a más de una zona.
zoneid	requerido	Identificador de la zona a la cual la unidad de planificación puid está restringida.

Ejemplo de una parte del archivo puzone.dat

```
puid, zoneid
18, 1
19, 2
19, 3
20, 1
```

Límite de la unidad de planificación

Nombre predeterminado: "pulock.dat"

Descripción: Este archivo es opcional en Marxan Z. Este archivo se usa cuando se quiere restringir ciertas unidades de planificación a una sola zona. Este archivo se debe componer de dos campos en el siguiente orden: puid, zoneid - organizado de menor a mayor por puid, luego por zoneid. No incluya aquí datos de identificadores de unidades de planificación que no estén en el archivo "Unidad de Planificación".

Título	Requerido/Opcional	Descripción
puid	Requerido	Identificador de la unidad de planificación que es restringida a una zona específica. La misma unidad de planificación puede ser incluida más de una vez para indicar la restricción de esa unidad a más de una zona.
zoneid	Requerido	Identificador de la Zona a la que está restringida la unidad de planificación identificada en puid.

Ejemplo de una parte del archivo pulock.dat

```
puid, zoneid
57,2
71,2
72,2
86,2
```

Zona objetivo y Zona Objetivo 2

Nombre Predeterminado: "zonetarget.dat" o "zonetarget2.dat"

Descripción: Estos archivos son opcionales en Marxan Z. La zona objetivo y la zona objetivo 2 no se requieren para el funcionamiento de Marxan Z. La Zona Objetivo y la Zona Objetivo 2 se usan para determinar objetivos en zonas en Marxan Z. Estos archivos se pueden usar al mismo tiempo con objetivos generales (target o prop) establecidos en el archivo de características, o pueden ser usados en lugar de los objetivos generales.

Zona objetivo: El archivo de la zona objetivo permite al usuario determinar un objetivo para cada característica en cada zona (por ejemplo, objetivos específicos para la zona). No es necesario indicar objetivos para cada característica que sean específicos a cada zona. El archivo de zona objetivo y el de contribución de zona *no se relacionan* entre sí. La zona disponible puede tener también objetivos específicos de la zona; un objetivo específico a una zona, asegurara que en esta zona el número de objetivos de las

características nombradas sea protegido. Los archivos deben tener los campos correctos en el orden correcto como se especifica en las tablas siguientes.

Ejemplo de cómo trabaja el archivo zonetarget.dat

En el archivo zonetarget.dat, la característica Y tiene un objetivo de 25 Km² en la zona 2. En el archivo feat.dat, la característica Y tiene un objetivo general de 50 Km². En la región de estudio la característica Y ocurre en 500 Km². Esto significa que queremos proteger en general un 10% de la característica Y, con 5% de la característica protegido en zona 2, y el restante 5% de la característica en cualquier otra zona protegida.

Campos requeridos en el archivo zonetarget.dat

Titulo	Requerido/Opcional	Descripción
zoneid	requerido	El identificador de zona
featureid	requerido	El identificador de característica
target	requerido	El valor del objetivo (con las mismas unidades usadas en el archivo puvfeat.dat), porcentaje, ocurrencia, o porcentaje de ocurrencia que deseamos incluir de la característica en una zona específica (por ejemplo, objetivo específico para una zona). No se permiten valores negativos para los objetivos.
Target type	opcional	Indica el tipo de objetivo especificado en la columna de objetivo (por ejemplo, cantidad, porcentaje, ocurrencia, porcentaje de ocurrencia). Si esta columna no se incluye, Marxan Z usará un valor predeterminado de 0

Campos requeridos en el archivo zonetarget2.da

Titulo	Requerido/Opcional	Descripción
zoneid	requerido	El identificador de zona
target	requerido	La cantidad del objetivo (en unidades del archivo puvfeat.dat), porcentaje, ocurrencia o porcentaje de ocurrencia de la característica que se quiere incluir en la zona especificada (por ejemplo, objetivo específico de zona). No se permiten valores negativos para los objetivos.
targettype	opcional	Tipo de objetivo indica el tipo de objetivo especificado en la columna de objetivo (por ejemplo, cantidad, ocurrencia, porcentaje de ocurrencia). Si esta columna no se incluye,



Titulo	Requerido/Opcional	Descripción
		Marxan Z usara un valor predeterminado de 0 (ver a continuación las definiciones del campo targettype)

Valores de tipo de objetivos

0 = Cantidad de objetivo (en unidades del archivo puvspr2.dat) de la característica. Similar al objetivo en el archivo feat.dat.

1 = Porcentaje del objetivo como proporción de la cantidad total de la característica. Similar al campo prop en el archivo feat.dat.

2 = Objetivo de ocurrencia. Si la característica ocurre en una unidad de planificación, sin importar en que cantidad, es considerado una ocurrencia. Similar al campo targetocc en el archivo feat.dat.

3 = Porcentaje de objetivo como proporción del total de ocurrencia de una característica. Similar al campo propocc en el archivo feat.dat.

Objetivo de Zona 2:

El objetivo de Zona 2 es idéntico al objetivo de zona *excepto* que no incluye el campo de featureid, lo que significa que el objetivo especificado se aplicara a todas las características dentro de una zona especificada. Si Usted usa ambos archivos - el de objetivo de zona y el de objetivo de zona 2 - el archivo de objetivo de zona 2 se antepondrá al de objetivo de zona.

Ejemplo de una parte del archivo zonetarget.dat

zoneid, featureid, target
1, 3, 250
2, 1, 50

Contribución de Zona y Contribución de Zona 2

Nombre predeterminado: "zonecontrib.dat" o "zonecontrib2.dat"

Descripción: Estos archivos son opcionales en Marxan Z. La contribución de Zona y la Contribución de Zona 2 no se requieren para el funcionamiento de Marxan Z, aunque uno u otro de estos archivos *debe ser usado* si un objetivo general es especificado en el archivo de características. Estos archivos *trabajan en tandem* con el objetivo general especificado en el archivo de características para especificar tasas de contribución diferencial al objetivo general. Los archivos de contribución de Zona *no se relacionan* con los archivos de objetivo de zona. Los archivos deben tener los campos correctos en el orden correcto especificado en las tablas siguientes.

Contribución de Zona: El archivo de objetivo de zona permite al usuario especificar una tasa de contribución para cada característica en cada zona al objetivo general especificado en el archivo de objetivo de zona. Los dos archivos trabajan en tándem uno con el otro.

El archivo "zonecontrib.dat" permite al usuario especificar una fracción de contribución para cada característica en cada zona.

Los campos requeridos para el archivo zonecontrib.dat

Titulo	Requerido/Opcional	Descripción
Zoneid	Requerido	El identificador de zona
Featureid	Requerido	El identificador de característica
Fraction	Requerido	La fracción de contribución para esta característica como se aplicara al objetivo general especificado en el archivo de características. No se permiten Contribuciones negativas.

El archivo "zonecontrib2.dat" permite al usuario especificar una fracción de contribución para cada zona. Este archivo omite el campo de featureid dado que todas las características en una zona tienen la misma fracción de contribución.

Los campos requeridos en el archivo zonecontrib2.dat

Titulo	Requerido/Opcional	Descripción
zoneid	Requerido	El identificador de zona
fraction	Requerido	La fracción de contribución para todas las características en esta zona como se aplicaran al objetivo general especificado en el archivo de características. No se permiten Contribuciones negativas.

Ejemplo de cómo trabaja en archivo zonecontrib.dat

En el archivo zonecontrib.dat, la característica Y tiene una contribución de 1 en la zona 2. En el archivo feat.dat, la característica Y tiene un objetivo general de 50 km². Una unidad de planificación contiene 25 km² de la característica Y, y se le ubica en la zona 2. Tendrá una contribución de 25 km² al objetivo para la característica Y (25 * 1), y requeriremos 25 km² adicionales de la característica Y para alcanzar el objetivo.

Si la característica Y tiene una contribución de 0.2 en la zona 2, entonces la contribución de asignar la misma unidad de planificación en la zona 2 es (25 * 0.2 = 5) 5 km², y requeriremos 45 km² adicionales de la característica Y para alcanzar el objetivo.

Ejemplo de una porción del archivo zonecontrib2.dat

```
zoneid, fraction
1, 0
2, 0.2
3, 1
```

Archivo de entrada de parámetros

Este archivo es requerido para el funcionamiento Marxan Z. Contiene todas las definiciones de los parámetros principales que controlan la forma en que trabaja Marxan Z. Una manera conveniente de construir y editar este archivo es usando el programa Inedit.exe. Sin embargo, el archivo se puede crear y editar usando un editor de texto.

Hay seis secciones en este archivo: Parámetros Generales, Parámetros de Calibración, umbral de costo, Entrada, Guardar, y Control de Programa. Las variables, sus valores predeterminados, y una breve descripción de las variables se enumeran en esta sección. Una descripción más detallada de las variables más complejas se describe al final de cada sección y en ocasiones en la sección 2.0 de este manual. En el archivo de entrada de parámetros, cualquier línea que no empieza con uno de los nombres validos de variables es ignorada, y por tanto puede ser usada como línea de comando. Note que todos los nombres de las variables son escritas en mayúsculas y sin espacios. Para asignar los valores de una variable, empiece la línea con el nombre de la variable y continúe con el valor que le quiere dar a dicha variable. Las variables se pueden presentar en cualquier orden en el archivo pero se producirá un error si cualquiera de las variables se define dos veces. La mayoría de las variables tienen valores predeterminados que serán usados en caso de no contar con valores definidos por el usuario. Un ejemplo del archivo se encuentra en el apéndice A.

Parámetros Generales

Variable	Valor Predeterminado	Breve Descripción
BLM	0	Modificador de la longitud de frontera
PROP	0.5	Proporción de unidades de planificación para empezar el análisis. Debe ser un número entre 0 y 1. Cuando se usa calibración simulada con mejora iterativa (recomendado), cambiar el valor si no tiene impacto medible en la solución final.
RANDSEED	-1	El número de semilla al azar debe ser un número entero. Si el



		valor es negativo, el programa seleccionará al azar la semilla. Un valor positivo solo se usaría con propósitos de depuración, en caso que usted quiera que más de una aplicación del programa sean idénticas.
NUMREPS	100	Numero de análisis independientes que tendrán las mismas condiciones de iniciación.
AVAILABLE ZONE	1	La zona disponible es tratada como una zona no protegida en Marxan Z. Aquí es donde usted indica el número identificador de la zona que está disponible en el archivo de entrada de zonas.

Modificador de la longitud de Frontera (BLM)

El BLM es una variable que controla la importancia de minimizar la longitud de frontera en relación con el costo de unidad de planificación. En otras palabras, el BLM es un parámetro que dirige la asignación de unidades de planificación a zonas en una formación de conglomerada, en lugar de seleccionar varias unidades de planificación desconectadas. Con un BLM pequeño, el algoritmo se concentra en minimizar el costo de unidad de planificación, mientras que un BLM grande, dará un énfasis mayor a minimizar la longitud de la frontera. Un BLM bajo, permitirá al programa seleccionar varias áreas pequeñas, mientras altos BLM forzarán la selección de un menor número de áreas, de mayor tamaño. Si el BLM es cero, la longitud de la frontera no tendrá ningún impacto en la solución final. No hay un número universal que se considere mejor, ya que está sujeto al costo y geometría de la región de estudio o unidades de planificación de cada proyecto de planificación. El usuario debe explorar los efectos de diferentes valores de BLM para determinar un valor de BLM eficiente. Un método para determinar un nivel eficiente de compactividad espacial, ha sido descrito por Stewart y Possingham (2005). Este método considera el efecto de incrementar el BLM y minimizar la longitud de frontera de una solución de Marxan.

Parámetros de Calibración

Variable	Valor predeterminado	Breve Descripción
N	1 million	Numero de iteraciones para la calibración de cada corrida (por ejemplo, el número de veces que Marxan Z trata de generar una solución por cada corrida).
STARTTEMP	1	Temperatura inicial para la calibración. Un valor de -1 indica que se usara calibración flexible y el programa automáticamente seleccionara una temperatura inicial. Se recomienda usar esto en Marxan Z.
COOLFAC	0	Esto determina que tan rápido el sistema se enfría. Si

Variable	Valor predeterminado	Breve Descripción
		se establece una STARTTEMP de -1, esta variable no es requerida para que Marxan Z Funcione. Pero se recomienda usarlo en Marxan Z.
NUMTEMP	10,000	Numero de decrecimiento de la temperatura para la calibración. Debe ser menor o igual al número de iteraciones. Un valor de 10,000 es ideal y se recomienda en Marxan Z. Usar un valor menor puede hacer el régimen muy tosco; usar un valor muy alto desencadenara problemas de errores de aproximacion del valor de la temperatura.

Numero de iteraciones

A mayor número de iteraciones usadas, mayor será el tiempo que el programa necesite para correr el análisis, y mayor será la posibilidad de que Marxan Z genere una mejor solución (por ejemplo, produciendo un valor menor de función objetivo). El número ideal de iteraciones varía de un proyecto a otro. Si se prueba con varias iteraciones y se genera una gráfica de número de iteraciones vs el valor de función objetivo, es posible determinar el punto en el que el incremento en el número de iteraciones deja de producir valores significativamente mejores de la función objetivo.

Umbral de costo

Si no se usan umbrales de costo, esta sección no debe ser incluida en los archivos de entrada de Marxan Z. Una Descripción detallada de los umbrales de costo se presenta en la sección 2.0 de este manual.

Variable	Valor predeterminado	Breve Descripción
COSTTHRESH	0	El límite de costo permite al usuario establecer a un costo determinado el corte en la configuración de zonificación. Indicar un límite de costo de cero deshabilita esta opción.
THRESHPEN1	0	Parámetro de penalización de limite 1
THRESHPEN2	0	Parámetro de penalización de limite 2

Entrada

Todos los posibles archivos de entrada y su nombre predeterminado se presentan en esta sección. Sin embargo, en la sección de entrada solo necesita incluir los archivos requeridos por Marxan Z y los archivos adicionales que usted elija utilizar, de acuerdo con lo descrito en este manual.

Variable	Nombre predeterminado	Breve Descripción
INPUTDIR	Definido por el usuario	Directorio de entrada donde los archivos de datos se almacenan en el ordenador del usuario.
PUNAME	pu.dat	Nombre del archivo de la unidad de planificación
FEATNAME	feat.dat	Nombre del archivo de atributos características
PUVFEATNAME	puvfeat.dat	Nombre del archivo de unidad de planificación contra el archivo de características
ZONESNAME	zones.dat	Nombre del archivo de zonas
COSTSNAME	costs.dat	Nombre del archivo de costo
ZONECOSTNAME	zonecost.dat	Nombre del archivo de costo de zona
BOUNDNAME	bound.dat	Nombre del archivo de longitud de frontera
ZONEBOUNDCOSTNAME	zoneboundcost.dat	Nombre del archivo de costo de frontera de la zona
PUZONENAME	puzone.dat	Nombre del archivo de la zona de unidad de planificación
PULOCKNAME	pulock.dat	Nombre del archivo de candado de la unidad de planificación
ZONETARGETNAME	zonetarget.dat	Nombre del archivo del objetivo de zona
ZONETARGET2NAME	zonetarget2.dat	Nombre del archivo del objetivo de zona 2
ZONECONTRIBNAME	zonecontrib.dat	Nombre del archivo de contribución de zona
ZONECONTRIB2NAME	zonecontrib2.dat	Nombre del archivo de contribución de zona 2

Guardar

Para generar cualquiera de los archivos mencionados anteriormente (es decir, guardar los archivos) en Marxan Z, se debe especificar un valor de 2 o 3 (un valor de 2 es el formato de ArcView, 3 es el formato de archivo delimitado por comas ASCII) después del nombre del archivo. Si se especifica un valor de cero (0) después del nombre del archivo, no se generará ese archivo de salida. Una Descripción de cada uno de estos archivos se provee en la sección de archivos de salida.

Variable	Valor Predeterminado	Breve Descripción
SCENNAME	Definido por el Usuario	Nombre del escenario para guardar los archivos



SAVERUN	3	Guardar cada corrida
SAVEBEST	3	Guardar la mejor corrida
SAVESUMMARY	3	Guardar el resumen de la información
SAVESCEN	3	Guardar el escenario
SAVETARGMET	3	Guardar los objetivos alcanzados
SAVESUMSOLN	3	Guardar la suma de las soluciones
SAVESOLUTIONSMATRIX	3	Guardar la matriz de soluciones
SOLUTIONSMATRIXHEADERS	1	Incluir filas de títulos en la matriz de soluciones.
SAVEPENALTY	3	Guardar las penalizaciones de característica computadas
SAVELOG	3	Guardar archivo de registro
SAVEANNEALINGTRACE	3	Guardar los detalles de calibración simulada.
ANNEALINGTRACEROWS	1000	Numero de iteraciones de las que se quiere generar reportes detallados.
SAVEITIMPTRACE	3	Reporte de detalles del mejoramiento iterativo.
ITIMPTRACEROWS	1000	Numero de iteraciones de las que se quiere generar reportes detallados.
SAVEZONECONNECTIVITYSUM	3	Guardar sumatoria de conectividad de las zonas.
OUTPUTDIR	Definido por el Usuario	Directorio de salida para guardar los archivos de resultados en el ordenador de usuario.

Control de Programa

Hay dos algoritmos básicos que se pueden usar para formular una solución en Marxan Z: calibración simulada y mejoramiento iterativo. Una descripción detallada de estos algoritmos se encuentra en la sección 2.0 de este manual. Estos algoritmos se pueden usar solos o en combinación. El modo de corrido seleccionado en el archivo de entrada de parámetros determina cual algoritmo, o combinación de algoritmos, se usara para formular una solución en Marxan Z.

Título	Valor predeterminado	Breve Descripción
RUNMODE	1	Usar el valor predeterminado de 1 aplicara calibración seguida por un algoritmo de mejoramiento iterativo, dando el mejor resultado posible en Marxan Z.
MISSLEVEL	1	Esta es la proporción del objetivo que

Titulo	Valor predeterminado	Breve Descripción
		debe ser alcanzado por una característica para que no sea contada como perdida. Un valor de 1 significa que el 100% del objetivo para una característica debe ser incluido en la solución, de lo contrario se considerara una meta no alcanzada.
ITIMPTYPE	0	0 se recomienda en Marxan Z.
VERBOSITY	2	Este valor indica cuanta información se mostrara en la pantalla de Marxan Z cuando está corriendo el análisis. Un valor de 0 mostrara la menor cantidad de información y valor de 3 mostrara la mayor cantidad de información. Una Descripción más detallada de cada valor se presenta en la sección de Pantalla de Resultados.

Modo de corrido

Hay cuatro modos de corrido posibles. Cada modo determina cual algoritmo se usara. Usar el valor predeterminado de 1 aplicara calibración seguida por el algoritmo de mejoramiento iterativo, dando el mejor posible resultado en Marxan Z.

Valor	Descripción
-1	No usa ninguno de los métodos
1	Aplica calibración seguida por el algoritmo de mejoramiento iterativo
4	Usa solo el mejoramiento iterativo
6	Usa solo calibración

Modo de mejoramiento iterativo

Hay cinco modos posibles de mejoramiento iterativo. El mejoramiento iterativo solo se usara cuando se seleccione un modo de corrido que use el algoritmo de mejoramiento iterativo. Se recomienda usar el valor predeterminado de 0 en Marxan Z.

Valor	Descripción
-1	No usa mejoramiento iterativo
0	Mejoramiento normal
1	Mejoramiento iterativo en dos pasos
2	Mejoramiento iterativo de intercambio
3	Normal Seguido por dos pasos

1.2 ARCHIVOS DE RESULTADOS

En el archivo de entrada de parámetros, el usuario puede indicar cuales archivos de resultados deben ser generados por Marxan Z. El programa puede generar siete tipos de archivos de resultados. Cada archivo toma el nombre del escenario definido en el archivo de entrada de parámetros, seguido por una abreviación para cada tipo de archivo. Esta sección describirá cada uno de los archivos de resultados.

Soluciones para cada corrida

Nombre en input.dat: SAVERUN

Nombre del archivo de resultados: escenario_r001.dat, escenario_r002.dat, etc., donde 'escenario' es el nombre dado por el usuario en el parámetro SCENNAME del archivo de parámetros pinput.dat.

Descripción: Se produce un archivo por cada corrida del algoritmo describiendo cuales unidades de planificación fueron seleccionadas en cada zona. Contiene los identificadores de las unidades de planificación, seguido por el identificador de zona, separado por una coma. Cada identificador de unidad se presenta en una nueva línea.

Mejor solución para todas las corridas

Nombre en input.dat: SAVEBEST

Nombre del archivo de resultados: escenario_best, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Se produce un archivo de la mejor solución a partir de una serie de corridas en Marxan Z. Contiene el identificador de la unidad de planificación, seguido del identificador de zona, separado por una coma. Cada identificador de unidad de planificación se presenta en una nueva línea.

Valores faltantes en cada corrido

Nombre en input.dat: SAVETARGMET

Nombre del archivo de resultados: escenario_mv001.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo contiene información sobre el desempeño de la solución final de cada corrida, con respecto a alcanzar el objetivo de la característica. Este archivo solo se generara para los archivos de solución de corridas individuales que también sean generados. Se debe seleccionar ya sea con la opción para las soluciones de cada corrida o con la opción para la mejor solución. El archivo contiene los siguientes títulos de



columna: feature, feature name, target, amount held, occurrence target, occurrences held, target met. Cada columna se describe a continuación. Para cada zona especificada en el archivo zones.dat (excepto para la zona disponible), hay columnas separadas de: target, amount held, y occurrences held, asociadas con la zona.

Titulo	Descripción
Feature	El identificador de la característica, indicado en feat.dat
Feature Name	El nombre opcional para la característica, indicado en feat.dat
Target	La cantidad objetivo de la característica, indicada en feat.dat
Total Amount	Cantidad total de la característica en la región de estudio.
Contributing Amount Held	Cantidad de la característica capturada, multiplicada por la fracción de contribución de la característica para la zona en la que esta característica es capturada.
Occurrence Target	El número de ocurrencias para la cantidad de objetivo de la característica.
Occurrences Held	Numero de ocurrencias de la característica capturada.
Target Met	'yes' si los dos objetivos anteriores se alcanzan, de lo contrario 'no'.
Target Zone1	La cantidad de objetivo para la característica en zona 1, indicada en zonetarget.dat
Amount Held Zone1	Cantidad de objetivo de la característica capturada en zona 1.
Contributing Amount Held Zone1	Cantidad de la característica capturada en zona 1 multiplicada por la fracción de contribución de la característica en zona 1.
Occurrence Target Zone1	Numero de ocurrencias de la cantidad de objetivo de la característica en zona 1, indicado en zonetarget.dat
Occurrences Held Zone1	El número de ocurrencias de la característica capturada en zona 1.
Target Met Zone1	'yes' si los dos objetivos anteriores se alcanzan, de lo contrario 'no'

Los seis campos especificados para la zona 1 se repiten para cada zona, substituyendo "Zone1" en el campo de nombre, por el nombre de la zona correspondiente.

Información de resumen

Nombre en input.dat: SAVESUM

Nombre del archivo de resultados: escenario_sum.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo contiene información resumida de cada corrida y contiene los siguientes títulos de columna que explican el contenido de cada fila: run no, score, cost (para cada costo indicado en el archivo pu.dat), planning units (por zona), boundary length (por zona), penalty, shortfall, missing values. Cada título se explica a continuación.

Título	Descripción
Run no	El número de corridas para esa línea.
Score	El puntaje de la función objetivo para la solución.
Cost	Valor de costo de la solución para cada costo en el archivo pu.dat.
Planning Unit Count (para cada zona)	Número de unidades de planificación en la solución para cada zona.
Planning Unit Cost (para cada zona)	El costo total de unidades de planificación para cada zona.
Boundary Length	La longitud de la frontera de la solución.
Penalty	El puntaje de penalización para características faltantes para producir la solución.
Shortfall	Shortfall es igual a la suma – de todas las características que no alcanzaron su objetivo - de los objetivos menos la cantidad en la solución. Si el objetivo de una característica se alcanza, no hay shortfall.
Missing Values	El número de características que no alcanzaron su objetivo.

Cuando hay varias características que no han alcanzado su objetivo, los últimos tres títulos son útiles para determinar qué tan pobre ha sido la corrida. Es posible tener una alta penalización y aun así estar muy cerca de alcanzar los objetivos, particularmente si los factores de penalización de la característica se han indicado muy altos. El shortfall es un buen indicador para saber si las características están muy cerca o muy lejos de alcanzar sus objetivos (el número de valores faltantes provee información adicional en este aspecto). Si hay cinco características cada una de las cuales no han alcanzado su objetivo, pero cuyos shortfall combinados resulta en un valor muy pequeño, esto entonces indicara que estas características están muy cerca de alcanzar su objetivo (por ejemplo 99% o más alto) y el usuario puede no tiene que preocuparse de esto.

Detalles del escenario

Nombre en input.dat: SAVESCEN

Nombre del archivo de resultados: escenario_sen.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.



Descripción: Este archivo contiene una lista documentada de las opciones que componen el escenario creado en el archivo de entrada de parámetros. Se usa para rastrear los detalles del escenario cuando se corren múltiples escenarios.

Solución resumida

Nombre en input.dat: SAVESUMSOLN

Nombre del archivo de resultados: escenario_ssoln.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo es la solución resumida de todas las corridas individuales en un escenario. Indica que tan frecuente fue incluida una unidad de planificación en una solución, a través de todas las corridas individuales. Adicionalmente, muestra que tan frecuente una unidad de planificación fue incluida en una zona individual (excepto para la zona disponible). Es una manera útil de explorar la irremplazabilidad de las unidades de planificación en una zona.

Matriz de Soluciones

Nombre en input.dat: SAVESOLUTIONSMATRIX

Nombre del archivo de resultados: escenario_solutionsmatrix.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo almacena cuales unidades de planificación fueron seleccionadas en cada zona para todas las corridas. Si el parámetro de SOLUTIONSMATRIXHEADERS, tiene un valor de 1, una línea de título se incluirá en el archivo. Si usted elige incluir la línea de título, una lista de los identificadores de las unidades de planificación se presentara en la línea. Hay una línea por cada corrida que contiene el identificador de zona para cada unidad de planificación, indicando en cual zona se ubicó cada unidad de planificación para la corrida correspondiente. Es una manera útil de exportar información sobre la selección de las unidades de planificación para el análisis de agrupamiento en R u otro paquete estadístico.

Penalización

Nombre en input.dat: SAVEPENALTY

Nombre del archivo de resultados: escenario_penalty.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo contiene la penalización computada por Marxan para todas las características. Corridas individuales en un escenario. Indica que tan alto ha sido el costo



de satisfacer los objetivos para una característica. Es útil para conocer la dificultad relativa que Marxan tiene al tratar de alcanzar los objetivos de las características.

Archivo de Pantalla de registro

Nombre en input.dat: SAVELOG

Nombre del archivo de resultados: escenario_penalty.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo muestra todas las pantallas de resultados cuando Marxan Z está corriendo los análisis.

Detalles de calibración

Nombre en input.dat: ANNEALINGTRACEROWS

Nombre del archivo de resultados: escenario_penalty.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo muestra todas las pantallas de resultados cuando se está corriendo análisis en Marxan Z.

Detalles del mejoramiento iterativo

Nombre en input.dat: ITIMPTRACEROWS

Nombre del archivo de resultados: escenario_penalty.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo muestra todas las la pantallas de resultados durante la ejecución de Marxan Z.

Resumen de Conectividad de Zona

Nombre en input.dat: SAVEZONECONNECTIVITYSUM

Nombre del archivo de resultados: escenario_penalty.dat, donde 'escenario' es el nombre dado por el usuario en el archivo de entrada de parámetros.

Descripción: Este archivo muestra todas las pantallas de resultados mostradas durante la ejecución de Marxan Z.

1.3 PANTALLA DE RESULTADOS

En el archivo de entrada de parámetros, el nivel de información mostrado en la pantalla de Marxan Z se puede determinar usando la variable VERBOSITY en la sección de controles de programa. Si se asigna un valor cero de VERBOSITY, no se mostrara ninguna información sobre el escenario. El nivel más alto posible es tres, y mostrara la información más detallada del escenario. Cualquier nivel mayor que cero, mostrara información básica resumida de cada corrida. El 2 incluirá información adicional sobre el éxito cargando los archivos para el análisis, número de líneas leídas con éxito, el puntaje de la configuración de zonas en diferentes puntos en una corrida, y otras medidas de tiempo. Cuando se use calibración flexible con este nivel de información, la temperatura inicial calculada y el decrecimiento también se muestran. El nivel 3 es típicamente más detallado de lo necesario; provee información sobre el puntaje de la actual configuración de zonas cada vez que la temperatura decrece cuando se usa calibración simulada.

Un ejemplo de una línea de un resumen de información básica (incluido en todos los niveles mayores a cero) es:

```
Run 1 Value 10 Cost 12 PU 100 Zonename2 50 Zonename3 25 Zonename4 25 Boundary 300 Missing 2  
Penalty 500
```

Run 1 indica que este es la primera corrida del algoritmo. El número de la corrida no siempre aparece en la misma línea con la valoración de la solución. Depende del nivel de detalle de la información.

Value 10 es el costo sumado al valor de la frontera, sumados a la penalización dadas las características faltantes. Dado que la penalización se incluye en este valor, el significado del valor requiere interpretación.

Cost 12 es el costo de la solución. Es la suma de los costos de todas las unidades de planificación en la solución, dada en las unidades de la variable 'cost field' en el archivo pu.dat

PU 100 es el número total de unidades de planificación en la solución.

Zonename x es el número de unidades de planificación en esa zona. Todas las zonas protegidas serán presentadas en la línea de resumen con el nombre de zona indicado en el archivo zones.dat las unidades de planificación en las zonas protegidas se sumara para igualar el número total de unidades de planificación en la solución.

Boundary 300 es el perímetro o longitud de frontera, de la solución.

Missing 2 es el número de objetivos que están sub-representados en la solución. Estos objetivos son las metas definidas en los archivos feat.dat y zonetarget.dat. Estos son



evaluados de acuerdo con el parámetro 'MISSLEVEL' en el archivo de parámetros input.dat

Penalty 500 es la penalización por no representar todas las características. Si todas son representadas, el valor será 0.

2.0 Función objetivo y Algoritmo usado

2.1 LA FUNCION OBJETIVO

La Función objetivo da un valor a cada configuración de zonificación de las unidades de planificación. Esto significa que a una sola unidad de planificación o a ninguna unidad de planificación se le puede dar un valor de Función objetivo. Una configuración que contenga cero unidades de planificación sería menos costosa de implementar, pero probablemente no alcanzará los objetivos de planificación y por tanto el valor de Función objetivo resultaría muy bajo.

Si tenemos una función objetivo que asigna un valor, o puntaje, a cada posible configuración de zonas, entonces podemos comparar la configuración de cualquier par de zonas y decir cuál es la mejor de las dos (de acuerdo con la función objetivo). Dado que el valor de la función objetivo puede ser evaluado por un ordenador, la puerta está abierta para usar un amplio rango de métodos para crear automáticamente configuraciones de zona que tengan buenos valores de Función objetivo.

En Marxan Z, esta función objetivo es usada por el algoritmo de mejoramiento iterativo, y por la calibración simulada. La función objetivo se diseñó con el propósito de integrarla con un optimizador de la calibración simulada, pero los dos son distintas entidades. Calibración simulada es un optimizador de propósito general y la función objetivo define las restricciones y objetivos para la configuración de una zona sin definir explícitamente como se encuentra una zonificación óptima.

La función objetivo consiste de dos secciones principales; la primera es una medida del costo de la configuración de la zona y la segunda es una penalización por no cumplir varios criterios. En su forma más simple, es una combinación del costo económico de la configuración de la zona y una penalización por no alcanzar todos los objetivos, si alguno no es alcanzado. Estos criterios pueden incluir un límite en el costo de la configuración de la zona y siempre incluye el nivel de representación del objetivo para cada característica. Así como esta es la medida opcional de la fragmentación de la configuración de la zona y un costo opcional de umbral de penalización. En esta función objetivo, entre más bajo el valor mejor la configuración de la zona:

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

‘Cost’ es la suma de cada costo multiplicado por el valor indicado en el archivo `zonecost.dat` de cada una de las unidades de planificación dentro de la configuración de zona.

'Boundary' es la longitud (o posible costo) de la frontera que rodea la configuración de la zona. La constante BLM es el multiplicador de la longitud de la frontera que determina la importancia dada a la longitud de frontera relativa al costo de la configuración de la zona. Si se asigna un valor de 0 al BLM entonces la longitud de la frontera no se incluye en la función objetivo.

El siguiente término es la penalización dada por no representar adecuadamente una característica, sumada sobre todas las características para cada zona. FPF es la sigla para 'feature penalty factor' que significa 'factor de penalización de característica' y es un factor de importancia para la característica, el cual determina la importancia relativa para satisfacer el objetivo de esa característica en particular. El término 'penalty' es una penalización asociada con cada característica sub-representada. Se expresa en términos de costo y longitud de la frontera y es aproximadamente el costo y la frontera adicional modificada, que son necesarios para capturar adecuadamente una característica que no ha sido adecuadamente representada en la configuración de la zona actual.

El término 'cost threshold penalty', significa penalización del umbral de costo, es una penalización aplicada a la función objetivo cuando se excede el costo del objetivo. Es una función del costo y posiblemente la longitud de la frontera del sistema, la cual en algunos algoritmos cambiara a medida que el algoritmo procede (este es el termino t en la formula anterior). Esta penalización también es opcional y puede ser excluida de la función objetivo.

Costo de la configuración de Zona

La función objetivo que usa Marxan Z, tiene la restricción de que el costo de la configuración de zona es la combinación lineal de costos de todas las unidades de planificación dentro de la configuración de zona, dividida por zona.

Longitud de la Frontera y Fragmentación

Dado que se incluye el término de longitud de frontera en la función objetivo podemos aplicar un control sobre el nivel de fragmentación en la configuración de la zona. Con el fin de permitir que la longitud de frontera sea agregada al cálculo del costo, se usa un factor multiplicativo. Esto es porque es muy probable que la longitud de frontera este expresada en unidades que son diferentes a la medida del costo. No solo las unidades son incompatibles sin un modificador de la longitud de frontera, sino que la importancia de la concisión en el costo no es inmediatamente obvia. Cambiar el modificador de la longitud de frontera permite a los planificadores explorar este problema.

Requerimientos de Representación para las características



En la función objetivo, si las características no alcanzan uno o más de sus requerimientos entonces atraerá una penalización dependiendo de qué tan sub-representada esta y el valor de la característica relativo a otras características.

Penalización de características

La penalización de características es la penalización dada a una configuración de zona, cuando no alcanza los objetivos de la característica. Se basa en el principio de que si una característica está por debajo del nivel de representación de su objetivo, entonces la penalización debería estar cerca al costo que incurra el incremento de la característica hasta el nivel de representación de su objetivo. Por ejemplo: si el requerimiento fuese el representar cada característica en al menos una oportunidad, entonces la penalización por no tener una característica dada sería el costo de la unidad más barata que contenga una vez la característica. Si hacen falta varias características entonces podría producir una configuración de zona que sea completamente representativa, agregando la unidad de planificación más barata que contenga cada una de las características faltantes. Esto no incrementaría el valor de la función objetivo para la configuración de zona, de hecho, si alguna de las unidades de planificación adicional tiene más de una de las características faltantes, entonces el valor de función objetivo disminuiría. Parecería ideal recalculer las penalizaciones después de realizar cada cambio a la configuración de zona. Sin embargo, esto requiere tiempo y resulta más eficiente trabajar con penalizaciones que cambian solo en la manera más simple de un punto del algoritmo al siguiente. Un algoritmo ambicioso se usa para calcular la forma más barata en la cual cada característica podría ser representada sola y esto forma la penalización base para cada característica. Marxan Z pone juntas las unidades de planificación más baratas que representan el objetivo.

Este método se describe mediante los siguientes pasos:

- I. Calcular un valor de 'costo por área' para cada unidad de planificación.
 - A. Determinar cuánto contribuye esta unidad de planificación al objetivo para una característica dada.
 - B. Determinar el costo económico de la unidad de planificación.
 - C. Determinar la longitud de frontera de la unidad de planificación.
 - D. El costo global es el costo económico + la longitud de frontera x BLM
 - E. Costo por hectárea es entonces el valor de la característica dividido por el costo global.

- II. Seleccionar la unidad de planificación con el costo por hectárea más bajo. Agregar su costo al costo total y el nivel de representación para la característica al nivel de representación total.
 - A. Si el nivel de representación es cercano al objetivo, entonces podría ser más barato elegir una unidad de planificación barata que tenga la cantidad requerida de dicha característica, sin importar el costo por área.

III. Continuar agregando estos totales hasta que se tenga una colección de unidades de planificación que representen adecuadamente la característica dada.

IV. La penalización para la característica es el costo total (incluyendo la longitud de frontera BLM) de estas unidades de planificación.

Por tanto, si una de las características estuviese completamente sub-representada, entonces la penalización sería la misma que el costo de agregar la serie de unidades de planificación, asumiendo que estas unidades están aisladas unas de otras, para poder establecer la longitud de frontera. Este valor es rápido de calcular pero tiende a ser más alto que el valor óptimo. Algunas veces hay maneras más eficientes de representar una característica, diferentes al algoritmo ambicioso, considere el siguiente ejemplo.

Ejemplo 1: La característica A aparece en un número de unidades de planificación, las mejores son:

Unidad de planificación	Costo	Cantidad de A representada
1	2	3
2	4	5
3	5	5
4	8	6

El objetivo para A es 10 unidades. Si usamos el algoritmo ambicioso representaríamos esto con las unidades 1, 2, y 3 (seleccionadas en ese orden) para un costo total de 11.0 unidades. Si elegimos solamente las unidades de planificación 2 y 3, aun representaríamos adecuadamente A, pero nuestro costo sería de 9 unidades.

Este ejemplo muestra un caso simple donde el algoritmo ambicioso no produce los mejores resultados. Este algoritmo es rápido y produce resultados razonables. El programa tiende a sobreestimar y nunca subestimar las penalizaciones cuando está usando el algoritmo ambicioso. No es deseable tener un valor de penalización que es muy bajo porque la función objetivo puede no mejorar cuando represente a las características en su totalidad. Si hay algunas características que no necesitan ser totalmente representadas, esto debería manejarse usando el factor de penalización de característica, que se describe más abajo. No es un problema tener penalizaciones que son más altas de lo necesario, a veces esto es deseable. El costo de frontera de una unidad de planificación los pasos descritos anteriormente es la suma de todas sus fronteras. Esto asume que la unidad de planificación no tiene fronteras comunes con el resto de la configuración de la zona y por tanto tendera nuevamente a sobreestimar el costo de la unidad de planificación y la penalización.

La penalización es calculada y fijada en el momento inicial en el que se aplica el algoritmo. Es aplicada de manera directa - si una característica ha alcanzado la mitad de



su objetivo entonces recibe la mitad de su penalización. El problema con esto es que usted se puede encontrar en una situación donde solo necesita una pequeña cantidad para alcanzar el objetivo de la característica, pero que no hay manera de hacer esto, lo que podría disminuir el valor del objetivo. Si tomamos el ejemplo 1 una vez más, la penalización para la característica A es 11 unidades. Si usted ya tiene las unidades de planificación 1 y 4 en la configuración de la zona, entonces tiene 9 unidades de la característica y la penalización es $11.0 \times (10-9)/10 = 1.1$ unidades. Entonces la característica atrae una penalización de 1.1 unidades y necesita solamente 1 unidad más para alcanzar su objetivo. No hay unidad de planificación con un costo así de bajo - la adición de cualquiera de las unidades de planificación restantes costaría a la configuración de zona mucho más que el aumento en la reducción de penalización.

Este problema se puede solucionar estableciendo un FPF (Feature Penalty Factor) más alto para todas las características. El FPF es un factor multiplicativo para cada característica, descrito más abajo.

Es posible que el objetivo para una característica se establezca más alto de lo que podría posiblemente alcanzarse. En Australia donde los requerimientos del JANIS (1997) establecen que el 15% del área pre-Europea de cada tipo de ecosistema de bosque debería estar protegido, podemos fácilmente tener objetivos que sean mayores que el área actual de cada ecosistema de bosque. En realidad, cuando este es el caso, el algoritmo escala la penalización, de tal manera que si, por ejemplo, el costo es de 100 unidades para proteger todo un ecosistema dado, pero eso representa solo la mitad del objetivo, entonces la penalización inicial será de 200 unidades. Esto significa que si usted se encuentra a medio camino de su objetivo la penalización para esa característica será la mitad de la penalización máxima, sin importar que tan alto sea el objetivo o si es un objetivo razonable.

Factor de penalización de característica

El factor de penalización de característica (FPF) es un factor multiplicativo que puede ser único para cada característica. Se basa inicialmente en el valor relativo de la característica, pero incluye una medida de cuán importante es representar esa característica en su totalidad. El efecto que tendrá varía entre los métodos que usan la función objetivo. Si es menor que 1, el algoritmo puede negarse a sumar una unidad de planificación para proteger esa característica si no hay otras características en la unidad de planificación. Un algoritmo puede fallar en la representación de características, acercándose pero no por encima del valor del objetivo. Para asegurar que cada característica alcance su objetivo puede ser a veces deseable establecer el FPF a un valor mayor que 1.

Penalización del umbral de costo

El umbral de costo es una opción que permite al usuario limitar la configuración de la zona a un costo establecido con longitud de frontera modificada. Esto ha sido incluido



para permitir una visualización del problema en una versión reversa. El reverso del problema sería encontrar la configuración de zona que tiene la mejor representación para todas las características contenidas por el máximo costo. Trabaja aplicando una penalización si el valor del umbral dado es excedido. La penalidad es la cantidad por la cual el umbral es excedido, multiplicado por el costo de penalidad del umbral. La penalidad depende del estado de calibración del algoritmo (que tan avanzado en el proceso de calibración está el sistema, expresado como una proporción). El multiplicador es:

$$\text{Penalización del Umbral de Costo} = \text{Cantidad sobre el Umbral} \times (Ae^{bt} - A).$$

Aquí, t es el tiempo durante el análisis y varía entre 0 y 1. A y B son los controles de parámetros. B controla que tan pronunciada es la curva (un valor alto de B provocará que el multiplicador varíe poco hasta avanzado el análisis). A controla el valor final. Un valor alto de A penalizará severamente cualquier exceso del umbral, un A más bajo podría permitir que el umbral sea excedido levemente. El multiplicador empieza en 0 cuando t es cero. Ambos A y B requieren cierta experimentación para ser establecidos. La penalización de umbral de costo es construida en la función objetivo y por tanto se aplicara al módulo de calibración junto con el módulo de mejoramiento iterativo.



2.2 METODOS DE ALGORITMOS

Hay dos algoritmos básicos que pueden ser usados para formular una solución en Marxan Z: calibración simulada, y mejoramiento iterativo. Cada uno de esos algoritmos se puede usar solo o en combinación con el otro. El tipo de análisis seleccionado en el archivo de entrada de parámetros determina cual algoritmo, o combinación de algoritmos, se usará para formular una solución en Marxan Z.

Calibración Simulada

La calibración simulada se basa en un mejoramiento iterativo con aceptación estocástica de malos movimientos para ayudar a evitar quedarse estancado en un mínimo local. La implementación usada en Marxan Z repetirá el análisis un número dado de veces. Por cada repetición, una unidad de planificación es elegida al azar y puede o no estar ya incluida en la configuración de zona. El cambio del valor de configuración de zona - el cual ocurriría si esta unidad de planificación fuese agregada o removida del sistema - se evalúa tal como se hace con el mejoramiento iterativo. Este cambio es combinado con un parámetro llamado temperatura y luego comparado con un número uniforme al azar. La unidad de planificación puede ser luego agregada o removida del sistema dependiendo de esta comparación.

La temperatura empieza en un valor alto y disminuye durante el algoritmo. Cuando la temperatura es alta, al inicio del proceso, buenos y malos cambios se aceptan. A medida que la temperatura disminuye la posibilidad de aceptar un cambio malo disminuye hasta que, finalmente, solo buenos cambios son aceptados. Por simplicidad, el algoritmo debería terminar antes de que solo acepte buenos cambios y el mejoramiento iterativo debería seguir, porque en este punto el algoritmo de calibración simulada se comporta como un algoritmo de mejoramiento iterativo ineficiente.

Hay dos tipos de calibración simulada en Marxan Z. Una es la calibración de agenda fija, en la cual la agenda de calibración (incluyendo la temperatura inicial y la tasa de temperatura decrece) se fija antes de que el algoritmo empiece. La otra es la calibración de agenda flexible, en la cual el algoritmo muestrea el problema y establece la temperatura inicial y la tasa a la que la temperatura decrece en base a su muestreo.

Calibración de agenda flexible

La calibración de agenda flexible comienza con número de veces del muestreo (número de iteraciones/100). Luego se establece la temperatura objetivo final como el cambio mínimo positivo (menos malo) que ocurre durante el periodo de muestreo. El máximo se establece de acuerdo a la fórmula:

$$T_{ini} = \text{cambio mínimo} + 0.1 \times (\text{Max cambio} - \text{Min cambio})$$



Esto se basa en la agenda flexible en Conolly (1990). Aquí, Tini es la temperatura inicial, los cambios (min y max) son los cambios malos mínimo y máximo que ocurrieron. En nuestro caso un cambio malo es el que incrementa el valor de la función objetivo (valor positivo).

Calibración de agenda fija

Con la calibración de agenda fija los parámetros que controlan la agenda de calibración se fijan para cada implementación del algoritmo. Esto se hace tradicionalmente haciendo algunos ensayos del algoritmo con diferentes parámetros para un número de iteraciones que es un orden de magnitud más corto que el número que será usado en el análisis final. Los parámetros a menudo se beneficiarían de ser cambiados para análisis más largos, pero basados en los ensayos. Los ensayos incluyen mirar los resultados finales y también rastrear el progreso de cada repetición del análisis de manera individual. Las agendas de calibración que resultan del proceso de agenda de calibración fija son generalmente superiores que los de agendas de calibración flexibles. La calibración flexible es ventajosa porque no requiere un usuario hábil para usar el algoritmo y porque es más rápida. Es más rápida en términos del tiempo requerido para procesar, debido a que hay mucho menos en el camino de los análisis iniciales, es considerablemente más rápido. Por esta razón es el principal algoritmo que será examinado aquí. Planificadores de uso de suelo y administradores tenderán a usar las opciones estándar y métodos automáticos en su mayoría, de tal manera que la habilidad de la calibración flexible para diseñar la configuración de zona es muy útil y definitivamente importante. La calibración flexible también es importante para investigaciones, evaluaciones y ensayos en el sistema que precedería el uso más cuidadoso y detallado de un algoritmo de agenda de calibración fija.

Estableciendo una calibración de agenda fija

Cuando se establece una agenda fija los dos parámetros que se deben cambiar son la temperatura inicial y final. La temperatura final es establecida a un valor apropiado para la función de enfriamiento. Si la temperatura final es muy baja el algoritmo usará mucho tiempo en un mínimo local incapaz de mejorar el sistema y continuará intentándolo. Si la temperatura final es muy alta gran parte del trabajo de calibración no será completado y la configuración de la zona será en su mayoría derivada de una agenda de mejoramiento iterativo que sigue calibración simulada y encuentra un mínimo local 'al azar'. Si la temperatura inicial es demasiado alta el sistema estará demasiado tiempo en altas temperaturas cercanas al equilibrio y menos tiempo donde la mayoría del trabajo de calibración debe ser hecho.

Por tanto, la mejor manera de tener una idea general de los valores que deberían tener los parámetros, es realizar un análisis de algoritmo y muestrear el valor del sistema regularmente para ver cuando alcanza el equilibrio a varias temperaturas, cuales son y en qué punto el sistema deja de cambiar o mejorar. Esto facilita el establecimiento de una



temperatura final provisional y también da estimadores de cuál debería ser la temperatura inicial razonable. De aquí en adelante las pruebas se pueden realizar observando el resultado final de múltiples análisis y diferentes parámetros a un número más pequeño de iteraciones.

Una vez que se han encontrado buenos valores, estos se deben escalar a un número mayor de iteraciones. Esto es porque el tiempo que el sistema permanece a temperaturas más bajas o críticas es importante y direccionara la búsqueda de buenos parámetros. Extender la duración del algoritmo incrementara el tiempo gastado a esas temperaturas más de lo necesario. Por tanto el método usado es el de mantener la temperatura final constante e incrementar el nivel de la temperatura inicial de tal manera que emplee un tiempo similar a bajos niveles pero permita buscar el espacio global a una escala mayor. Para un análisis corto es a menudo mejor tener el sistema corriendo a algunas temperaturas críticas por el tiempo que sea posible. Para un análisis más largo, es ventajoso incrementar el rango de temperaturas usadas.

Mejoramiento iterativo

El mejoramiento iterativo es un método simple de optimización. Se ha reemplazado ampliamente por la calibración simulada pero aún puede ser usado en beneficio de otros algoritmos. Hay tres tipos básicos de mejoramiento iterativo que pueden ser usados en Marxan Z. Ellos se diferencian en la serie de cambios posibles que son considerados en cada paso. Cada uno de ellos empieza con una 'semilla' de solución. Esto puede ser cualquier configuración de zona con algunas, todas, o ninguna unidad de planificación asignadas a zonas. Es útil usar el resultado final de otro algoritmo tal como la calibración simulada y como la solución inicial para el mejoramiento iterativo. En este caso el algoritmo de mejoramiento iterativo se usa solamente para asegurar que ningún mejoramiento simple sea posible.

En cada iteración, el algoritmo considerara un cambio al azar para evaluar si realizar este cambio mejora el valor de la función objetivo. Si el cambio mejora el sistema entonces se realiza tal cambio, de lo contrario, otro que no se haya evaluado se evaluará al azar. Esto continúa hasta que cada cambio posible ha sido considerado y ninguno mejora el sistema. La configuración de zonas resultante es un mínimo local (u optimo local).

Los tres tipos básicos de mejoramiento iterativo difieren en el tipo de cambios que consideran. El tipo más simple se llama 'mejoramiento iterativo normal' y el único cambio que es considerado es agregar o remover cada unidad de planificación de la configuración de zonas. Esta es la misma 'serie de movimientos' considerada por el algoritmo ambicioso y por la calibración simulada.

El segundo tipo de mejoramiento iterativo se llama 'permuta' y selecciona unidades de planificación al azar, si la unidad de planificación seleccionada puede mejorar el sistema al ser adicionada o removida de él, entonces esto se realiza, de lo contrario se considera



un intercambio. Si la unidad de planificación elegida ya se encuentra en la configuración de zonas, entonces los cambios considerados son remover esa unidad de planificación pero agregar otra en algún otro lugar. Si la unidad de planificación elegida no está en la configuración de zona entonces los cambios considerados son agregar ésta al sistema pero removiendo una que ya está en el sistema. Cada 'permuta' posible se considera al azar deteniéndose cuando se encuentra una que mejora el sistema. Dado que el número de intercambios posibles puede ser muy grande, esta opción es mucho más lenta.

El tercer tipo es llamado 'dos permutas', en este método además de evaluar cada unidad de planificación (al azar) para ver si agregar o remover tal unidad mejora el sistema, se considera cada combinación posible con dos cambios. Estos cambios incluyen, agregar o remover la unidad de planificación elegida en conjunto con agregar o remover cada otra unidad de planificación. El número de tales movimientos es aún más grande que aquel en el método de 'permuta', así que este método debería ser usado con precaución.

Hay una cuarta opción que es realizar el método normal primero, para obtener un buen óptimo local, y luego usar el método de 'dos permutas'. Dado que el número de mejoramientos que encuentra el método de 'dos permutas' debería ser mucho más pequeño después de que el algoritmo de mejoramiento iterativo normal ha pasado sobre la solución 'semilla' entonces esto debería ser mucho más rápido, que realizar un análisis de 'dos permutas'.

La principal ventaja del mejoramiento iterativo es que el elemento azaroso le permite producir múltiples soluciones. En promedio las soluciones pueden ser pobres, pero si puede producir diferentes soluciones suficientemente rápido, entonces puede producir algunas muy buenas al cabo de muchas repeticiones. En teoría es posible alcanzar el mínimo global realizando un análisis de mejoramiento iterativo empezando ya sea con una configuración de zona vacía o una situación donde cada unidad de planificación empieza en una zona seleccionada al azar.

El principal uso del mejoramiento iterativo será el de seguir un algoritmo diferente para un pulido final a escala. Esto es particularmente cierto en el caso de los métodos de 'permuta' y 'dos permutas'. Aun cuando sigue otro algoritmo, estos pueden requerir largos periodos de tiempo.



3.0 AGRADECIMIENTOS

El proceso de mejoramiento y la plataforma de apoyo del programa Marxan han sido posible gracias al apoyo financiero de generosas organizaciones de diferentes partes del mundo que han hecho sus donaciones.

Queremos expresar nuestro agradecimiento a Ecotrust, organización que ha otorgado financiamiento substancial para el desarrollo del programa. También, a la Organización de Patrimonio Nacional por su substancial contribución financiera para el desarrollo del programa como tal, así como el esquema de investigación que dio origen a este programa. Así mismo agradecemos el apoyo financiero que actualmente recibimos del Centro de Análisis Aplicado para la toma de decisiones Medioambientales.



4.0 REFERENCIAS CLAVE

Literatura arbitrada y reportes en los que se ha utilizado Marxan se han compilado en el Centro de Ecología de la Universidad de Queensland. Una lista actualizada se puede consultar en la página web de Marxan:

<http://www.uq.edu.au/marxan/index.html?page=80365&p=1.1.6.3>

Airame, S. 2005. Channel Islands National Marine Sanctuary: Advancing the Science and Policy of Marine Protected Areas. Pages 91-124 in A. Scholz and D. Wright, editors. Place Matters: Geospatial Tools for Marine Science, Conservation, and Management in the Pacific Northwest. Oregon State University Press, Corvallis.

Ardon, J.A., J. Lash, D. Haggarty. 2002. Modelling a Network of Marine Protected Areas for the Central Coast of BC, version 3.1. Living Oceans Society, Sointula, BC, Canada.

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.

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

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.

Chan, A., A. Cundiff, N. Gardner, Y. Hrovat, L. Kircher, C. Klein. 2006. Marine Protected Areas Along California's Central Coast: A Multicriteria Analysis of Network Design. Thesis. University of California, Santa Barbara.

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

Cook, R. R. (2006) Developing Alternatives for Optimal Representation of Seafloor Habitats and Associated Communities in Stellwagen Bank National Marine Sanctuary

Connolly, D. J. (1990) "an improved annealing scheme for QAP", European Journal of Operations Research, 46, 93-100.

Ferdaña, Z. (2005) Nearshore marine conservation planning in the Pacific Northwest: Exploring the use of a siting algorithm for representing marine biodiversity, in



- Wright, D.J. and Scholz, A.J. (eds.), "Place Matters: Geospatial Tools, for Marine Science, Conservation, and Management in the Pacific Northwest," Corvallis, OR: OSU Press
- Game, E. T. and H. S. Grantham. (2008). Marxan User Manual: For Marxan version 1.8.10. University of Queensland, St. Lucia, Queensland, Australia, and Pacific Marine Analysis and Research Association, Vancouver, British Columbia, Canada.
- Geselbracht, L., R. Torres, G.S. Cumming, D. Dorfman, M. Beck. 2005. Marine/Estuarine Site Assessment for Florida: A Framework for Site Prioritization. The Nature Conservancy. Accessed on May 9, 2006 at <http://myfwc.com/wildlifelegacy/PDF/MarineSitePrioritizationFrameworkfinalrptPart1.pdf>.
- JANIS (1997) Nationally agreed criteria for the establishment of a comprehensive, adequate and representative reserve system for forests in Australia. A report by the Joint ANZECC/MCFFA National Forest Policy Statement Implementation Subcommittee. National forest conservation reserves: Commonwealth proposed criteria. Commonwealth of Australia, Canberra.
- Kirkpatrick, S., Gelatt jr, C. D., and Vecchi, M. P. (1983) Optimization by Simulated Annealing. *Science*, **220**, 671-680.
- Leslie, H.; M. Ruckelshaus, I. Ball, S. Andelman, H. Possingham. 2003. Using Siting Algorithms in the Design of Marine Reserve Networks. *Ecological Applications* 13(1) S185-S198.
- Lewis, A., S. Slegers, D. Lowe, L. Muller, L. Fernandes, J. Day. 2003. Use of Spatial Analysis and GIS techniques to re-zone the Great Barrier Reef Marine Park. Coastal GIS Workshop.
- 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.
- Possingham, H., Ball, I. R., Andelman, S. (2000) "Mathematical methods for identifying representative reserve networks" in *Quantitative methods for conservation biology*. Ferson, S., and Burgman, M. (eds). Springer-Verlag, New York.
- 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. ASOR, Adelaide, pp. 536-545.

Pressey, R. L., Humphries, C. J., Margules, C. R., Vane-Wright, R. I., and Williams, P. H., (1993) "Beyond opportunism: Key principles for systematic reserve selection", *TREE*, **8**, 124-128

Pressey, R. L., Johnson, I. R., and Wilson, P. D., (1994) "Shades of irreplaceability: towards a measure of the contribution of sites to a reservation goal."

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

Pressey, R. L., Possingham, H. P., Margules, C. R., (1996) "Optimality in Reserve Selection Algorithms: When Does it Matter and How Much?" *Biological Conservation* **76**, 259-267.

Richardson, E.A., M.J. Kaiser, G. Edwards-Jones, and H.P. Possingham. Sensitivity of marine-reserve design to the spatial resolution of socioeconomic data. *Conservation Biology*, in press. Royal Commission on Environmental Pollution. 2004. *Turning the tide: Addressing the impact of fisheries on the marine environment*. Page 497. RCEP, London.

Sala, E., O. Aburto-Oropeza, G. Paredes, I. Parra, J.C. Barrera, P.K. Dayton. 2002. A General Model for Designing Networks of Marine Reserves. *Science* 298: 1991-1993.

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

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

Stewart, R.R. M. Watts, T.J. Ward and H.P. Possingham (in prep). Identifying the optimal trade-off effects when designing a Multiple-Use Marine Park with multiple objectives

Stewart, R.R., L. Kircher, M. Watts, T.J. Ward and H.P. Possingham (in prep). A systematic planning framework to manage the multiple uses of coastal and marine resources and minimize potential for conflict

Stoms, D.M. 1994. Scale dependence of species richness maps. *Professional Geographer* 46: 346-358.

Warman, L.D., A.R.E. Sinclair, G.G.E. Scudder, B. Klinkenberg, R.L. Pressey. Sensitivity of Systematic Reserve Selection to Decisions about Scale, Biological Data, and Targets: Case Study from Southern British Columbia. *Conservation Biology* 18: 655-666.

Watts, M. E., I. R. Ball, H. P. Possingham et al (2008). Marxan with zones - optimised landscape and seascape zoning using spatially explicit annealing, in prep.



Apéndice A Ejemplo del archivo de entrada de Parámetros

Parámetros Generales

BLM 0
PROP 0.5
RANDSEED -1
NUMREPS 100

Parámetros de calibración

NUMITNS 1000000
STARTTEMP -1
COOLFAC 0
NUMTEMP 10000

Archivos de entrada

INPUTDIR input
FEATNAME feat.dat
PUNAME pu.dat
PUVFEATNAME puvfeat.dat
BOUNDNAME bound.dat
ZONESNAME zones.dat
COSTSNAME costs.dat
ZONETARGETNAME zonetarget.dat
ZONECONTRIBNAME zonecontrib.dat
ZONECOSTNAME zonecost.dat

Guardar archivos

SCENNAME california
SAVERUN 2
SAVEBEST 2
SAVESUM 2
SAVESCEN 2
SAVETARGMET 2
SAVESUMSOLN 2
SAVELOG 1
OUTPUTDIR output

Control de programa.

RUNMODE 1
MISSLEVEL 1
ITIMPTYPE 0
VERBOSITY 2