

---

# EVALUACIÓN DE TRES TÉCNICAS HEURÍSTICAS PARA RESOLVER UN MODELO DE PLANIFICACIÓN DEL APROVECHAMIENTO EN PLANTACIONES FORESTALES INDUSTRIALES

MARÍA ALEJANDRA QUINTERO M, MAURICIO JEREZ y MAGDIEL ABLAN

---

## RESUMEN

Se desarrollaron tres algoritmos heurísticos basados en las técnicas de Recocido Simulado, Búsqueda Tabú y Algoritmos Genéticos, para resolver un problema de planificación del aprovechamiento forestal que consiste en obtener una secuencia de cortas para 20 rodales, en un periodo de planificación de siete años, tal que los costos de aprovechamiento sean mínimos y se satisfaga la demanda anual de madera de una fábrica de pulpa para papel. A fin de evaluar el desempeño de las técnicas heurísticas utilizadas y hacer comparaciones, el problema también fue resuelto utilizando una técnica de programación matemática (programación entera binaria) que permitió obtener la solución óptima. Se realizaron 100 corridas de los algoritmos heurísticos y se calcularon estadísticas básicas para las variables: valor objetivo, error relativo respecto a la solución

óptima, iteración en que se consigue la mejor solución, y tiempo de computación utilizado. Las tres técnicas mostraron buen desempeño para resolver el problema planteado; su efectividad para encontrar la solución fue 94,7% en Recocido Simulado, 91,0% para Algoritmos Genéticos y 86,2% en Búsqueda Tabú. Todos los algoritmos heurísticos emplearon tiempos de computación considerablemente menores que el requerido por la programación entera binaria. Debido a su capacidad para encontrar buenas soluciones en tiempos de computación razonables, las técnicas heurísticas representan un enfoque que puede ser utilizado en la resolución de problemas de planificación forestal, especialmente aquellos difíciles de resolver mediante técnicas de programación matemática clásicas por ser complejos o requerir tiempos de cómputo muy elevados.

La planificación de las actividades forestales comprende un conjunto de decisiones referentes a la selección de los regímenes de manejo a utilizar, el diseño de planes de corta, la programación de trabajos en el campo y otros aspectos estratégicos, tácticos y operacionales, a fin de asegurar el cumplimiento de los objetivos establecidos. En la toma de decisiones deben considerarse factores biológicos tales como el crecimiento, ambientales como el impacto que puede causar el manejo de un bosque particular, y

financieros como por ejemplo el costo de cultivar y aprovechar un bosque, o los beneficios que se esperan obtener de una inversión. Estos factores están relacionados entre sí, lo que hace de la planificación forestal una tarea compleja.

La elaboración de planes forestales puede abordarse mediante técnicas de programación matemática. Entre las más utilizadas se encuentran la programación lineal y sus variantes (programación entera, programación por metas, programación lineal), la programación dinámica y la progra-

mación no lineal, las cuales son de gran ayuda pues permiten optimizar el uso de los recursos y facilitan la evaluación de diferentes escenarios (Bettinger y Chung, 2004). La principal ventaja de utilizar técnicas de programación matemática convencionales es que puede garantizarse que los resultados obtenidos corresponden a la solución óptima del modelo planteado; además, existe una gran cantidad de software de optimización que cuenta con algoritmos contrastados y de propósito general desarrollados para encontrar soluciones. Sin embargo, cuando el pro-

---

**PALABRAS CLAVE / Algoritmos Genéticos / Búsqueda Tabú / Planificación Forestal / Programación Entera Binaria / Recocido Simulado / Técnicas Heurísticas /**

Recibido: 31/05/2010. Modificado: 11/04/2011. Aceptado: 15/04/2011.

**María Alejandra Quintero M.** Ingeniera de Sistemas y M.Sc. en Estadística Aplicada, ULA, Venezuela. Profesora de la Escuela de Ingeniería Forestal, Facultad de Ciencias Forestales y Ambientales, ULA, Venezuela. e-mail: mariaq@ula.ve

**Mauricio Jerez.** Ingeniero Forestal y M.Sc. en Estadística Aplicada, ULA, Venezuela. Ph.D. en Manejo Ecológico de Bosques, Louisiana State University, EEUU. Profesor del Centro de Estudios Forestales y Ambientales de Postgrado (CEFAP), Facultad de Ciencias Forestales y Ambientales, ULA, Venezuela. e-mail: jerez@ula.ve

**Magdiel Ablan.** Ingeniera de Sistemas, ULA, Venezuela. Ph.D. en Ciencias del Ambiente, University of North Texas, EEUU. Profesora del Centro de Simulación y Modelos (CESIMO), Facultad de Ingeniería, ULA, Venezuela. e-mail: mablan@ula.ve

---

blema tiene un alto grado combinatorio y se encuentra dentro de la categoría “difícil de resolver” (*NP-hard*), estas técnicas presentan limitaciones: los algoritmos no pueden encontrar la solución óptima o requieren de tiempos de computación muy altos (Zanackis y Evans, 1981; Dréo *et al.*, 2006).

Esta situación puede presentarse en los modelos de planificación del aprovechamiento o planificación de cosechas forestales, que generalmente incluyen variables binarias para representar la asignación de tratamientos o programas de corta a diferentes unidades de manejo (rodales), y suele ocurrir que algunas instancias del problema pueden ser resueltas eficientemente con técnicas de programación matemática, pero si su tamaño aumenta, al considerar mayor cantidad de rodales, incrementar el número de restricciones o prolongar el período de planificación, la habilidad de los algoritmos exactos para resolverlo se ve comprometida, debido a que el número de posibles soluciones crece exponencialmente (Bettinger *et al.*, 2009b). Varios estudios han demostrado que algunos problemas de planificación del aprovechamiento forestal pueden ser considerados difíciles de resolver, y que para encontrar una solución deben emplearse técnicas diferentes a la programación matemática (Loockwood y Moore, 1993; Dahlin y Sallnas, 1993; Boston y Bettinger, 1999; McDill y Braze, 2001; Murray y Weintraub, 2002).

Otros problemas de difícil solución en planificación forestal son los que tienen un enfoque multiobjetivo, donde se busca optimizar varios objetivos en forma simultánea, como por ejemplo beneficios económicos, conservación de la biodiversidad, fijación de carbono, protección del suelo, regulación de recursos hídricos, recreación, etc. También son complejos los problemas de transporte forestal y los que incluyen un conjunto de restricciones espaciales relacionados a la forma, ubicación y yuxtaposición de las unidades de manejo. Para abordar este tipo de problemas, se puede hacer uso de técnicas heurísticas (Bettinger *et al.*, 2009a; Shan *et al.*, 2009).

Las técnicas heurísticas son procedimientos de solución conformados por una regla o conjunto de reglas que buscan buenas soluciones de una manera sencilla y rápida. En el campo de la optimización, las heurísticas pueden encontrar una solución de alta calidad a un costo computacional razonable, pero sin garantizar optimalidad o factibilidad, e incluso en algunos casos, no se llega a establecer lo cerca que se está de dicha situación (Reeves, 1996). Estas técnicas han sido utilizadas sobre todo en problemas difíciles de resolver, demostrando que pueden encontrar buenas soluciones mucho más rápidamente que los métodos de programación matemática, por lo que se pre-

fieren para la resolución práctica de un problema (Voß, 2001).

El destacable éxito de los métodos heurísticos para tratar problemas de optimización complejos ha generado creciente interés en los últimos años y una cantidad considerable de nuevas aplicaciones, entre las que se encuentran algunas experiencias en el sector forestal. Tres de las heurísticas que se han destacado en la solución de problemas de planificación forestal complejos son el Recocido Simulado (Lockwood y Moore, 1992; Tarp y Helles, 1997; Rodrigues *et al.*, 2004b; Crowe y Nelson, 2005; Chen y Gadow, 2008), la Búsqueda Tabú (Bettinger *et al.*, 1997, 2007; Brumelle *et al.*, 1998, Richards y Gunn, 2000, 2003; Rodrigues *et al.*, 2003; Díaz *et al.*, 2007), y los Algoritmos Genéticos (Lu *et al.*, 2000; Falcão y Borges, 2001; Ducheyne *et al.*, 2004; Rodrigues *et al.*, 2004a; Thompson *et al.*, 2009). También existen trabajos donde se hacen comparaciones entre distintas heurísticas para tratar problemas de planificación forestal, tales como los de Boston y Bettinger (1999), Falcão y Borges (2002), Palahí *et al.* (2004), Heinonen y Pukkala (2004), Pukkala y Kurttila (2005), Liu *et al.* (2006) y Zhu y Bettinger (2008).

En este artículo se utilizan las técnicas heurísticas Recocido Simulado, Búsqueda Tabú y Algoritmos Genéticos para resolver un problema de planificación de cosechas en plantaciones forestales industriales, en el que se busca obtener una secuencia de cortas para un conjunto de rodales durante un período de planificación de 7 años, minimizando los costos totales de aprovechamiento y considerando algunas restricciones operativas. El problema básicamente es el mismo que el tratado por Chiari *et al.* (2008) mediante un modelo de programación lineal entera binaria, aunque con datos actualizados. Algunos de los inconvenientes que la programación entera presenta para resolver este modelo se refieren al hecho que ante cambios en ciertas variables o supuestos, no puede encontrar la solución óptima. El objetivo de este trabajo fue evaluar el desempeño de algunas heurísticas en la resolución del problema de planificación de cosechas planteado, como técnicas alternativas de solución que puedan ser utilizadas bajo diferentes escenarios y en los casos en los que la programación entera binaria no puede encontrar la solución en un tiempo satisfactorio. Para ello se desarrollaron tres algoritmos, cada uno basado en uno de los métodos heurísticos antes mencionados. Se realizaron un conjunto de corridas cuyos resultados permitieron calcular estadísticas para distintas variables de desempeño.

Con el fin de validar los resultados obtenidos por las técnicas heurísticas, se resolvió el problema utilizando valores de las variables de entrada que permi-

ten conseguir una solución óptima mediante programación entera binaria. De acuerdo a Bettinger *et al.* (2009b) el nivel más estricto en la validación de las técnicas heurísticas es contrastar las soluciones que éstas encuentran con la solución óptima de una instancia problema que pueda ser resuelto con un método exacto; luego, si su desempeño es satisfactorio, las heurísticas pueden ser aplicadas a otras instancias del problema. Idealmente, todas las instancias del problema deberían poder resolverse con una técnica de programación matemática; sin embargo, esto no es lo que ocurre en problemas complejos. Aplicando esta técnica de validación, las soluciones generadas por los algoritmos heurísticos desarrollados en este trabajo fueron comparadas con la solución óptima obtenida mediante programación entera binaria.

## Materiales y métodos

### Formulación del problema

El problema consiste en cómo planificar un sistema de cortas para una unidad de producción forestal que permita abastecer de madera en forma continua a una planta de pulpa para la producción de cartón, de manera tal de minimizar los costos totales de aprovechamiento. Se trata de plantaciones de *Eucalyptus* (*E. urophylla*, *E. grandis*, *E. xurograndis*) dispuestas en 20 rodales, que a su vez están agrupados en diferentes fincas. De cada rodal se conoce su superficie (ha), la curva de incremento medio anual del volumen aprovechable (IMA en  $m^3 \cdot ha^{-1} \cdot año$ ) y la edad en años (Tabla I). El turno mínimo para la corta de un rodal es a los seis años y el turno óptimo a los siete años. A partir de la información básica de los rodales se calculó el volumen de madera disponible en cada rodal para cada año. El aprovechamiento anual de madera puede realizarse en uno o varios rodales simultáneamente, buscando cubrir la demanda de la planta de procesamiento, fijada en 90000ton/año. Se considera que un rodal sólo puede ser aprovechado una vez durante el período de planificación o también pudiera no ser aprovechado.

Los costos totales de aprovechamiento en un año dado incluyen costo de cosecha, costo de transporte de la madera desde los diferentes rodales aprovechados hasta la planta, y costo de oportunidad, referido este último a la cantidad de dinero que se dejaría de ganar si se corta un rodal en un año distinto al turno óptimo. El costo de cosecha se asumió constante para todos los rodales, los de transporte dependen de la distancia de cada rodal a la planta, y los de oportunidad fueron calculados de acuerdo al enfoque de alternativas financieras utilizado por Chiari *et al.* (2008).

El objetivo es obtener un plan de aprovechamiento de costo mínimo

TABLA I  
DATOS BÁSICOS DE LOS RODALES

Rodal	Edad (años)	Área (ha)	IMA (m³ha⁻¹año⁻¹)
1	6	245	24
2	7	78	27
3	2	315	28
4	8	958	8
5	7	305	15
6	3	340	15
7	2	134	15
8	6	507	10
9	4	276	14
10	3	171	14
11	2	175	14
12	2	154	23
13	8	156	19
14	5	310	22
15	4	385	22
16	2	98	29
17	1	336	29
18	9	528	16
19	9	528	16
20	9	528	16

que indique qué rodales se deben cortar en cada año de un horizonte de planificación de siete años para cubrir las necesidades de materia prima de la planta.

#### Modelo matemático

El modelo consta de una función objetivo que representa la minimización del costo total de aprovechamiento en el que se incurre durante todo el horizonte de planificación, es decir,

$$\text{Minimizar } Z = \sum_{i=1}^{20} \sum_{j=1}^7 C_{ij} \times X_{ij} \quad (1)$$

sujeta a las siguientes restricciones:

$$\sum_{i=1}^{20} V_{ij} \times X_{ij} \geq 90000 \quad j=1, 2, \dots, 7 \quad (2)$$

$$\sum_{j=1}^7 X_{ij} \leq 1 \quad i=1, 2, \dots, 20 \quad (3)$$

$$X_{ij} = (0, 1) \quad i=1, 2, \dots, 20; \quad j=1, 2, \dots, 7 \quad (4)$$

donde  $C_{ij}$ : costo total de aprovechamiento (costos de cosecha + transporte + oportunidad) si se corta el rodal  $i$  en el año  $j$ ;  $X_{ij}$ : variable de decisión binaria de valor 1 si se corta el rodal  $i$  en el año  $j$ , y 0 en caso contrario; y  $V_{ij}$ : volumen total (ton) presente en el rodal  $i$  el año  $j$ .

Las restricciones (2) obligan a cumplir la demanda anual de volumen de madera de la planta; el grupo de inecuaciones (3) corresponde a las restricciones de singularidad e indican que un rodal sólo puede ser cortado en un año del horizonte de planificación; y las restricciones (4) señalan que las variables de decisión del modelo son binarias.

Los datos utilizados (Tabla I) y los supuestos establecidos en la formulación del problema permitieron construir un escenario bajo el cual el modelo puede ser resuelto mediante programación entera binaria y técnicas heurísticas. Para obtener la solución óptima se utilizó el algoritmo de ramificación y acotamiento para programación entera provisto en SAS OR (SAS, 2004).

#### Recocido Simulado

La técnica de Recocido Simulado (RS) es un procedimiento iterativo que parte de una solución inicial que puede ser obtenida en forma aleatoria y hace una búsqueda local para generar, también de manera aleatoria, una solución vecina. La calidad de ambas soluciones es medida de acuerdo a una función de evaluación relacionada con la función objetivo, si la solución vecina es mejor que la solución inicial se acepta como nueva solución y se realiza nuevamente el proceso de búsqueda y evaluación de una nueva solución vecina, en caso contrario, el método puede aceptar la solución vecina como nueva solución dependiendo de una función de probabilidad (P), la cual está dada por:

$$P(\Delta E) = e^{-(\Delta E/t)} \quad (5)$$

donde  $\Delta E$ : diferencia entre el valor de la función de evaluación de la solución vecina y el valor de la función de evaluación de la solución inicial, y  $t$ : parámetro de control llamado temperatura. Este parámetro va cambiando durante la ejecución del algoritmo; comienza con un valor alto haciendo que  $P(\Delta E)$  sea cercana a 1, lo que significa que una solución de peor calidad tiene una alta probabilidad de ser aceptada; la aceptación de una solución con un valor objetivo menos favorecedor permite salir de un posible mínimo local y explorar otras áreas del espacio de soluciones (Dowland y Díaz, 2001; Dréo *et al.*, 2006). A medida que transcurren las iteraciones del algoritmo la temperatura va disminuyendo y con ella disminuye también la probabilidad de aceptar una solución peor. Para cada nivel de temperatura un número de nuevas soluciones debe ser ensayado antes de que la temperatura sea reducida. El proceso de búsqueda se repite hasta que el algoritmo ya no es capaz de generar soluciones de mejor calidad, cuando la temperatura se acerca a cero o cuando se ha alcanzado un número máximo de iteraciones.

#### Búsqueda Tabú

Al igual que el Recocido Simulado, la técnica Búsqueda Tabú (BT)

está basada en la exploración de soluciones vecinas evitando óptimos locales, pero trabaja de manera determinística tratando de emular los procesos de memoria del ser humano (Reeves, 1996). El método memoriza las soluciones que han sido examinadas recientemente, denominándolas puntos tabú (prohibidos) y evita que estos puntos sean nuevamente considerados al seleccionar una próxima solución. El objetivo es conducir la búsqueda a nuevas zonas del espacio de soluciones e impedir búsquedas repetidas en áreas ya exploradas.

El algoritmo más sencillo de BT comienza con la selección de una solución de partida  $X$  y la inicialización de una estructura de memoria, llamada memoria de corto plazo o lista tabú, luego continúa un procedimiento iterativo de búsqueda de una mejor solución. En cada iteración se define un conjunto de soluciones vecinas de  $X$ , el cual excluye aquellas soluciones que se han visitado recientemente y que se encuentran almacenadas en la estructura de memoria. La BT mide la calidad de todas las soluciones vecinas que pertenecen al conjunto utilizando una función de evaluación relacionada a la función objetivo del problema, y selecciona la mejor de ellas para que sea la nueva solución  $X$ , aún cuando esta solución desmejore el valor objetivo en relación a otras soluciones encontradas en iteraciones previas. Al finalizar cada iteración la solución  $X$  se almacena en la memoria o lista tabú. El algoritmo culmina cuando se satisface un criterio de parada, el cual pudiera ser que se ha superado un número máximo de iteraciones, que la solución obtenida es cercana a la solución óptima esperada o ha pasado un cierto número de iteraciones sin que haya habido una mejora en la función objetivo (Hertz *et al.*, 1995). Al final el algoritmo debe mostrar la mejor solución encontrada durante todo el proceso. Versiones mejoradas de la BT incluyen una estructura de memoria de largo plazo, estrategias de intensificación de la búsqueda en un área promisoría del espacio de soluciones y diversificación hacia otras zonas no exploradas.

#### Algoritmos Genéticos

Los Algoritmos Genéticos (AG) son una técnica de búsqueda aleatoria desarrollada por Holland (1975) para hallar soluciones óptimas globales en un espacio de búsqueda multidimensional. Están inspirados en la teoría de evolución de las especies de Darwin, y se basan en la mecánica de selección natural y en principios de genética (Goldberg, 1989). Los AG evalúan simultáneamente un conjunto de soluciones al cual se le denomina población, una solución es considerada un individuo. A cada individuo se le asigna un valor que mide la calidad de esa solución, lo que en la terminolo-

gía de los AG se denota como grado de adaptación (*fitness*). En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos y poder sobrevivir. La función de adaptación que se utiliza en un AG

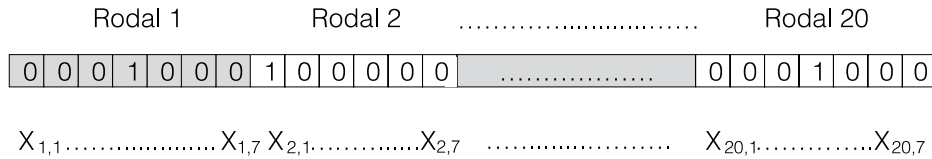


Figura 1. Representación de una solución del problema.

está relacionada con la función objetivo del problema que se desea resolver. Cuanto mayor sea la adaptación de un individuo al problema o dicho de otra manera, cuanto mejor sea una solución, mayor será la probabilidad de que sea seleccionado para reproducirse, cruzando su material genético con otro individuo. Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas de las características de sus padres. Además, los individuos descendientes que surgen del cruzamiento pueden mutar de acuerdo a cierta probabilidad, es decir, cambiar algunas de sus características. Los procesos de cruzamiento y mutación dan lugar a una nueva población, la cual reemplaza a la anterior y verifica la interesante propiedad de contener mayor proporción de buenas características en comparación con la población anterior (Cabezas, 2002).

Este procedimiento se repite un determinado número de iteraciones (generaciones), y después de varios ciclos de evolución la población deberá contener individuos más aptos y al final del proceso, si el AG ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

*Implementación de los algoritmos heurísticos*

Se diseñaron tres algoritmos, cada uno enfocado en resolver el problema de planificación del aprovechamiento forestal utilizando una de las heurísticas expuestas. La implementación se realizó en Visual Basic 6.0 y las pruebas se llevaron a cabo en un computador con procesador modelo Intel(R) Core (TM) 2 Quad de 2.4GHz, 3GB de memoria RAM de y disco duro de 465GB. En el desarrollo de los algoritmos heurísticos se utilizaron las siguientes estrategias de diseño:

*Representación de la solución.* Se hizo mediante un vector de variables binarias de tamaño igual al número de rodales multiplicado por periodo de planificación (20x7= 140). En cada elemento del vector se almacena

una variable binaria  $X_{ij}$  que indica si el rodal  $i$  se corta o no en el año  $j$ . En la Figura 1 se muestra una representación gráfica, puede observarse que las 7 variables (correspondientes a cada año del período de planificación) relacionadas con un mismo rodal son

consecutivas. Esta representación vectorial se usó en los tres algoritmos.

*Criterio de parada.* Para poder realizar comparaciones entre los métodos heurísticos, los tres algoritmos utilizaron el mismo criterio de finalización, el cual fue un número máximo de iteraciones.

*Solución inicial.* En los algoritmos de RS y BT la solución inicial se generó de manera aleatoria: para cada rodal se determina si éste será aprovechado, lo cual ocurre con una probabilidad igual a 0,95. En ese caso se elige al azar el año en que se cortará (todos los años en que se puede cortar un rodal tienen igual probabilidad) y se le asigna a la variable binaria correspondiente el valor 1 y a las demás variables relacionadas a ese rodal se le asigna 0; si un rodal no es aprovechado todas sus variables son iguales a 0. En el caso de AG, es necesario crear una población inicial de N individuos; para ello cada individuo (solución) de la población es generado de forma aleatoria de acuerdo al procedimiento anterior.

*Generación y evaluación de soluciones vecinas.* La forma de generar soluciones vecinas y medir su calidad se hizo de la misma manera en los algoritmos de RS y BT, considerando que una solución vecina de una solución  $X$  será igual a  $X$  excepto por la asignación del año de corta de un rodal. El procedimiento para crear una solución vecina es elegir aleatoriamente un rodal  $i$  y un año  $j$ , si la variable  $X_{ij}$  vale 1 su valor se cambia a 0 en la solución vecina (no se aprovecha el rodal), si  $X_{ij}$  es igual a 0 se le asigna 1 en la solución vecina y a todas las demás variables relacionadas al rodal  $i$  se les da el valor 0. En relación a la manera de medir la calidad de una solución en los algoritmos RS y BT, se utilizó una función de evaluación que incluye la función objetivo original (Ec. 1) y una penalización por la violación total (medida en toneladas) de las restricciones de demanda. De esta manera, las soluciones que violan las restricciones son penalizadas y tienen un peor valor de la función de evaluación que aquellas soluciones que satisfacen las restriccio-

nes. La función de evaluación está dada por la ecuación

$$f_p(\mathbf{X}) = f(\mathbf{X}) + P \times VT(\mathbf{X}) \quad (6)$$

donde  $f(\mathbf{X})$ : función objetivo del problema original,  $P$ : penalización (USD/ton) que se impone a la violación de las restricciones de demanda, y  $VT(\mathbf{X})$ : violación total (ton) de las restricciones de demanda para una solución  $\mathbf{X}$ , la cual se calcula según la Ec (7).

$$VT(\mathbf{X}) = \sum_{k=1}^7 (20000 - g_k) \times I_k \quad (7)$$

donde  $g_k$ : valor de la restricción de demanda para el año  $k$  evaluada en  $\mathbf{X}$ , o lo que es igual, el volumen aprovechado en el año  $k$  si se aplica el plan de cortas (solución)  $\mathbf{X}$ ;  $I_k$ : variable indicadora que vale 1 si en el año  $k$  se viola la restricción de demanda y en otro caso vale 0.

*Curva de enfriamiento y parámetros del algoritmo de Recocido Simulado.* Se utilizó la forma exponencial o geométrica para la curva que determina la velocidad de disminución de la temperatura a medida que avanzan las iteraciones del algoritmo. La ecuación es

$$t_{k+1} = \alpha t_k \quad (8)$$

donde  $t_{k+1}$ : temperatura en la iteración  $k+1$ ,  $t_k$ : temperatura en la iteración  $k$ , y  $\alpha$ : constante cercana a 1, escogida por lo general en el intervalo de 0,9 a 0,99.

Los parámetros necesarios para ejecutar el algoritmo de RS son temperatura inicial, tasa de disminución de la temperatura ( $\alpha$ ) y número de iteraciones en cada nivel de temperatura ( $n_{it}$ ). Todos fueron definidos mediante experimentación.

*Estrategias utilizadas en el algoritmo de Búsqueda Tabú.* Hay varios aspectos que deben definirse para implementar un algoritmo de BT, tales como la permanencia tabú (número de iteraciones que una solución permanece en condición de tabú), la estructura de la memoria de corto plazo y los criterios de aspiración que permiten obviar la condición de tabú para algunas soluciones. En este trabajo se consideró un valor fijo para la permanencia tabú, el cual fue definido mediante experimentación. En cuanto a la estructura de la memoria de corto plazo o lista tabú, corresponde a un vector de tamaño igual al número de rodales (20), y el valor guardado en la posición  $i$  del vector es el número de iteraciones durante las cuales no puede modificarse el año de cosecha del rodal  $i$ , por ser un movimiento tabú. Se utilizó como criterio de aspiración permitir una solución tabú si su valor objetivo es mejor que el de todas las soluciones hasta ese momento encontradas.

*Función de adaptación, operadores y parámetros del algoritmo genético.* La función utilizada en el AG para medir

el grado de adaptación de un individuo fue

$$FIT(\mathbf{X}) = \text{cota} - f^p(\mathbf{X}) \quad (9)$$

donde  $FIT(\mathbf{X})$ : función de adaptación o *fitness*,  $f^p(\mathbf{X})$ : función derivada de la función objetivo original del problema (Ec. 6), y la cota es una cantidad igual o mayor que el máximo valor que puede tomar  $f^p(\mathbf{X})$ .

Se implementó un AG simple basado en el propuesto por Holland (1975). Los operadores genéticos que se incluyen para crear una nueva población a partir de una existente son selección, cruzamiento y mutación.

La selección: permite elegir los individuos de una población que serán cruzados o que sobrevivirán en la nueva población. La selección se hizo de acuerdo a una probabilidad proporcional a la adaptación y se utilizó el método de la ruleta, procedimientos que se encuentran detallados en textos sobre AG (Goldberg, 1989; Reeves, 2003; Dréo *et al.*, 2006).

Se utilizó cruzamiento de un punto con una probabilidad  $P_c$ . Para preservar la estructura original de una solución y garantizar que de los siete elementos del vector que corresponden a un rodal como máximo uno de ellos tenga el valor 1 (restricciones de singularidad), los puntos de cruzamiento que pueden utilizarse están limitados a aquellas posiciones del vector donde finaliza un rodal (7, 14, 21, ..., 133), entre estos valores se selecciona aleatoriamente el punto de cruzamiento.

Se aplicó mutación simple con una probabilidad  $P_m$ . La mutación se llevó a cabo seleccionando aleatoriamente un rodal  $i$  y un año  $j$  y si el valor de  $X_{ij}$  es igual a 0 se le asigna 1 y a todas las demás variables relacionadas al rodal  $i$  se les asigna 0, en caso contrario, si  $X_{ij}$  vale 1 se cambia su valor a 0.

Los parámetros necesarios para correr el AG son tamaño de la población ( $N$ ), probabilidad de cruzamiento ( $P_c$ ) y probabilidad de mutación ( $P_m$ ), los cuales fueron definidos mediante experimentación.

#### Criterios de evaluación de los algoritmos heurísticos

El desempeño de los algoritmos heurísticos fue evaluado en función del valor objetivo obtenido, el error relativo de este valor con respecto al valor objetivo óptimo encontrado mediante programación entera binaria, la iteración en la que consigue la mejor solución y el tiempo de computación necesario para ejecutar los algoritmos. Para ello se realizaron 100 corridas de cada algoritmo, en cada una

de ellas se registró el valor de estas variables y posteriormente se calcularon sus estadísticas básicas (media, desviación estándar, valor mínimo y máximo). El número de iteraciones realizadas en una corrida de un algoritmo fue igual a 500 para las tres técnicas heurísticas. El error relativo de una solución fue calculado utilizando la Ec. 10, que es análoga a la definición clásica.

$$\text{Error relativo} = \frac{V_{\text{obh}} - V_{\text{opt}}}{V_{\text{opt}}} \times 100\% \quad (10)$$

donde  $V_{\text{opt}}$ : valor objetivo óptimo obtenido mediante programación entera binaria, y  $V_{\text{obh}}$ : valor objetivo obtenido en una corrida de un algoritmo heurístico.

La metodología utilizada para evaluar el comportamiento de las técnicas heurísticas RS, BT y AG se ajusta a las técnicas de validación propuestas por Bettinger *et al.* (2009b). Estos autores definen seis niveles de validación para las técnicas heurísticas en problemas de planificación forestal, comenzando con la ausencia de validación (nivel 1) y finalizando con la comparación de las soluciones heurísticas con la solución óptima obtenida mediante un método de programación matemática (nivel 6). Igualmente indican que dependiendo de las heurísticas utilizadas y de los problemas a ser solucionados pueden ser apropiados uno o más de estos niveles. En el presente trabajo se utiliza validación en los niveles 2 y 6. El nivel 2 concierne al uso de estadísticas básicas para valorar la calidad de las soluciones generadas por las heurísticas, para lo que se requiere de un conjunto de soluciones independientes; este supuesto de independencia se satisface en las heurísticas estocásticas como RS y AG, y en técnicas determinísticas como BT se puede cumplir si se generan soluciones iniciales aleatorias, tal como se hizo en este trabajo. El nivel 6 es el de mayor orden ya que implica la comparación de dos técnicas, una heurística y una exacta, para resolver el mismo problema. En este trabajo, al calcular el error relativo de las soluciones heurísticas respecto a la solución óptima obtenida mediante programación entera se realizó validación de nivel 6.

## Resultados y Discusión

En la Tabla II se presenta la solución óptima obtenida utilizando programación entera binaria y se especifica la secuencia de corta que tiene el costo total de aprovechamiento mínimo, siendo éste de 7832018,3 unidades monetarias. Los valores de los parámetros de los algoritmos heurísticos que fueron obtenidos mediante experimentación y que posteriormente fueron usados en las 100 corridas realizadas de cada algoritmo se muestran en la Tabla III.

La técnica RS registró un valor objetivo promedio menor que las demás técnicas, siendo 3,4% más bajo que el obtenido por el AG y 7,5% menor que el valor objetivo promedio de la BT. Esta última técnica fue la que mostró mayor variabilidad en el valor objetivo y el AG tuvo la menor variabilidad (Tabla IV).

En cuanto al error relativo, debido a su forma de cálculo sigue el mismo patrón del valor objetivo, esto es, su valor medio es menor en RS, seguido por AG y BT. A partir del error relativo medio se puede calcular la eficacia de los métodos heurísticos para encontrar la solución del problema, haciendo la resta entre el 100% y el valor del error relativo promedio. De esta manera el RS tiene una eficacia de 94,7%; el AG de 91,0% y la BT de 86,2%. Estos valores indican que los métodos heurísticos utilizados fueron efectivos al encontrar soluciones próximas a la óptima, destacándose RS y AG con eficacias >90%.

La mejor solución fue encontrada con RS. Esta solución tiene un error relativo del 0,22%, lo que significa que está bastante cerca de la óptima. En la tabla V se presentan las secuencias de corta que corresponden a las mejores soluciones encontradas por cada uno de los métodos heurísticos. En los tres casos estas soluciones tienen un error relativo <7%, por lo que puede decirse que son buenas soluciones del problema. Cabe destacar que estas secuencias de corta son bastante distintas a la secuencia óptima de corta encontrada mediante programación matemática (Tabla II); los rodales que se aprovechan en cada año varían de una técnica a otra y en pocos casos hay coincidencias.

El tiempo necesario para ejecutar 500 iteraciones de los algoritmos heurísticos fue medido en todas las corridas. La técnica más rápida fue BT con un tiempo promedio de 1,2s. El RS tuvo un tiempo promedio de 24,3s (~20 veces mayor que BT) y el AG registró un tiempo de ejecución promedio de 141,3s (~118 veces mayor que BT y 6 veces

TABLA II  
SOLUCIÓN ÓPTIMA OBTENIDA MEDIANTE PROGRAMACIÓN ENTERA BINARIA

Año	Rodales a cortar	Volumen aprovechado (ton)
1	4 y 19	101492
2	1 y 20	90873
3	8 y 18	91378
4	9, 13 y 15	90315
5	5, 10 y 14	90515
6	2, 6, 17	91197
7	3, 11 y 12	90057
Volumen total (ton)		645827
Costo total de aprovechamiento (unidades monetarias)		7832018,3
Tiempo de ejecución (s)		3563

TABLA III  
PARÁMETROS DE LOS ALGORITMOS HEURÍSTICOS

Recocido Simulado	Búsqueda Tabú	Algoritmos Genéticos
Temperatura inicial: $5 \times 10^{11}$	Permanencia tabú: 15	Tamaño de la población (N): 1000
Iteraciones del algoritmo: 500	Iteraciones del algoritmo: 500	Iteraciones del algoritmo: 500
Iteraciones en cada nivel de temperatura ( $n_n$ ): 300		Probabilidad de cruzamiento ( $P_c$ ): 1
Tasa de disminución de la temperatura ( $\alpha$ ): 0,9		Probabilidad de mutación ( $P_m$ ): 0,05

TABLA V  
MEJOR SOLUCIÓN OBTENIDA POR LOS ALGORITMOS HEURÍSTICOS

Año	RS		BT		AG	
	Rodales a cortar	Volumen aprovechado (ton)	Rodales a cortar	Volumen aprovechado (ton)	Rodales a cortar	Volumen aprovechado (ton)
1	4, 19	101492	5, 13, 18	98536	2, 5, 19	92309
2	14, 18	90681	1, 2, 4	95533	1, 18	90873
3	8, 20	91378	8, 19	93580	8, 13, 15	91812
4	5, 10, 15	90303	2, 14, 20	118665	14, 20	104220
5	1, 6, 13	90908	3, 9, 10, 11	90996	4, 10, 12, 16	100980
6	2, 3, 9	90144	12, 15, 16	90788	3, 6, 7	91656
7	11, 12, 17	90998	6, 7, 17	99927	9, 11, 17	95966
Volumen total		645905		688025		667815
Costo total de aprovechamiento (unidades monetarias)		7849573,9		8238053,5		8315115,4
Error relativo (%)		0,22		5,18		6,17

RS: Recocido Simulado, BT: Búsqueda Tabú, AG: Algoritmos Genéticos.

mayor que RS). En los AG el tiempo de computación depende en gran parte del tamaño de la población. En este estudio se utilizó una población de 1000 individuos, por lo que en cada corrida el algoritmo debe evaluar 1000 soluciones. Por esta razón el

tiempo de ejecución se incrementa considerablemente en relación a las otras técnicas.

El tiempo de computación que requirió el programa SAS OR para obtener la solución óptima mediante programación entera fue de casi una hora (3563s), ~25 veces mayor que el tiempo de cómputo promedio del algoritmo heurístico más lento. Estos resultados corroboran una de las ventajas que se le atribuyen a los métodos heurísticos, el poder encontrar una solución a un problema en forma mucho más rápida que los métodos de programación matemática clásicos (Reeves, 1996; Voß, 2001; Dréo *et al.*, 2006). En situaciones reales es importante poder obtener soluciones en un tiempo oportuno para poder tomar decisiones de planificación; en problemas complejos los métodos de programación matemática presentan limitaciones al respecto, incluso en muchos casos son incapaces de obtener una solución en un tiempo razonable (Zanackis y Evans, 1981). En este caso particular pudiera parecer poco relevante la diferencia en tiempos de ejecución, puesto que una hora para encontrar la solución óptima pudiera ser

desviación estándar, máximo y mínimo muestran una alta variabilidad en estos resultados. El RS en promedio no mejoró la solución después de la iteración 125; sin embargo, en esta técnica la variabilidad es mucho menor. La observación de esta variable permite determinar un número de iteraciones máximo más adecuado para los algoritmos. En este caso podría disminuirse el valor de este parámetro para RS y BT, lo que acortaría el tiempo de ejecución. Otra alternativa para reducir el tiempo de cómputo puede ser el uso de un criterio de parada diferente, tal como finalizar el algoritmo después que hayan ocurrido cierto número de iteraciones sin conseguir mejora. No obstante, en este trabajo se fijó un número de iteraciones igual para todos los algoritmos con la finalidad de realizar comparaciones entre ellos.

Considerando los resultados expuestos puede decirse que RS fue la técnica heurística más eficiente al resolver el problema planteado, ya que obtuvo un valor objetivo promedio menor, fue capaz de conseguir la solución más cercana a la óptima y aunque no tuvo el tiempo de computación más bajo, éste fue bastante razonable y mucho menor que el tiempo requerido por el método de programación entera binaria. Los AG también tuvieron un buen desempeño al resolver el problema, por lo cual se considera una buena alternativa de solución en este caso particular. La BT es la heurística que requirió un tiempo de cómputo menor pero fue la menos eficaz en encontrar una solu-

razonable, pero es necesario considerar que el tiempo de computación requerido por la programación entera binaria puede crecer considerablemente cuando se incrementa el número de rodiles o se cambian algunos datos como la longitud del período de planificación, los valores del incremento medio anual o la edad de los rodiles, o que en algunas situaciones esta técnica no es capaz encontrar la solución óptima. En estas situaciones, la disminución en el tiempo de ejecución que registran las técnicas heurísticas y su capacidad para encontrar siempre una solución son de gran importancia.

En las diferentes corridas también se registró la iteración en la que un algoritmo obtiene su mejor solución. Después de esa iteración el proceso de búsqueda no mejora el valor objetivo de las soluciones encontradas. Se observó que la BT, en promedio, encontró la solución en la iteración 24 y el AG en la iteración 224. Los valores de la

TABLA IV  
ESTADÍSTICAS BÁSICAS DE LAS VARIABLES DE DESEMPEÑO EVALUADAS

	RS	BT	AG
Valor Objetivo (unidades monetarias)			
$\bar{X}$	8246495,3	8911026,4	8536323,5
DE	408056,7	900412,0	184519,6
Min	7849573,9	8238053,5	8315115,4
Max	10071125,3	12073707,9	8921306,0
Error relativo (%)			
$\bar{X}$	5,29	13,77	8,99
DE	5,21	11,50	2,36
Min	0,22	5,18	6,17
Max	28,59	54,16	13,91
Iteración donde se encuentra la mejor solución			
$\bar{X}$	125	24	224
DE	8	66	95
Min	105	3	44
Max	139	496	462
Tiempo de ejecución (s)			
$\bar{X}$	24,26	1,34	141,32
DE	0,45	0,02	1,23
Min	23,68	1,32	139,38
Max	25,55	1,42	145,37

$\bar{X}$ : media, DE: desviación estándar, Min: valor mínimo, Max: valor máximo, s: segundos, RS: Recocido Simulado, BT: Búsqueda Tabú, AG: Algoritmos Genéticos.

ción; en tal sentido cabe señalar que se utilizó la versión básica de esta técnica y el desempeño del algoritmo podría mejorarse incluyendo estrategias avanzadas de diseño tales como diversificación, intensificación y memoria a largo plazo. También pudieran ensayarse otros mecanismos para obtener soluciones vecinas.

Es difícil comparar los resultados de este estudio con los reportados por otros autores. Tal como lo señalan Pukkala y Kurttila (2005), un problema para realizar comparaciones es que el desempeño de los métodos heurísticos depende del tipo de problema, de las estrategias de diseño utilizadas en los algoritmos y de los valores de los parámetros utilizados, por lo que las conclusiones pueden variar considerablemente de un caso a otro. Sin embargo, es posible señalar que el comportamiento de las heurísticas utilizadas concuerda con el trabajo de Bettinger *et al.* (2002) quienes categorizaron un conjunto de heurísticas en tres clases: muy buenas, adecuadas y menos que adecuadas. De acuerdo al valor de la función objetivo que son capaces de obtener, estos autores sitúan la técnica de RS en el primer grupo, y a BT y AG en el segundo. En relación al tiempo de computación, los resultados acá obtenidos se ajustan a la clasificación propuesta por Pukkala y Kurttila (2005), en la que la BT se encuentra entre las técnicas de mediana rapidez y tanto RS como AG pertenecen al grupo de las heurísticas más lentas. Otro tipo de heurísticas más rápidas pero menos eficaces en encontrar valores objetivo cercanos al óptimo y que se encuentran en la clasificación de estos autores, tales como el ascenso aleatorio y la técnica Hero, no fueron empleadas en este estudio.

Además de la posibilidad de encontrar soluciones cercanas al óptimo en un tiempo de computación menor al requerido por una técnica de programación matemática y su capacidad de hallar soluciones en casos donde los métodos exactos no pueden, otra ventaja de los métodos heurísticos es que pueden brindar diferentes soluciones de un problema al realizar varias corridas. Por su parte, los métodos de programación matemática sólo pueden encontrar una única solución, que aunque es la óptima pudiera tener en algunos casos dificultades para ser llevada a la práctica. Es más útil para el planificador forestal tener varias alternativas de solución y de esta forma elegir aquella que además de tener un buen valor de la función que se está optimizando, tenga mayor factibilidad de implementación.

A pesar de las ventajas que ofrecen las técnicas heurísticas, en la práctica han tenido limitaciones de uso debido al tiempo que debe invertirse para desarrollar los algoritmos y a la necesidad de contar con personal capacitado en el área. Las heurísticas comúnmente son diseñadas

para resolver un problema particular y, como resultado, el software desarrollado para un caso específico es difícil de utilizar en otro problema pues cada uno tiene características diferentes; de allí que no existan paquetes de computación estándar que puedan usarse de manera general en planificación forestal. No obstante, tal como lo señalan Baskent y Keles (2005) es importante considerar que una vez desarrollado un algoritmo heurístico para un problema es posible generar en forma rápida muy buenas soluciones para diferentes instancias del mismo, incluso aquellas que no pueden resolverse por técnicas de programación matemática. Esto puede compensar el esfuerzo requerido.

## Conclusiones

Los métodos heurísticos son una excelente alternativa en la solución de problemas de planificación forestal, puesto que permiten manejar la complejidad implícita en este tipo de problemas y solventar las limitaciones que tienen los métodos de programación matemática en muchas situaciones reales. Las heurísticas utilizadas en este trabajo mostraron efectividad para resolver el problema de planificación del aprovechamiento forestal planteado. Destacaron la técnica de Recocido Simulado y los Algoritmos Genéticos con eficacias >90%, mientras que la Búsqueda Tabú presentó una eficacia de 86,2%. Los resultados sugieren que los algoritmos heurísticos desarrollados pueden ser utilizados en aquellas instancias del problema que no pueden resolverse por programación entera binaria, siendo de esta manera métodos de solución más robustos para el problema tratado.

El tiempo de computación promedio de los algoritmos heurísticos fue menor en todos los casos que el tiempo de ejecución requerido por el método de programación entera binaria. Este es un factor clave a considerar al elegir un método de solución para un problema de planificación forestal. Los resultados obtenidos muestran que las heurísticas presentan ventajas que pueden ser aprovechadas para resolver problemas de planificación forestal satisfactoriamente, tanto en tiempo de cómputo como en calidad de las soluciones obtenidas.

## REFERENCIAS

- Baskent EZ, Keles S (2005) Spatial forest planning: A review. *Ecol. Modell.* 88: 145-173.
- Bettinger P, Chung W (2004) The key literature of, and trends in, forest-level management planning in North America, 1950-2001. *Int. For. Rev.* 6: 40-50.
- Bettinger P, Sessions J, Boston K (1997) Using Tabu search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecol. Modell.* 94: 111-123.
- Bettinger P, Boston K, Kim YH, Zhua J (2007) Landscape-level optimization using tabu search

- and stand density-related forest management prescriptions. *Eur. J. Oper. Res.* 176: 1265-1282.
- Bettinger P, Boston K, Siry JP, Grebner DL (2009a) *Forest Management and Planning*. Academic Press. San Diego, CA, EEUU. 331 pp.
- Bettinger P, Sessions J, Boston K (2009b) A review of the status and use of validation procedures for heuristics used in forest planning. *Int. J. Math. Comput. For. Nat. Resource Sci.* 1: 26-37.
- Boston K, Bettinger P (1999) An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems. *For. Sci.* 45: 292-301.
- Brumelle S, Granot D, Helme M, Vertinsky I (1998) A tabu search algorithm for finding good forest harvest schedules satisfying green-up constraints. *Eur. J. Oper. Res.* 106: 408-424.
- Cabezas CA (2002) Algoritmos Genéticos, una opción para la optimización de funciones. *Tecnol. Quím.* 23: 65-69.
- Chiari R, Carrero O, Jerez M, Quintero MA, Stock J (2008) Modelo preliminar para la planificación del aprovechamiento en plantaciones forestales industriales en Venezuela. *Interiencia* 33: 802-809.
- Crowe KA, Nelson JD (2005) An evaluation of the simulated annealing algorithm for solving the area-restricted harvest-scheduling model against optimal benchmarks. *Can. J. For. Res.* 35: 2500-2509.
- Dahlin B, Sallnas O (1993) Harvest scheduling under adjacency constraints -A case study from the Swedish sub-alpine region. *Scand. J. For. Res.* 8: 281-290.
- Dowland KA, Díaz BA (2001) Diseño de heurísticas y fundamentos del recocido simulado. *Rev. Iberoam. Intel. Artif.* 20: 34-52.
- Dréo J, Pétrowski A, Siarry P, Taillard E (2006) *Metaheuristics for Hard Optimization*. Springer. Berlin, Alemania. 369 pp.
- Ducheyne EI, De Wulf RR, De Baets B (2004) Single versus multiple objective genetic algorithms for solving the even-flow forest management problem. *For. Ecol. Manag.* 201: 259-273.
- Falcão AO, Borges JG (2001) Designing an evolution program for solving integer forest management scheduling models: an application in Portugal. *Forest Sci.* 47: 158-168.
- Falcão AO, Borges JG (2002) Combining random and systematic search heuristic procedures for solving spatially constrained forest management scheduling models. *For. Sci.* 48: 608-621.
- Goldberg DE (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley. Nueva York, EEUU. 412 pp.
- Heinonen T, Pukkala T (2004) A comparison of one- and two-compartment neighbourhoods in heuristic search with spatial forest management goals. *Silva Fenn.* 38: 319-332.
- Hertz A., Taillard E, De Werra D (1995) A tutorial on tabu search. En *Proc. Giornate di Lavoro AIRO'95, Enterprise Systems: Management of Technological and Organizational Changes*. pp. 13-24.
- Holland JH (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Harbor, MI, EEUU. 211 pp.
- Liu G, Han S, Zhao X, Nelson JD, Wang H, Wang W (2006) Optimisation algorithms for spatially constrained forest planning. *Ecol. Modell.* 194: 421-428.
- Lockwood C, Moore T (1993) Harvest scheduling with spatial constraints: a simulated annealing approach. *Can. J. For. Res.* 23: 468-478.
- Lu F, Eriksson LO (2000) Formation of harvest units with genetic algorithms. *Ecol. Manag.* 130: 57-67.

- McDill ME, Braze J (2001) Using branch and bound algorithm to solve forest planning problems with adjacency constraints. *Forest Sci.* 47: 403-418.
- Murray AT, Weintraub A (2002) Scale and unit specification influences in harvest scheduling with maximum area restrictions. *For. Sci.* 48: 779-789.
- Palahi M, Pukkala T, Pascual L, Trasobares A (2004) Examining alternative landscape metrics in ecological forest planning: a case for capercaillie in Catalonia. *Invest. Agr.: Sist. Rec. For.* 13: 527-538.
- Pukkala T, Kurttila M (2005) Examining the performance of six heuristic search techniques in different forest planning problems. *Silva Fenn.* 39: 67-80.
- Reeves CR (1996) Modern heuristic techniques. En Rayward-Smith VJ, Osman IH, Reeves CR, Smith GD (Eds.) *Modern Heuristic Search Methods*. Wiley, Chichester, RU. pp 1-25.
- Reeves CR (2003) Genetic Algorithms. En Glover F, Kochenberger GA (Eds.) *Handbook of Metaheuristics*. Kluwer. Boston, MA, EEUU. pp. 37-54.
- Richards EW, Gunn E (2000) A Model and tabu search method to optimize stand harvest and road construction schedules. *For. Sci.* 46: 188-203.
- Richards EW, Gunn E (2003) Tabu search design for difficult forest management optimization problems. *Can. J. Forest Res.* 33: 1126-1133.
- Rodrigues FL, Leite HG, Santos HN, De Souza AL (2003) Solução de problemas de planejamento florestal com restrições de inteireza utilizando Busca Tabu. *Arvore* 27: 701-713.
- Rodrigues FL, Leite HG, Santos HN, De Souza AL, Silva GF (2004a) Metaheurística algoritmo genético para solução de problemas de planejamento florestal com restrições de integridade. *Arvore* 28: 233-245.
- Rodrigues FL, Leite HG, Santos HN, De Souza AL, Silva GF (2004b) Metaheurística simulated annealing para solução de problemas de planejamento florestal com restrições de integridade. *Arvore* 28: 247-256.
- SAS (2004) *SAS/OR 9.1 User's Guide*. SAS Institute Inc. Cary, NC, EEUU. 168 pp.
- Shan Y, Bettinger P, Cieszewski CJ, Li RT (2009) Trends in spatial forest planning. *Int. J. Math. Comput. For. Nat. Resource Sci.* 1: 86-112.
- Tarp P, Helles F (1997) Spatial optimization by simulated annealing and linear programming. *Scand. J. Forest Res.* 12: 390-402.
- Thompson MP, Hamann JD, Sessions J (2009) Selection and penalty strategies for genetic algorithms designed to solve spatial forest planning problems. *Int. J. For. Res.* 2009: 1-14.
- Voß S (2001) Meta-heuristics: The state of the art. En Nareyek A (Eds.) *Local Search for Planning and Scheduling*. Springer, Berlin, Alemania. pp 1-23.
- Zanackis SH, Evans JR (1981) Heuristic "optimization": Why, when, and how to use it. *Interfaces* 11: 84-91.
- Zhu J, Bettinger P (2008) Assessment of three heuristics for developing large-scale spatial forest harvest scheduling plans. *J. Appl. Sci.* 8: 4113-4120.

## EVALUATION OF THREE HEURISTIC ALGORITHMS TO SOLVE A FOREST HARVEST PLANNING MODEL IN INDUSTRIAL FOREST PLANTATIONS

María Alejandra Quintero M, Mauricio Jerez and Magdiel Ablan

### SUMMARY

Three heuristic algorithms, based on Simulated Annealing, Tabu Search and Genetic Algorithms, were developed in order to solve a problem in forest harvest planning consisting on a cutting sequence for 20 stands, in a planning horizon of seven years, such as to minimize harvest costs and meet the annual demand of wood for a pulpwood mill. With the aim of evaluating the performance of the heuristic techniques and making comparisons, the problem was also solved through a mathematical programming technique (integer binary programming) that permitted to find out the optimal solution for the problem. One hundred runs of each heuristic algorithm were executed and descriptive statistics were calculated for the variables: objective value, relative error with respect to the optimal solution, iteration

in which the best solution is obtained, and computing time. The three techniques showed a good performance in solving the problem; the efficiency to find the solution was 94.7% with Simulated Annealing, 91.0% with Genetic Algorithms and 86.2% with Tabu Search. Every heuristic algorithm employed computing times considerably lower than that required for binary integer programming. Due to its capacity to find good solutions in reasonable computing times, heuristic techniques represent a feasible approach for solving forest management planning problems, specially those that are difficult to solve with classical mathematical programming techniques due to their complexity and very long computing times.

## AVALIACÃO DE TRÊS TÉCNICAS HEURÍSTICAS PARA RESOLVER UM MODELO DE PLANIFICAÇÃO DO APROVEITAMENTO EM PLANTACÕES FLORESTAIS INDUSTRIAIS

María Alejandra Quintero M, Mauricio Jerez e Magdiel Ablan

### RESUMO

Desenvolveram-se três algoritmos heurísticos baseados nas técnicas de Recozimento Simulado, Busca Tabú e Algoritmos Genéticos, para resolver um problema de planificação do aproveitamento florestal que consiste em obter uma sequência de talas para 20 rodais, em um período de planificação de sete anos, de forma que os custos de aproveitamento sejam mínimos e se satisfaça a demanda anual de madeira de uma fábrica de polpa para papel. A fim de avaliar o desempenho das técnicas heurísticas utilizadas e fazer comparações, o problema também foi resolvido utilizando uma técnica de programação matemática (programação inteira binária) que permitiu obter a solução ótima. Realizaram-se 100 corridas dos algoritmos heurísticos e se calcularam estatísticas básicas para as variáveis: valor objetivo, erro relativo à solução ótima, iteração na qual é atingi-

da a melhor solução, e tempo de computação utilizado. As três técnicas mostraram bom desempenho para resolver o problema definido; sua efetividade para encontrar a solução foi 94,7% em Recozimento Simulado, 91,0% para Algoritmos Genéticos e 86,2% em Busca Tabú. Todos os algoritmos heurísticos empregaram tempos de computação consideravelmente menores que o requerido pela programação inteira binária. Devido a sua capacidade para encontrar boas soluções em tempos de computação razoáveis, as técnicas heurísticas representam um enfoque que pode ser utilizado na resolução de problemas de planificação florestal, especialmente aqueles difíceis de resolver mediante técnicas clássicas de programação matemática por ser complexos ou requerer elevados tempos de processamento.