

# Métodos heurísticos en la planificación del manejo forestal: Un ejemplo de aplicación

*Heuristics methods in forest management planning: An example*

MARÍA ALEJANDRA QUINTERO M.<sup>1</sup>,  
MAURICIO JEREZ RICO<sup>1,2</sup>  
y MAGDIEL ABLAN<sup>3</sup>

Universidad de Los Andes,

- 1 Facultad de Ciencias Forestales y Ambientales, Departamento de Botánica y Ciencias Básicas, Mérida, Venezuela, E-mail: mariaq@ula.ve
- 2 Facultad de Ciencias Forestales y Ambientales, Centro de Estudios Forestales y Ambientales de Postgrado, Mérida, Venezuela, E-mail: jerez@ula.ve
- 3 Facultad de Ingeniería, Centro de Simulación y Modelos, Mérida, Venezuela, E-mail: mablan@ula.ve

Recibido: 15-02-10 / Aceptado: 20-05-10

## Resumen

En los últimos años se ha venido incrementando el uso de métodos heurísticos para resolver problemas complejos de planificación y manejo forestal que no pueden ser abordados por las técnicas clásicas de programación matemática, tales como la programación lineal. Los métodos heurísticos son técnicas alternativas para tratar problemas de optimización difíciles, ya que son capaces de encontrar buenas soluciones en un tiempo de computación razonable. Sin embargo, todavía no se ha prestado suficiente atención a estas técnicas en el campo forestal venezolano. El objetivo de este trabajo fue ilustrar el potencial de dos técnicas heurísticas: Recocido Simulado y Búsqueda Tabú, mediante su aplicación a un ejemplo sencillo de planificación de cosechas en una plantación forestal. Se desarrollaron dos algoritmos heurísticos basados en estas técnicas y los resultados fueron comparados con la solución óptima obtenida mediante Programación Lineal Entera Binaria. El comportamiento de los métodos heurísticos fue evaluado en términos del valor objetivo obtenido, el error relativo, la rapidez para encontrar una solución y el tiempo de ejecución de los algoritmos. Las dos técnicas heurísticas mostraron un buen desempeño para el problema particular planteado. La Búsqueda Tabú presentó una eficacia del 100% mientras que en el algoritmo de Recocido Simulado la eficacia fue del 99,96%. Los tiempos de ejecución de ambos algoritmos fueron considerablemente menores que el tiempo utilizado por el algoritmo de optimización de programación lineal entera.

**Palabras clave:** Búsqueda Tabú, Recocido Simulado, planificación de cortas, programación entera binaria.

## Abstract

In recent years, there has been an increased use of heuristics for solving complex problems in forest planning and management that cannot be approached with the classical techniques in mathematical programming such as lineal programming. Heuristics are alternative techniques for dealing with difficult optimization problems because they are capable of finding good solutions in a reasonable computation time. However, so far no much attention has been lent to these techniques in the field of forestry in Venezuela. The objective of this paper was to illustrate the potential of two heuristic techniques: Simulated Annealing and Tabu Search through a simple example of forest harvest planning for a forest plantation. Two heuristic algorithms were developed based on these techniques, and the results were compared with the optimal solution obtained with binary integer linear programming. The performance of the heuristic methods was evaluated in terms of the resulting objective value, the relative error, the speed to find a solution, and the execution time. Both heuristic techniques showed good performance for the particular problem. Tabu Search showed a 100% efficacy whereas Simulated Annealing showed a 99,96%. The time of computation for both algorithms was considerably lower than the time employed by the Integer Lineal Programming algorithm.

**Key words:** Simulated annealing, tabu search, harvest planning, Integer binary programming.

## 1. Introducción

Los planes forestales son descripciones específicas de las acciones que deben llevarse a cabo para alcanzar los objetivos planteados para un bosque, proporcionando una guía para implementar actividades, predecir niveles futuros de cosecha, optimizar el uso de recursos limitados y mantener o

desarrollar áreas de hábitat mientras se logra un balance entre diferentes factores como presupuesto, personal, etc. (Bettinger *et al.*, 2009). Muchas de las situaciones y decisiones que deben abordarse dentro de la planificación forestal, tales como la selección entre diferentes alternativas de manejo, programación de cortas, programación del transporte en vías forestales, entre muchas otras, pue-

den ser modeladas usando técnicas de programación matemática.

Los modelos de programación matemática constan de una función objetivo que se desea maximizar o minimizar tal como ingresos, beneficios, costos o impacto ambiental; sujeto a un conjunto de restricciones que pueden estar relacionadas a requerimientos de capacidad, producción, limitaciones sociales, legislativas y ambientales. La programación matemática proporciona la solución óptima del problema, la cual se obtiene mediante algoritmos contrastados y de propósito general disponibles en los paquetes de optimización.

En el campo de la planificación y manejo forestal este tipo de modelos se ha venido utilizando con éxito desde finales de los años 60. Durante la década de los años 80, técnicas típicas de la programación matemática como la programación lineal y sus variantes (programación entera, programación por metas), comenzaron a reemplazar a los métodos tradicionales de planificación forestal, constituyéndose en herramientas de gran ayuda en los procesos de toma de decisiones. Sin embargo, la programación matemática se ha encontrado con limitaciones para resolver problemas actuales de planificación cada vez más complejos, tales como: problemas de grandes dimensiones en los que la cantidad de variables y el número de posibles soluciones es tan elevado que resolverlo mediante un algoritmo exacto de programación matemática, aún utilizando computadoras con gran capacidad de cálculo y software adecuado, no es posible o puede tomar mucho tiempo; problemas que tratan de optimizar simultáneamente diferentes objetivos, por ejemplo, los objetivos económicos (beneficios, costos) y ecológicos (conservación de la biodiversidad, fijación de carbono); problemas que incluyen relaciones funcionales no lineales que no pueden considerarse con técnicas clásicas de programación matemática sin hacer una simplificación.

Las limitaciones de los métodos de programación matemática para resolver problemas complejos de planificación forestal han propiciado el uso de técnicas heurísticas como métodos alternativos. En contraposición a los algoritmos exactos de programación matemática, los métodos heurísticos se limitan a proporcionar una buena solución del problema, no necesariamente óptima, pero en un tiempo de computación aceptable; en problemas

difíciles de resolver, han demostrado ser más eficientes, encontrando buenas soluciones en forma mucho más rápida. Entre los métodos heurísticos que se han utilizado en el campo de la planificación forestal se encuentran la simulación Monte Carlo (O'Hara *et al.*, 1989; Nelson y Brodie 1990; Clements *et al.*, 1990); recocado simulado (Lockwood y Moore, 1993; Ohman y Ericsson, 2002; Crowe y Nelson, 2005); búsqueda tabú (Bettinger *et al.*, 1997, 2007; Brumelle *et al.*, 1998; Richards y Gunn, 2000, 2003; Díaz *et al.*, 2007); y algoritmos genéticos (Lu y Eriksson, 2000; Falcão y Borges, 2001; Ducheyne *et al.*, 2004; Thompson *et al.*, 2009).

En la literatura no se encuentran aplicaciones de los métodos heurísticos en la planificación del manejo forestal en Venezuela, incluso los modelos basados en técnicas de programación matemática son poco utilizados en el país. En este contexto es importante citar el modelo de programación entera desarrollado por Chiari *et al.* (2008) para la planificación del aprovechamiento forestal.

El objetivo de este artículo es ilustrar el potencial que tienen los métodos heurísticos en la solución de problemas de planificación forestal. Para ello, se planteó como ejemplo una aplicación sencilla de planificación de cortas en plantaciones forestales, cuya solución óptima fue obtenida primeramente mediante una técnica clásica de programación matemática: la programación entera binaria. Se desarrollaron algoritmos basados en dos de las técnicas heurísticas más utilizadas: Recocado Simulado y Búsqueda Tabú, y se examinó su desempeño respecto de la programación entera, en términos de precisión y tiempo de cómputo.

## 2. Materiales y métodos

### 2.1 El problema

Se consideró una situación hipotética en la que se necesita obtener un plan de cortas para una unidad de producción forestal, tal que permita abastecer de madera en forma continua a una planta de producción de pulpa y minimice los costos totales de aprovechamiento. La unidad de producción forestal está conformada por seis rodales de una misma especie (eucalipto), el período de planificación es de cinco años y las necesidades anuales de materia prima de la planta son de 20.000 toneladas.

De cada rodal se conoce su edad (años), área (ha) y la curva de incremento medio anual (IMA), y a partir de estos datos se calculó el volumen disponible de madera en cada rodal para cada año del período de planificación. El turno mínimo se estableció a los 6 años y el turno óptimo a los 7 años para la especie considerada.

El costo total de aprovechamiento en cada año y para cada rodal se calculó con base en tres componentes: costo de cosecha, costo de transportar la madera a 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 consideró constante para todos los rodales, los costos de transporte dependen de la distancia de cada rodal a la planta, y los costos de oportunidad fueron calculados de acuerdo al enfoque de alternativas financieras utilizado por Chiari *et al.* (2008). Debido al carácter ilustrativo del ejemplo, todos los costos se encuentran en unidades monetarias (u.m.) sin especificación de alguna moneda particular. Los datos necesarios para formular el modelo se muestran en los cuadros 1, 2 y 3.

## 2.2 El modelo matemático

El modelo que representa el problema en estudio es el siguiente:

$$\text{Minimizar } Z = \sum_{i=1}^6 \sum_{j=1}^5 C_{ij} * X_{ij} \quad (\text{Ec.1})$$

Sujeto a las siguientes restricciones:

$$\sum_{i=1}^6 V_{ij} * X_{ij} \geq 20000 \quad j=1, 2, \dots, 5 \quad (\text{Ec.2})$$

$$\sum_{j=1}^5 X_{ij} = 1 \quad i=1, 2, \dots, 6 \quad (\text{Ec.3})$$

$$X_{ij} = (0, 1) \quad i=1, 2, \dots, 6 ; j=1, 2, \dots, 5 \quad (\text{Ec.4})$$

Donde:

$C_{ij}$  = costo total de aprovechamiento (costo de cosecha + costo de transporte + costo de oportunidad), si se corta el rodal  $i$  en el año  $j$

$V_{ij}$  = volumen total (toneladas), presente en el rodal  $i$  el año  $j$

$X_{ij}$  = variable de decisión binaria cuyo valor es igual a 1, si se corta el rodal  $i$  en el año  $j$ , y vale 0 en caso contrario

**Cuadro 1.** Datos de los rodales.

Rodal	Edad (años)	Area (ha)	IMA (m³/ha/año)
1	7	245	29
2	4	78	26
3	6	305	18
4	8	154	27
5	5	208	27
6	2	310	26

**Cuadro 2.** Volumen disponible en cada rodal.

Rodal	Volumen (t)				
	año 1	año 2	año 3	año 4	año 5
1	39788	43198	46168	48733	50926
2	0*	0	9734	11356	12330
3	26352	30744	33379	35674	37656
4	25281	27019	28520	29803	30887
5	0	26957	31450	34145	36493
6	0	0	0	0	12896

\* Las celdas con volumen igual a 0, se refieren a rodales que no pueden ser cortados en un año determinado debido a que no han alcanzado la edad mínima de corta.

**Cuadro 3.** Costo total de aprovechamiento

Rodal	Costo total (u.m.)*				
	año 1	año 2	año 3	año 4	año 5
1	1273216	1388498,3	1501958,8	1610544,7	1712374,5
2	0	0	314574	363417,6	396119,9
3	857294	983808	1075210,4	1169903,1	1263884,6
4	973304,6	1052724,1	1128516,7	1199410,1	1264696,5
5	0	1172496	1263375,4	1354704,4	1767886,3
6	0	0	0	0	1048314,8

\* Unidades monetarias

El modelo tiene como función objetivo la minimización de los costos totales de aprovechamiento según la ecuación 1, sujeta a tres grupos de restricciones. El primer grupo (ecuación 2) corresponde al cumplimiento de las necesidades de materia prima de la planta; el segundo grupo (ecuación 3) son las restricciones de singularidad, que se refieren al hecho que un rodal sólo puede ser cortado en un año del período de planificación, la igualdad de estas restricciones indica que al final del período

do de planificación todos los rodales deben haber sido aprovechados; y el tercer grupo (ecuación 4) indica que las variables de decisión del modelo son binarias.

## 2.3 Técnicas de optimización utilizadas

### 2.3.1 Programación entera binaria

Es una técnica de optimización derivada de la programación lineal que se basa en representar un problema utilizando funciones matemáticas lineales y variables de decisión binarias que asumen el valor 0 ó 1. Un modelo de programación binaria está conformado por una función objetivo y una serie de restricciones que limitan la solución; tanto la función objetivo como las restricciones del problema son funciones lineales de variables binarias.

Se han desarrollado varios algoritmos para encontrar la solución óptima de modelos de programación entera o cualquiera de sus variantes (programación entera pura, mixta o binaria). Uno de los más utilizados es el algoritmo de ramificación y acotamiento, el cual se encuentra en muchos de los programas de computación para optimización disponibles en el mercado. El problema con estos algoritmos para programación entera es que a pesar de las capacidades computacionales de hoy en día, presentan serios inconvenientes cuando se trabaja con problemas muy grandes, con cientos de restricciones y variables (Hillier y Lieberman, 2001).

### 2.3.2 Recocido Simulado

La técnica heurística del recocido simulado, en inglés *Simulated Annealing*, está basada en un algoritmo propuesto por Metropolis *et al.* (1953) en el marco de la termodinámica estadística, para simular el proceso de enfriamiento de un material (recocido). Esta heurística comprende una secuencia de iteraciones que modifica de forma aleatoria una solución actual. El método comienza con una solución inicial, realiza una búsqueda local y genera una solución vecina de manera aleatoria, entendiéndose por solución vecina aquella que se encuentra cerca en el espacio de soluciones y que se obtiene cambiando algún(os) atributo(s) de la solución actual. Si la solución vecina es mejor que la solución inicial, en términos de su valor objetivo, el algoritmo la acepta y se reinicia el proceso de búsqueda a partir de esta nueva solución; en caso

contrario, cuando la solución vecina generada no es mejor que la solución inicial, el método puede aceptarla dependiendo de una función de probabilidad, la cual está dada por la ecuación 5:

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

Donde:

$\Delta E$  = diferencia entre el valor objetivo de la solución actual (solución inicial) y el valor objetivo de la solución vecina

$t$  = parámetro del algoritmo llamado temperatura.

Aceptar una solución de menor calidad permite salir de un posible mínimo local y explorar otras áreas del espacio de soluciones. El algoritmo comienza siempre con una temperatura alta, por lo tanto en las primeras iteraciones  $P(\Delta E)$  es cercana a 1, lo que significa que una nueva solución de peor calidad tiene una alta probabilidad de ser aceptada. La probabilidad de aceptar una solución peor va disminuyendo a medida que avanzan las iteraciones del algoritmo y la temperatura decrece. Este proceso se repite  $n_{it}$  veces. Cuando este ciclo iterativo se completa, la temperatura se disminuye y comienza nuevamente el proceso de creación, evaluación y posible aceptación de soluciones vecinas. Cuando la temperatura es lo suficientemente baja o después de haber alcanzado un número máximo de iteraciones, el algoritmo finaliza y debe indicar cuál fue la mejor solución encontrada. El diagrama de flujo correspondiente al algoritmo básico de recocido simulado se presenta en la figura 1.

### 2.3.3 Búsqueda Tabú

Es un método heurístico originalmente propuesto por Glover (1986) para resolver problemas de optimización combinatoria. Esta técnica al igual que el Recocido Simulado se basa en la búsqueda de soluciones vecinas evitando óptimos locales, pero lo hace utilizando el concepto de "memoria" para conducir la búsqueda a nuevas zonas del espacio de soluciones e impedir búsquedas repetidas en áreas ya exploradas. 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 algoritmo más sencillo de Búsqueda Tabú comienza con la selección de una solución de par-

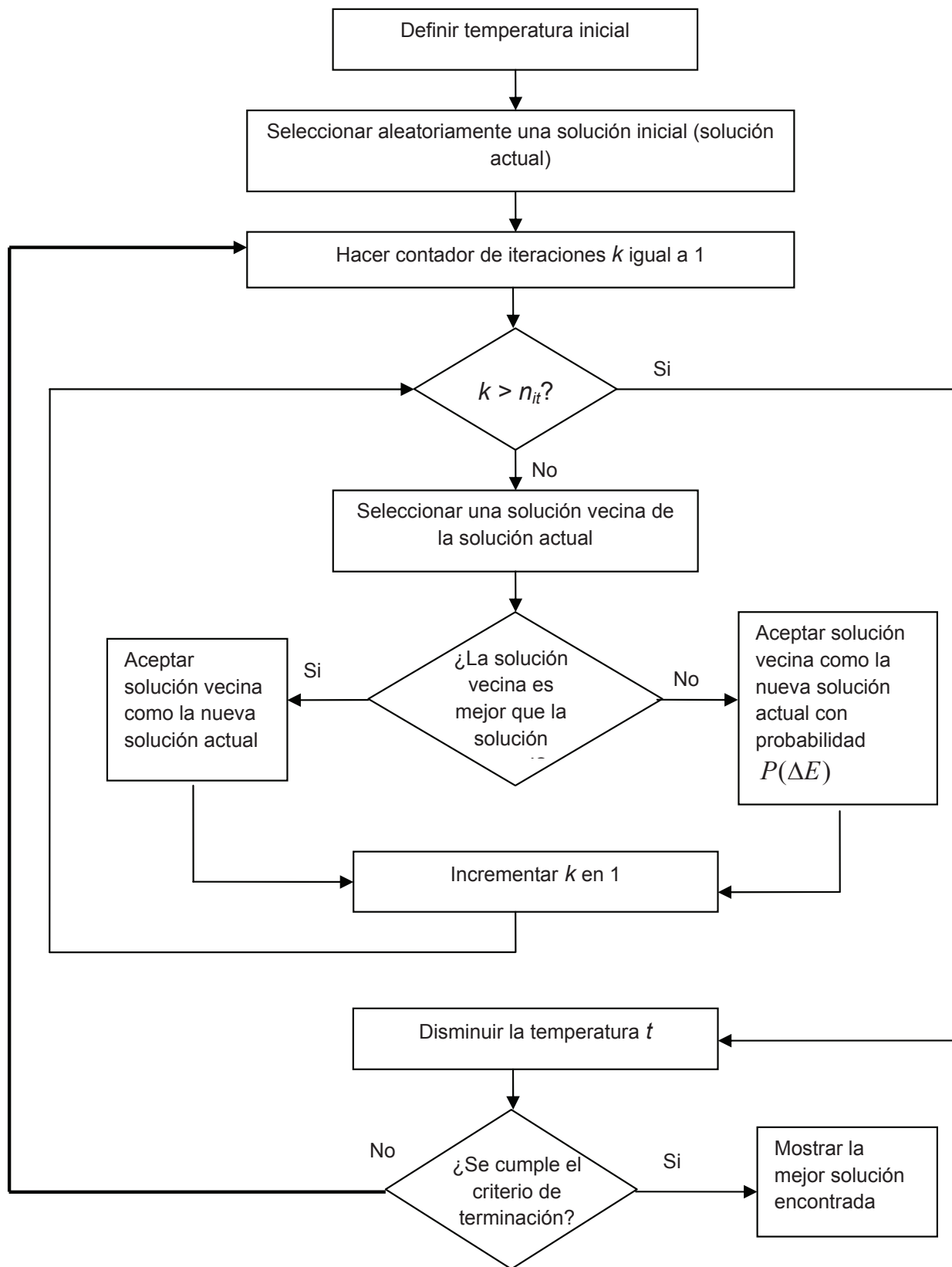


Figura 1. Diagrama de flujo básico de recocido simulado.

tida y la inicialización de una estructura de memoria (memoria de corto plazo), luego continúa un procedimiento iterativo de búsqueda de una mejor solución. En cada iteración se define un conjunto de soluciones vecinas de la solución actual, no se incluyen en este conjunto las soluciones vecinas que sean tabú, es decir, aquellas que hayan sido exploradas recientemente.

La Búsqueda Tabú evalúa todas las soluciones vecinas del conjunto y selecciona la mejor de ellas, aún cuando esta solución desmejore el valor objetivo en relación a otras soluciones previamente encontradas. Si esta solución es mejor que la mejor solución encontrada hasta el momento, se establece como la nueva mejor solución encontrada. Al fina-

lizar cada iteración, la solución actual se almacena en la memoria de corto plazo, también llamada lista tabú. De esta manera, la búsqueda tabú en forma iterativa va de una solución a otra hasta que se satisfaga un criterio de parada, lo que puede ocurrir cuando se ha superado un número máximo de iteraciones o ha pasado un cierto número de iteraciones sin que haya habido una mejora en la función objetivo. En la figura 2 se muestra el diagrama de flujo básico de la Búsqueda Tabú. En versiones mejoradas del algoritmo se puede incluir una estructura de memoria de largo plazo, con estrategias de intensificación de la búsqueda en un área promisorria del espacio de soluciones y diversificación hacia otras zonas no exploradas (Glover y Melián, 2003).

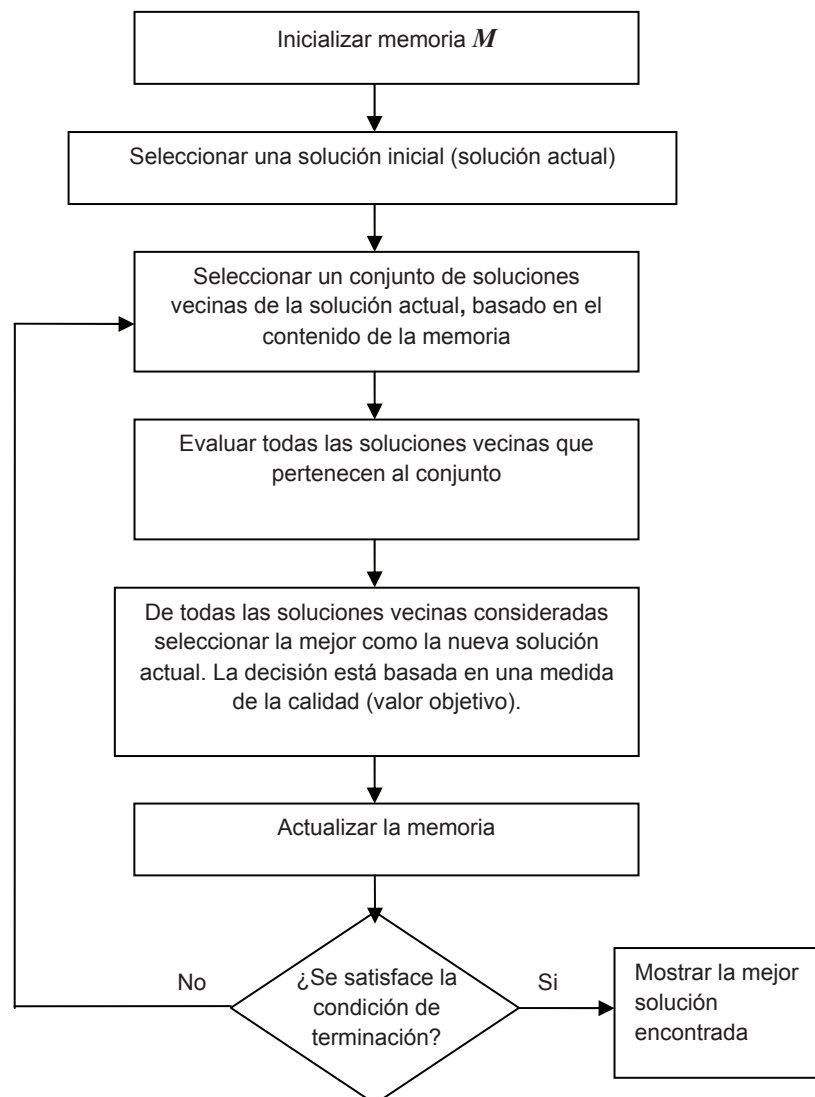


Figura 2. Diagrama de flujo básico de búsqueda tabú.

## 2.4 Solución del modelo mediante programación entera binaria

Con el fin de realizar comparaciones con los métodos heurísticos, se resolvió el modelo haciendo uso del software de optimización SAS OR, el cual utiliza el algoritmo exacto de ramificación y acotamiento para obtener la solución óptima de modelos de programación entera (SAS, 2004).

## 2.5 Solución del modelo mediante métodos heurísticos

Se diseñaron e implementaron dos algoritmos heurísticos para resolver el problema, utilizando las técnicas de Recocido Simulado y Búsqueda Tabú. Los programas de computación correspondientes se hicieron en Visual Basic 6.0. Los detalles para la implementación de los algoritmos heurísticos se describen a continuación.

### 2.5.1 Representación de una solución

Se hizo mediante un vector de ceros y unos, cada elemento del vector almacena una variable binaria  $X_{ij}$  que indica si un determinado rodal  $i$  se cortará o no en el año  $j$ . El tamaño del vector es igual al número de rodales multiplicado por el período de planificación ( $6 \times 5 = 30$ ). Esta representación vectorial se utilizó tanto en el algoritmo de recocido simulado como en el de Búsqueda Tabú. En la figura 3 se muestra la representación de una posible solución; cada rodal tiene asociados cinco elementos del vector correspondientes a los cinco años del período de planificación, uno de estos elementos tiene asignado el valor 1 para indicar que en ese año se corta ese rodal, los demás años en que también se pudiera aprovechar el rodal tienen asignado el valor 0 y las celdas marcadas con x se refieren a años en los que no se puede cortar un rodal debido que no ha alcanzado la edad mínima de corta.

### 2.5.2 Generación de la solución inicial

Se hizo aleatoriamente en los dos algoritmos heurísticos. Para cada uno de los rodales se elige al azar un año en el que será cortado y se asigna 1 a la variable (posición del vector) correspondiente, a las demás variables de ese rodal que representan a los otros años del período de planificación se les asigna 0.

### 2.5.3 Generación de soluciones vecinas

El procedimiento utilizado para generar una solución vecina a partir de una solución actual, es elegir aleatoriamente en esta última un rodal para modificar su año de corta. Así, una solución se diferencia de una vecina solamente por uno de los rodales el cual se corta en años diferentes.

### 2.5.4 Función de costo

Esta función mide la calidad de una solución y se define de tal manera que represente el problema que se está tratando de resolver, también se le denomina función de evaluación y está relacionada a la función objetivo. En los algoritmos de Recocido Simulado y Búsqueda Tabú desarrollados en este trabajo se utilizó una función de costo que incluye la función objetivo original (Ecuación 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 costo que aquellas soluciones que satisfacen las restricciones. La función de costo o función de evaluación está dada por la ecuación 6.

$$f^p(\mathbf{X}) = f(\mathbf{X}) + P^* VT(\mathbf{X}) \quad (\text{Ec.6})$$

Donde:

$f(\mathbf{X})$  = la función objetivo del problema original

$P$  = la penalización (u.m./t) que se impone a la violación de las restricciones de demanda

$VT(\mathbf{X})$  = la violación total (t) de las restricciones de demanda para una solución  $\mathbf{X}$ , la cual se calcula según la ecuación 7.

$$VT(\mathbf{X}) = \sum_{k=1}^5 (20000 - g_k) \cdot I_k \quad (\text{Ec.7})$$

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, es decir, el volumen que se corta en ese año es menor a la cuota anual ( $g_k < 20000$ ), y en otro caso vale 0.

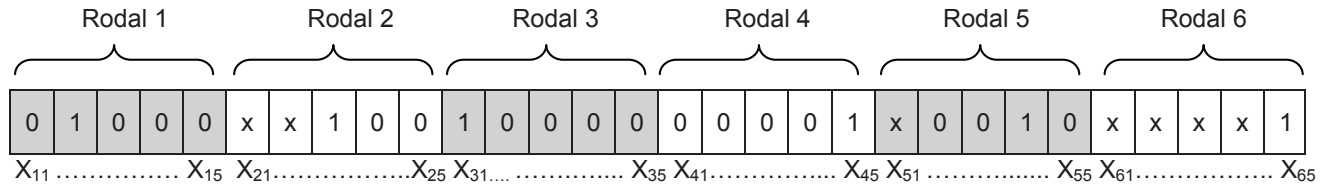


Figura 3. Representación de una solución del problema.

Las restricciones de singularidad (Ecuación 3) no se consideraron en la función de costo debido a que el cumplimiento de estas restricciones se garantiza en los procedimientos utilizados para generar una solución inicial y soluciones vecinas.

**2.5.5 Curva de enfriamiento para el algoritmo de recocido simulado**

Determina la velocidad de disminución de la temperatura a medida que avanzan las iteraciones del algoritmo. Existen diferentes maneras de abordar el decrecimiento de la temperatura, en este trabajo se utilizó la forma exponencial o geométrica según la ecuación 8, debido a su simplicidad y a los buenos resultados que ha dado en numerosas aplicaciones.

$$t_{k+1} = \alpha t_k \tag{Ec.8}$$

Donde:

- $t_{k+1}$  = la temperatura en la iteración  $k+1$
- $t_k$  = la temperatura en la iteración  $k$  y  $\alpha$  es una constante cercana a 1, escogida por lo general en el rango de 0,9 a 0,99.

**2.5.6 Estrategias de implementación para el algoritmo de Búsqueda Tabú**

Hay varios aspectos que deben definirse para implementar un algoritmo de Búsqueda Tabú, tales como la permanencia tabú, la estructura de la memoria de corto plazo y los criterios de aspiración que permiten obviar la condición de tabú para algunas soluciones.

- *Permanencia tabú*: es el número iteraciones que una solución, movimiento o atributo permanece en condición de tabú, puede ser un número fijo o puede variar en el transcurso de la búsqueda. En este trabajo se consideró un valor fijo para la permanencia tabú.

- *Estructura de la memoria de corto plazo o lista tabú*: se implementó como un vector de tamaño igual al número de rodales; 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. Estos valores se actualizan en cada iteración.
- *Criterios de aspiración*: permiten considerar soluciones tabú de excelente calidad y que pueden ser ignoradas por estar prohibidas. En el algoritmo desarrollado se permite un movimiento tabú si el valor objetivo de la solución resultante es mejor que las soluciones hasta ese momento encontradas.

**2.5.7 Criterio de parada de los algoritmos**

A fin de poder realizar comparaciones se utilizó el mismo criterio de parada para los dos algoritmos heurísticos, definido como un número máximo de iteraciones. Los programas finalizan después de realizar 100 iteraciones.

**2.5.8 Definición de los parámetros de los algoritmos**

En el caso de recocido simulado los parámetros son: temperatura inicial, tasa de disminución de la temperatura ( $\alpha$ ) y número de iteraciones en cada nivel de temperatura ( $n_{it}$ ). En el algoritmo de búsqueda tabú se tiene como parámetro la permanencia tabú. Los parámetros requeridos en los dos algoritmos heurísticos se establecieron de forma empírica mediante experimentación, pues no existen valores estándar que puedan ser utilizados en diferentes problemas.

**2.6 Evaluación del desempeño de los algoritmos heurísticos**

Para evaluar el desempeño de las técnicas heurísticas utilizadas en este trabajo se realizaron 100



corridas de cada algoritmo, en todas las corridas se registró el valor objetivo obtenido, la iteración en la que se encuentra la mejor solución, el tiempo de ejecución y se calculó el error relativo del valor objetivo obtenido (costo total de aprovechamiento) con respecto al valor óptimo obtenido con el método exacto de ramificación y acotamiento. La ecuación 9 es la utilizada para el error relativo, la cual se expone a continuación:

$$\text{Error relativo} = \frac{V_{obh} - V_{opt}}{V_{opt}} \times 100\% \quad (\text{Ec.9})$$

Donde:

$V_{opt}$  = valor objetivo óptimo obtenido mediante programación entera binaria

$V_{obh}$  = valor objetivo obtenido en una corrida de un algoritmo heurístico.

Para cada heurística se calcularon las estadísticas básicas de las variables valor objetivo, error relativo, iteración en que se consigue la mejor solución y el tiempo de ejecución. La eficacia de una técnica heurística se determinó como la diferencia entre el 100% y el error relativo medio. Además, se contabilizó el número de corridas en las cada heurística obtuvo la solución óptima.

### 3. Resultados y discusión

La solución obtenida por el programa de optimización SAS OR utilizando el algoritmo de ramificación y acotamiento se muestra en el cuadro 4, así como también el volumen total aprovechado y el tiempo de ejecución en milisegundos (ms) requerido por SAS OR para resolver el problema. Los parámetros utilizados para ejecutar los algoritmos de recocido simulado y búsqueda tabú se presentan en el cuadro 5 y las estadísticas básicas de las variables de desempeño evaluadas en el cuadro 6.

El algoritmo de búsqueda tabú tiene un valor objetivo medio igual al valor óptimo obtenido por el algoritmo de ramificación y acotamiento y una desviación estándar igual a cero, esto se debe a que encontró la solución óptima del problema en el 100% de las corridas realizadas. El algoritmo de Recocido Simulado consiguió la solución óptima del modelo en el 87% de las corridas, por lo que registró un valor objetivo medio superior al valor óptimo. El error relativo cuantifica esta diferencia

entre el valor objetivo obtenido por el algoritmo de Recocido Simulado y el valor óptimo, puede notarse que el error relativo medio es del 0,04%, con un valor mínimo y un valor máximo iguales a 0% y 0,89%, respectivamente; estos valores del error son bastantes bajos, lo que indica que en las corridas en las que el algoritmo de Recocido Simulado no alcanzó la solución óptima encontró soluciones muy cercanas al óptimo.

Al calcular la eficacia de las técnicas heurísticas como la diferencia entre el 100% y el error relativo medio, se obtuvo una eficacia del 100% para la Búsqueda Tabú y del 99,96% para Recocido Simulado. Ambos algoritmos tuvieron un excelente desempeño al resolver el problema planteado y concuerdan con lo señalado por Bettinger *et al.* (2002), en un estudio en el que clasificaron varias técnicas heurísticas en tres tipos de acuerdo al valor de la función objetivo que obtienen: muy buenas, adecuadas y no tan adecuadas; la Búsqueda Tabú y el Recocido Simulado se encuentran en el grupo de las técnicas "muy buenas".

El tiempo de ejecución promedio de los algoritmos heurísticos utilizados fue menor que el tiempo de ejecución del algoritmo de ramificación y acotamiento. Aunque en este problema las diferencias en el tiempo de ejecución de los algoritmos son imperceptibles a simple vista, ya que están en el orden de los milisegundos (ms), en problemas de mayor envergadura el tiempo de computación necesario para conseguir una solución es un factor muy importante.

La Búsqueda Tabú resultó ser una técnica más rápida que el Recocido Simulado, en cuanto al tiempo de ejecución del algoritmo y a la velocidad con que encuentra la solución, ya que en promedio la Búsqueda Tabú consiguió la mejor solución en la iteración 14 y el tiempo de ejecución medio fue de 43,5 ms, mientras que el Recocido Simulado en promedio encontró en la mejor solución en la iteración 26 y el tiempo de ejecución medio fue de 250,8 ms.

De acuerdo a las variables de desempeño evaluadas, la técnica de Búsqueda Tabú fue la mejor en la resolución del problema planteado obteniendo la solución óptima en todas las corridas realizadas y en un tiempo de ejecución menor. Sin embargo, es importante destacar una característica de los resultados obtenidos con Recocido Simulado y que es común observar al resolver problemas de optimi-

**Cuadro 4.** Solución óptima del problema obtenida por el algoritmo de ramificación y acotamiento.

Año	Rodal(es) a cortar	Volumen total aprovechado (t)
1	1	39788
2	5	26957
3	3	33379
4	4	29803
5	2 y 6	25226
Total		155153
Costo mínimo (u.m.*)		5457192,8
Tiempo de ejecución (ms**)		710

\* Unidades monetarias; \*\* Milisegundos

**Cuadro 5.** Parámetros utilizados en los algoritmos heurísticos

Recocido Simulado	Búsqueda Tabú
<ul style="list-style-type: none"> <li>• Temperatura inicial: <math>10 \times 10^8</math></li> <li>• Iteraciones del algoritmo: 100</li> <li>• Iteraciones en cada nivel de temperatura: 50</li> <li>• Tasa de disminución de la temperatura (<math>\alpha</math>) : 0,9</li> <li>• Valor de penalización : 1000</li> </ul>	<ul style="list-style-type: none"> <li>• Permanencia tabú: 2</li> <li>• Iteraciones del algoritmo: 100</li> <li>• Valor de penalización: 1000</li> </ul>

**Cuadro 6.** Estadísticas básicas para las variables de desempeño de los algoritmos heurísticos

	Recocido Simulado	Búsqueda Tabú
<b>Valor Objetivo (u.m.)</b>		
Media	5459613,5	5457192,8
Desviación estándar	7983,5	0
Mínimo	5457192,8	5457192,8
Máximo	5505711,7	5457192,8
<b>Error relativo (%)</b>		
Media	0,04	0
Desviación estándar	0,15	0
Mínimo	0	0
Máximo	0,89	0
<b>Iteración en la que se encuentra la mejor solución</b>		
Media	26	14
Desviación estándar	15	12
Mínimo	2	3
Máximo	53	43
<b>Tiempo de ejecución (ms)</b>		
Media	250,8	43,5
Desviación estándar	3,9	5,4
Mínimo	246,0	35,0
Máximo	264,0	62,0

zación con técnicas heurísticas, el algoritmo alcanzó la solución óptima en el 87% de las corridas y en el 13% de las corridas restantes encontró varias soluciones cercanas al óptimo, que representan otras secuencias de corta alternativas cuyo costo total de aprovechamiento está muy cerca del costo mínimo. Esta es una ventaja que en general presentan las técnicas heurísticas con respecto a los métodos clásicos de programación matemática, éstos encuentran una única solución, que aunque es la óptima, a veces en la práctica no es posible implementarla debido a los supuestos que se hicieron al elaborar el modelo, a los errores presentes en los datos utilizados o simplemente porque carece de factibilidad de aplicación. En contraparte, las heurísticas pueden encontrar varias buenas soluciones ofreciendo diferentes alternativas de decisión.

#### 4. Conclusiones

Los métodos heurísticos tienen un enorme potencial en la resolución de problemas de planificación forestal, en los últimos años han mostrado su efectividad en diferentes aplicaciones, desde problemas de planificación sencillos hasta problemas complejos que no pueden ser resueltos con técnicas clásicas de programación matemática. Además, pueden encontrar buenas soluciones a problemas de grandes dimensiones en un tiempo de computación aceptable. Los beneficios que aportan las técnicas heurísticas se derivan de la posibilidad de resolver problemas de planificación más realistas y menos rígidos que los que se pueden plantear con técnicas derivadas de la programación matemática.

Las dos técnicas heurísticas utilizadas en este trabajo dieron buenos resultados al resolver el ejemplo de aplicación. La Búsqueda Tabú fue superior al Recocido Simulado puesto que presentó un error relativo menor, una eficacia del 100%, mayor rapidez para encontrar la solución y menores tiempos de ejecución. El Recocido Simulado consiguió la solución óptima en la mayoría de las corridas del programa, y además encontró otras soluciones muy cercanas al óptimo que constituyen planes de corta alternativos entre los cuales el planificador puede decidir. Los resultados obtenidos en este trabajo no pueden generalizarse para otras situaciones, ya que el desempeño de las técnicas heurís-

ticas depende del problema particular y de las estrategias diseñadas para resolverlo.

Un factor muy importante que se evidenció en esta investigación es el hecho que los métodos heurísticos pueden resolver problemas de optimización en un tiempo de computación menor al que puede requerir un algoritmo exacto como el de ramificación y acotamiento. En muchas situaciones reales se requiere de procedimientos que suministren de forma rápida una solución suficientemente buena, en estos casos las técnicas heurísticas juegan un papel preponderante.

#### 5. Referencias bibliográficas

- BETTINGER, P., J. SESSIONS y K. BOSTON. 1997. Using Tabu search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecological Modelling* 94(2-3): 111-123.
- BETTINGER, P., K. BOSTON, J. SESSIONS y W. CHUNG. 2002. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. *Silva Fennica* 36(2): 561-584.
- BETTINGER, P., K. BOSTON, Y.H. KIM y J. ZHUA. 2007. Landscape-level optimization using tabu search and stand density-related forest management prescriptions. *European Journal of Operational Research* 176(2): 1265-1282.
- BETTINGER, P., K. BOSTON, J.P. SIRY y D.L. GREBNER. 2009. *Forest Management and Planning*. Elsevier, Academic Press. EEUU. 331 p.
- BRUMELLE, S., D. GRANOT, M. HALME e I. VERTINSKY. 1998. A tabu search algorithm for finding a good forest harvest schedule satisfying green - up constraints. *European Journal of Operations Research* 106: 408-424.
- CHIARI, R., O. CARRERO, M. JEREZ, M.A. QUINTERO y J. STOCK. 2008. Modelo preliminar para la planificación del aprovechamiento en plantaciones forestales industriales en Venezuela. *Interciencia* 33 (011): 802-809.
- CLEMENTS, S.E., P.L. DALLAIN y M.S. JAMNICK. 1990. An operational, spatially constrained harvest scheduling model. *Canadian Journal of Forest Research* 20: 1438-1447.
- CROWE, K.A. y J. D. NELSON. 2005. An evaluation of the simulated annealing algorithm for solving the area-restricted harvest-scheduling model against optimal benchmarks. *Canadian Journal of Forest Research-*

- Revue Canadienne De Recherche Forestiere* 35(10): 2500-2509.
- DÍAZ, A., J.A. FERLAND, C.C. RIBEIRO, J.R. VERA y A. WEINTRAUB. 2007. A tabu search approach for solving a difficult forest harvesting machine location problem. *European Journal of Operational Research* 179(3): 788-805.
- DUCHEYNE, E.I., R.R. DE WULF y B. DE BAETS. 2004. Single versus multiple objective genetic algorithms for solving the even-flow forest management problem. *Forest Ecology and Management* 201(2-3): 259-273.
- FALCÃO, A.O. y J.G. BORGES. 2001. Designing an evolution program for solving integer forest management scheduling models: an application in Portugal. *Forest Science* 47(2): 158-168.
- GLOVER, F. 1986. Future paths for integer programming and links to Artificial Intelligence. *Computers and Operations Research* 13: 533-549.
- GLOVER, F. y B. MELIÁN. 2003. Búsqueda Tabú. *Revista Iberoamericana de Inteligencia Artificial* 19: 29-48.
- HILLIER, F.S. y G.J. LIEBERMAN. 2001. *Introduction to Operations Research*. Séptima edición. Mc Graw Hill. NY, EEUU. 1204 p.
- LOCKWOOD, C. y T. MOORE. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. *Canadian Journal of Forest Research* 23: 468-478.
- LU, F. y L.O. ERIKSSON. 2000. Formation of harvest units with genetic algorithms. *Ecology and Management* 130(1-3): 57-67.
- METROPOLIS, N., A.W. ROSENBLUTH, M.N. ROSENBLUTH., A.H. TELLER y E. TELLER. 1953. Equation of state calculation by fast computing machines. *Journal of Chemistry* 21: 1087 – 1091.
- NELSON, J. y J.D. BRODIE. 1990. Comparison of random search algorithm and mixed integer programming for solving area – based forest plans. *Canadian Journal of Forest Research* 20: 934-942.
- O'HARA, A.J., B.H. FAALAND y B.B. BARE. 1989. Spatially constrained timber harvest scheduling. *Canadian Journal of Forest Research* 19: 715 – 724.
- OHMAN, K. y L.A. ERIKSSON. 2002. Allowing for spatial consideration in long – term forest planning by linking programming with simulated annealing. *Forest Ecology and Management* 161 (1): 221-230.
- RICHARDS, E. W. y E. GUNN. 2000. A Model and tabu search method to optimize stand harvest and road construction schedules. *Forest Science* 46 (2): 188-203.
- RICHARDS, E.W. y E.A. GUNN. 2003. Tabu search design for difficult forest management optimization problems. *Canadian Journal of Forest Research* 33(6): 1126-1133.
- SAS. 2004. SAS/OR 9.1 User's Guide. SAS Institute Inc. Cary, NC, EEUU. 168 p.
- THOMPSON, M.P, J.D. HAMANN y J. SESSIONS. 2009. Selection and penalty strategies for genetic algorithms designed to solve spatial forest planning problems. *International Journal of Forestry Research* 2009: 1-14.