

Universidad de Costa Rica  
Escuela de Ciencias de la Computación e Informática  
CI-0116 Análisis de Algoritmos y Estructuras de Datos

Análisis Comparativo de Algoritmos de Búsqueda:  
Exhaustiva, Greedy y Ramificación y Poda

Priscilla López Quesada — C14301  
Karol Valeria Bolaños Sánchez — C31205

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Marco Teórico</b>	<b>3</b>
2.1. Búsqueda Exhaustiva . . . . .	3
2.2. Algoritmo Greedy . . . . .	3
2.3. Ramificación y Poda . . . . .	3
<b>3. Metodología</b>	<b>4</b>
<b>4. Resultados Experimentales</b>	<b>4</b>
4.1. Problema de Asignación . . . . .	4
4.2. Problema de Distribución . . . . .	5
4.3. Problema de la Mochila . . . . .	5
4.4. Problema del Vendedor Viajero . . . . .	6
<b>5. Análisis Comparativo</b>	<b>6</b>
5.1. Comparación de Tiempos de Ejecución . . . . .	6
5.2. Comparación de Calidad de Soluciones . . . . .	7
5.3. Eficiencia de la Poda . . . . .	7
<b>6. Análisis</b>	<b>8</b>
<b>7. Conclusiones</b>	<b>9</b>

# 1. Introducción

El informe tiene como objetivo analizar y comparar tres estrategias de búsqueda en la resolución de problemas de optimización combinatoria: búsqueda exhaustiva, algoritmo greedy y ramificación y poda. Para este análisis se implementaron y probaron estos tres algoritmos en cuatro problemas clásicos de optimización: el problema de asignación, distribución, mochila y el problema del vendedor.

La comparación se realizó usando tres categorías de tamaño: pequeñas, medianas y grandes. Esto permitió observar cómo cada algoritmo se comporta a medida que aumenta la complejidad del problema, especialmente en términos de tiempo de ejecución, número de soluciones evaluadas y la solución obtenida. Los resultados otorgan evidencia sobre las ventajas y limitaciones de cada técnica, lo cual es necesario para comprender cuándo es apropiado utilizar cada enfoque en la práctica.

## 2. Marco Teórico

### 2.1. Búsqueda Exhaustiva

La búsqueda exhaustiva garantiza encontrar la solución óptima al evaluar todas las posibles configuraciones del espacio de búsqueda. Su principal ventaja es la facilidad en su concepto y que es muy certera. Aún así, su mayor desventaja es su complejidad, porque crece de manera factorial o exponencial dependiendo del problema, lo que la hace impráctica para instancias de tamaño mediano o grande.

### 2.2. Algoritmo Greedy

Los algoritmos greedy construyen una solución a través de decisiones localmente óptimas en cada paso, sin considerar las consecuencias futuras. Esto resulta en algoritmos muy eficientes con complejidad usualmente polinomial. De igual manera, no garantiza la optimalidad de la solución final, y la calidad del resultado depende fuertemente de la estructura del problema.

### 2.3. Ramificación y Poda

Ramificación y poda combina la exploración del espacio de búsqueda con la capacidad de eliminar ramas que no pueden conducir a soluciones mejores que la mejor conocida hasta el momento. Esto utiliza cotas superiores e inferiores para podar el árbol de búsqueda, reduciendo el número de soluciones evaluadas. Aunque su implementación es más compleja que los otros dos enfoques, logra garantizar optimalidad con un tiempo de ejecución mucho mejor que la búsqueda exhaustiva en la mayoría de los casos.

### 3. Metodología

Para cada uno de los cuatro problemas, se generaron instancias de tres tamaños diferentes. Las instancias pequeñas permitieron validar la correctitud de las implementaciones mediante comparación con la búsqueda exhaustiva. Las instancias medianas sirvieron para observar el inicio de las diferencias significativas en el desempeño de los algoritmos, y por último, las instancias grandes nos demostraron las verdaderas limitaciones y fortalezas de cada algoritmo.

Las pruebas se ejecutaron midiendo el tiempo de ejecución en segundos, el número de soluciones evaluadas y el valor de la función objetivo alcanzado. Para el problema de asignación y distribución se buscó maximizar la ganancia, para la mochila maximizar el beneficio, y para el vendedor viajero minimizar el costo total del recorrido.

## 4. Resultados Experimentales

### 4.1. Problema de Asignación

El problema de asignación presenta un espacio de búsqueda factorial de tamaño  $n!$ , donde  $n$  representa el número de elementos a asignar. Los resultados están en la Tabla 1.

Cuadro 1: Resultados del problema de asignación

Método	Tamaño	Ganancia	Soluciones	Tiempo (s)
<i>Tamaño pequeño</i>				
Greedy	8	1963	—	0.0000
Exhaustiva	3	630	6	0.0000
Ramificación	8	2095	16	0.0000
<i>Tamaño mediano</i>				
Greedy	12	3130	—	0.0000
Exhaustiva	4	1036	24	0.0000
Ramificación	12	3277	45	0.0300
<i>Tamaño grande</i>				
Greedy	18	4750	—	0.0000
Exhaustiva	5	1195	120	0.0000
Ramificación	18	5012	148	15.8500

En este problema se observa que el algoritmo greedy proporciona soluciones rápidas pero no tan óptimas. Para el tamaño grande, la ramificación y poda alcanzó una ganancia de 5012, mientras que greedy solo obtuvo 4750, representando una diferencia del 5.2%. El tiempo de ejecución de ramificación y poda fue de 15.85 segundos, lo cual es aceptable considerando que evaluó únicamente 148 soluciones en lugar de las 18! (aproximadamente  $6,4 \times 10^{15}$ ) que necesitaría una búsqueda exhaustiva completa.

## 4.2. Problema de Distribución

El problema de distribución involucra asignar  $n$  elementos idénticos a  $m$  categorías, generando un espacio combinatorio de tamaño  $\binom{n+m-1}{m-1}$ . Los resultados están en la Tabla 2.

Cuadro 2: Resultados del problema de distribución

Método	Tamaño	Ganancia	Soluciones	Tiempo (s)
<i>Tamaño pequeño</i>				
Greedy	8/8	331	—	0.0000
Exhaustiva	3/3	81	20	0.0000
Ramificación	8/8	332	35	0.0100
<i>Tamaño mediano</i>				
Greedy	12/12	356	—	0.0000
Exhaustiva	4/4	156	70	0.0000
Ramificación	12/12	426	22	0.8000
<i>Tamaño grande</i>				
Greedy	18/18	732	—	0.0000
Exhaustiva	5/5	170	252	0.0000
Ramificación	18/18	750	204	3930.2400

Este problema tiene un comportamiento que consideramos interesante. En tamaño mediano, ramificación y poda logró una ganancia de 426 con solo 22 evaluaciones, superando significativamente al greedy que obtuvo 356. Pero, en el tamaño grande, el tiempo de ejecución se fue más de una hora (3930 segundos), aunque la diferencia en ganancia fue: 750 versus 732. Esto nos dice que las cotas para este problema no son tan efectivas como en otros casos, y el espacio de búsqueda es difícil de podar de una forma más eficaz.

## 4.3. Problema de la Mochila

El problema de la mochila tiene un espacio de búsqueda exponencial de tamaño  $2^n$ , donde cada elemento puede estar presente o ausente en la solución. En la Tabla 3 podemos observar los resultados:

El problema de la mochila tiene un resultado muy interesante para ramificación y poda. En el tamaño grande, alcanzó el beneficio de 689 evaluando solamente 35 de las  $2^{18} = 262144$  posibles combinaciones, en apenas 0.02 segundos. Además, el algoritmo greedy coincidió con el óptimo en este caso, lo que nos dice que las instancias generadas fueron favorables para el ordenamiento por relación valor-peso. Esto es típico en el problema de la mochila cuando los ítems tienen características relativamente uniformes.

Cuadro 3: Resultados del problema de la mochila

Método	Tamaño	Beneficio	Soluciones	Tiempo (s)
<i>Tamaño pequeño</i>				
Greedy	8	338	—	0.0000
Exhaustiva	3	78	4	0.0000
Ramificación	8	338	12	0.0000
<i>Tamaño mediano</i>				
Greedy	12	480	—	0.0000
Exhaustiva	4	156	9	0.0000
Ramificación	12	480	28	0.0000
<i>Tamaño grande</i>				
Greedy	18	689	—	0.0000
Exhaustiva	5	158	16	0.0000
Ramificación	18	689	35	0.0200

#### 4.4. Problema del Vendedor Viajero

El problema del vendedor viajero es uno de los problemas más clásicos, con una búsqueda factorial de  $(n - 1)!$  rutas posibles. Los resultados están en la Tabla 4.

El problema del viajero nos enseña una de las mayores debilidades del algoritmo greedy. En el tamaño grande, greedy produjo una ruta con costo 832, mientras que ramificación y poda encontró el óptimo en 609, lo que representa una diferencia del 36.6 %. Esta diferencia tan grande se debe a que la heurística del vecino más cercano utilizada por greedy puede quedar atrapada en óptimos locales fácilmente. Por otro parte, ramificación y poda ocupó 101 segundos y evaluó 549 soluciones, lo cual es manejable considerando que el espacio completo contiene  $17! \approx 3,56 \times 10^{14}$  rutas.

## 5. Análisis Comparativo

### 5.1. Comparación de Tiempos de Ejecución

La Figura 1 presenta una comparación de los tiempos de ejecución para cada problema en tamaño grande. Utilizamos una escala logarítmica por las grandes diferencias en magnitud.

Podemos observar con estos datos que el algoritmo greedy es realmente el más rápido, con tiempos prácticamente instantáneos en todos los casos. La ramificación y poda muestra un desempeño variable: increíble en la mochila (0.02s), medianamente aceptable en asignación (15.85s) y vendedor (101.05s), pero muy problemático en distribución (3930.24s).

Cuadro 4: Resultados del problema del vendedor viajero

Método	Tamaño	Costo	Soluciones	Tiempo (s)
<i>Tamaño pequeño</i>				
Greedy	8	836	—	0.0000
Exhaustiva	3	579	2	0.0000
Ramificación	8	569	62	0.0000
<i>Tamaño mediano</i>				
Greedy	12	597	—	0.0000
Exhaustiva	4	498	6	0.0000
Ramificación	12	533	64	0.1500
<i>Tamaño grande</i>				
Greedy	18	832	—	0.0000
Exhaustiva	5	698	24	0.0000
Ramificación	18	609	549	101.0500

## 5.2. Comparación de Calidad de Soluciones

Para analizar las soluciones, calculamos la diferencia de optimalidad del algoritmo greedy respecto a ramificación y poda en los problemas de tamaño grande. Los resultados están en la Tabla 5.

Cuadro 5: Gap de optimalidad del algoritmo greedy

Problema	Greedy	Ramificación	Gap	Gap (%)
Asignación	4750	5012	262	5.23 %
Distribución	732	750	18	2.40 %
Mochila	689	689	0	0.00 %
Vendedor	832	609	223	36.62 %

Los resultados nos dicen que la efectividad del algoritmo greedy depende totalmente del problema. En la mochila alcanzó el óptimo, en distribución estuvo muy cerca (2.4 %), en asignación tuvo una desviación moderada (5.2 %), pero en el vendedor viajero falló muchísimo (36.6 %). Esta variabilidad nos confirma que los algoritmos greedy requieren un análisis cuidadoso del problema antes de ser aplicados en contextos donde la optimalidad es necesaria.

## 5.3. Eficiencia de la Poda

Un detalle importante de la ramificación y poda es su capacidad para reducir el espacio de búsqueda. La Tabla 6 muestra cuántas soluciones evaluó en relación al espacio total.

Estos números nos demuestran el poder de la poda. En el problema de asignación, se evaluó muchísimo menos del espacio total. En la mochila, apenas el 0.013 % de las combinaciones

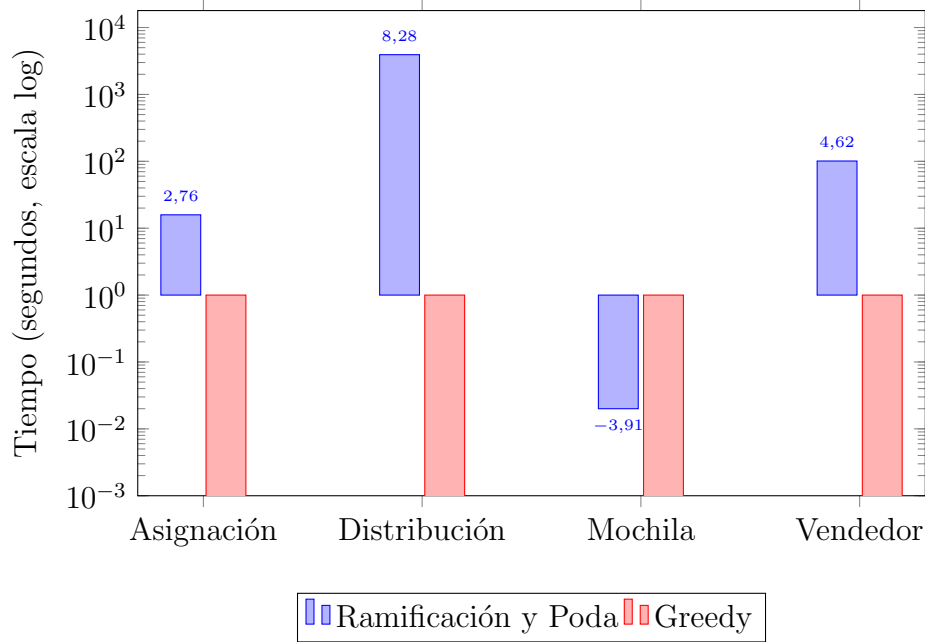


Figura 1: Comparación de tiempos de ejecución en problemas de tamaño grande

Cuadro 6: Eficiencia de poda en problemas de tamaño grande

Problema	Espacio total	Evaluadas	Porcentaje
Asignación	$18! \approx 6,4 \times 10^{15}$	148	$2,3 \times 10^{-12} \%$
Distribución	$\binom{35}{17} \approx 4,5 \times 10^9$	204	$4,5 \times 10^{-6} \%$
Mochila	$2^{18} = 262144$	35	0,013 %
Vendedor	$17! \approx 3,6 \times 10^{14}$	549	$1,5 \times 10^{-10} \%$

fue necesario.

## 6. Análisis

Los resultados confirman las características teóricas de cada algoritmo. La búsqueda exhaustiva, aunque garantiza optimalidad, solo fue aplicable para tamaños muy pequeños. Incluso con 5 elementos, el tiempo fue despreciable, pero en tamaños mayores es claramente impráctica por el crecimiento factorial o exponencial.

El algoritmo greedy demostró ser muy eficiente en tiempo de ejecución, pero su eficacia varió mucho entre problemas. En la mochila logró el óptimo, posiblemente debido a que la heurística de ordenar por valor-peso es particularmente adecuada para este tipo de instancias. En distribución y asignación, la pérdida de optimalidad fue medianamente aceptable para muchas aplicaciones prácticas. Pero, en el vendedor, la diferencia del 36 % respecto al óptimo lo hace inadecuado si se necesita calidad en la solución.

La ramificación y poda el algoritmo más versátil, porque demostró optimalidad en todos los casos mientras mantuvo tiempos de ejecución razonables, excepto en el problema de



distribución. El caso de la distribución, aunque logró una solución óptima, el tiempo de más de una hora nos dice que las cotas implementadas no fueron suficientemente restrictivas, permitiendo que el árbol de búsqueda creciera demasiado. Esto indica que la efectividad de ramificación y poda depende totalmente de las cotas usadas.

Algo que nos llamó la atención es cómo el número de soluciones evaluadas no siempre se relaciona directamente con el tiempo de ejecución. Por ejemplo, en el vendedor viajero se evaluaron 549 soluciones en 101 segundos, mientras que en distribución se evaluaron solo 204 en 3930 segundos. Esto es porque al calcular las cotas y gestionar el árbol de búsqueda tiene un costo por nodo que varía según el problema. En distribución, al parecer cada evaluación de nodo es altamente más costosa.

## 7. Conclusiones

Este análisis comparativo nos permite concluir sobre cada algoritmo que:

La búsqueda exhaustiva debe reservarse exclusivamente para tamaños pequeños o para validar la correctitud de otros algoritmos. Su crecimiento factorial o exponencial la hace impráctica para problemas reales.

Los algoritmos greedy son aceptables cuando se ocupa velocidad y se pueden aceptar soluciones medianamente óptimas. Su eficacia depende totalmente de la estructura del problema y la heurística elegida. Respecto al tiempo real o cuando se procesan grandes tamaños de datos donde la optimalidad no es crítica, son una excelente opción.

La ramificación y poda es la técnica recomendada cuando se necesita garantizar optimalidad en problemas de tamaño mediano a grande. Su implementación es más compleja y requiere un diseño más cuidadoso de las cotas, pero los resultados son muy buenos. La excepción fue el problema de distribución, donde el tiempo nos dice que para ese problema específico podrían usarse otras formas de implementación.

En conclusión, para problemas de optimización en aplicaciones reales, sería mejor utilizar greedy para obtener rápidamente una solución inicial y, si se requiere, aplicar ramificación y poda para mejorarla o verificar su optimalidad. Esto permitiría usar las ventajas de cada uno, como velocidad inicial con la posibilidad de mejora.

Finalmente, es importante destacar que los resultados obtenidos son respecto a los tamaños específicos generados para estas pruebas. Los resultados de los algoritmos pueden variar con diferentes distribuciones de datos o restricciones por aparte, por lo que es necesario realizar análisis específicos antes.