



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

---

FACULTAD DE CIENCIAS

RECOCIDO SIMULADO Y BÚSQUEDA TABÚ  
PARA PROBLEMAS DE PLANEACIÓN

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN

PRESENTA:

RODRIGO FERNANDO VELÁZQUEZ CRUZ

TUTORA

DRA. MARÍA DE LUZ GASCA SOTO





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Índice general

<b>Introducción</b>	<b>2</b>
<b>1. Antecedentes</b>	<b>3</b>
1.1. Problemas de optimización . . . . .	3
1.2. Heurísticas . . . . .	6
1.3. Metaheurísticas . . . . .	8
1.4. Problemas de planeación . . . . .	14
1.5. Planeación de proyectos sujeto a recursos (RCPPSP) . . . . .	21
<b>2. Planeación de proyectos sujeto a recursos con interrupciones (PRCPSP)</b>	<b>24</b>
2.1. Descripción del problema . . . . .	24
2.2. Estrategia de solución . . . . .	26
2.2.1. Proceso Branch and Bound . . . . .	28
2.2.2. Conclusiones del proceso Branch and Bound . . . . .	43
<b>3. Recocido Simulado</b>	<b>44</b>
3.1. Descripción . . . . .	44
3.2. Aplicado a PRCPSP . . . . .	46
<b>4. Búsqueda Tabú</b>	<b>98</b>
4.1. Descripción . . . . .	98
4.2. Intensificación y diversificación . . . . .	99
4.3. Aplicado a PRCPSP . . . . .	100
<b>5. Comparación de las metaheurísticas para el PRCPSP</b>	<b>127</b>
5.1. Generación de ejemplares aleatorios . . . . .	127
5.2. Resultados . . . . .	131
5.2.1. Ejemplar de 10 actividades . . . . .	132
5.2.2. Ejemplar de 20 actividades . . . . .	137
5.2.3. Ejemplar de 40 actividades . . . . .	142
5.2.4. Ejemplar de 80 actividades . . . . .	147
5.3. Análisis . . . . .	153
<b>Conclusiones</b>	<b>155</b>
<b>Bibliografía</b>	<b>156</b>
<b>Índice alfabético</b>	<b>158</b>
<b>Glosario</b>	<b>159</b>

# Introducción

El problema principal en el que se enfoca la investigación es la planeación<sup>1</sup> de proyectos sujeto a recursos con interrupciones, conocido en la literatura como PRCPSP<sup>2</sup>, donde las actividades del proyecto pueden ser interrumpidas un número entero de veces; además, este problema es una generalización del problema clásico de planeación de proyectos sujeto a recursos, conocido en la literatura como RCPSP<sup>3</sup>, ambos problemas usan la precedencia de actividades, de manera que una actividad sólo puede ser planeada cuando las actividades anteriores se hayan planeado y utilizan recursos renovables, los cuales se pueden usar a lo largo del tiempo en la planeación de cualquier actividad del proyecto siempre que no se exceda la disponibilidad de un recurso en cada tiempo.

Es de interés el problema PRCPSP porque al permitir interrupciones en la planeación de actividades, éstas no se quedan atadas a la utilización de un recurso durante todo el tiempo que se realizan, por lo que, si los recursos renovables son máquinas, personas, entre otros, se pueden alternar las actividades de tal manera que los recursos se aprovechen totalmente; esto significa, tener la mayor cantidad de recursos ocupados durante la realización de las actividades, como se indica en los siguientes capítulos, donde la acción de interrumpir una actividad para que sea realizada con algún otro recurso puede costar menos tiempo que no usar interrupciones. El objetivo principal del problema PRCPSP consiste en minimizar el tiempo que puede tomar la planeación de las actividades de un proyecto haciendo un número entero de interrupciones.

Dada la complejidad del problema, la investigación también se enfoca en la aplicación de las metaheurísticas de Recocido Simulado y Búsqueda Tabú; además de realizar una comparación del desempeño de ambas con distintos ejemplares del problema, estos ejemplares son construidos de una manera específica, porque de estos se conoce la planeación óptima, con lo cual se puede comprobar si la aplicación de las metaheurísticas obtiene buenas soluciones para el problema.

En el Capítulo 1 se presentan los antecedentes necesarios para el desarrollo de este trabajo. En la primer sección se presentan conceptos básicos en la optimización de problemas, en la segunda y tercer sección se describen los conceptos básicos de las heurísticas y metaheurísticas respectivamente, en la cuarta sección se presentan algunos problemas de planeación sobre máquinas y en la quinta sección se presenta el problema clásico de planeación RCPSP. En el Capítulo 2 se presenta el problema principal de la investigación, el PRCPSP, junto a un algoritmo de búsqueda específico. En el Capítulo 3 se presenta la metaheurística de Recocido Simulado y su aplicación para el PRCPSP. En el Capítulo 4 se presenta la metaheurística de Búsqueda Tabú y su aplicación para el PRCPSP. En el Capítulo 5 se describe la manera en que se construyen ejemplares y cómo se contruye su planeación óptima, con el objetivo de tener ejemplares conocidos para los cuales se pueda comprobar la efectividad de las metaheurísticas, posteriormente, se comparan las dos metaheurísticas para los ejemplares del PRCPSP, presentando las planeaciones obtenidas para algunos ejemplares del problema.

---

<sup>1</sup>Se usará planeación como traducción de *scheduling* ya que se adapta bien a la idea global del problema

<sup>2</sup>Del inglés '*Pre-emptive Resource-constrained Project Scheduling Problem*'

<sup>3</sup>Del inglés '*Resource-constrained Project Scheduling Problem*'



# Capítulo 1

## Antecedentes

En este capítulo se presentan los antecedentes necesarios para la actual investigación. En la sección 1.1 se presentan los conceptos básicos de problemas de optimización. En la sección 1.2 se presenta una breve descripción de las heurísticas, además de la búsqueda local. En la sección 1.3 se presenta una descripción de las metaheurísticas. En la sección 1.4 se presentan algunos problemas muy conocidos de planeación en máquinas. En la sección 1.5 se presenta el problema clásico de planeación RCPSP.

### 1.1. Problemas de optimización

Los problemas de optimización son muy relevantes en cualquier actividad humana, pues éstos representan el impulso natural de obtener la mejor solución ante cualquier problema. En los métodos de optimización, es de interés que los métodos sean eficientes en la práctica; es decir, que encuentren soluciones satisfactorias en un tiempo adecuado, también deben ser manejables, esto es, no sean complejos de usar y que se puedan adaptar a problemas de optimización similares a los problemas para los cuales éstos se diseñan. Al resolver un problema, se toman decisiones que guían hacia soluciones distintas, por lo que, se deben tomar las mejores decisiones en la búsqueda de una solución al problema.

Para resolver un problema que involucra optimización es necesario tomar en cuenta los siguientes pasos:

- **Reconocer el problema.** Hay que saber con precisión cuál es el problema de optimización que se intenta resolver, el objetivo a optimizar y más importante, si es un problema para el cual se conoce en la teoría que es decidible; también se requiere conocer la complejidad de éste. Asimismo es necesario identificar el problema, es decir, que esté descrito de manera clara y precisa, además de conocer los parámetros del problema, precondiciones, postcondiciones, restricciones y límites del problema.
- **Modelar el problema.** Para el problema se construye un modelo matemático, en éste se describen las relaciones y propiedades que describen al problema, representándose con símbolos y expresiones matemáticas (de aquí que el problema sea descrito de manera clara y precisa). Aunque al modelar el problema se abandonan características del mundo real (por lo complejo que es modelar aspectos reales), se tienen modelos más precisos y fáciles de resolver, con el costo de ser menos relevantes o descriptivos, aún con esto, el modelo debe ser una representación válida del mundo real y contener lo que en principio se quiere modelar; por último, el modelo debe ser apropiado para las estrategias de aproximación a utilizar.
- **Optimizar el problema.** Una vez modelado el problema, no es suficiente con obtener cualquier solución, es necesario buscar soluciones satisfactorias, lo primero es utilizar algoritmos exactos, si

éstos no resuelven el problema en un tiempo adecuado; es decir, por la complejidad del problema puede que sea tan difícil y al usar un algoritmo exacto se tome un tiempo exponencial, entonces es necesario usar otra estrategia como los algoritmos de aproximación o bien, como se detalla en las secciones posteriores, las heurísticas y metaheurísticas.

- **Implementar la solución.** En esta parte se traslada el modelo y el algoritmo o estrategia a utilizar en software, con el cual, se comprueba si éstos obtienen buenas soluciones al problema y en caso de ser necesario, hacer modificaciones en los pasos anteriores tanto al modelo como los algoritmos.

## Conceptos básicos

A continuación se presentan conceptos básicos de los problemas de optimización que son utilizados constantemente en el resto de la investigación.

**Definición 1.1.1** Un **problema de optimización** es un par  $(S, f)$  donde  $S$  es un conjunto de soluciones y  $f : S \rightarrow \mathbb{R}$  es una función que asocia un costo a cada solución. El problema consiste en encontrar una solución  $s^* \in S$  que tenga el mejor costo de entre todas las soluciones  $s$  del conjunto  $S$ ; es decir, se requiere que:

$$f(s^*) \leq f(s), \forall s \in S$$

Esta solución se conoce como **óptimo global**.

**Definición 1.1.2** Un **problema de optimización combinatoria** es un problema de optimización donde el conjunto de soluciones  $S$  es finito y donde cada solución es una posible permutación, combinación de componentes en el problema.

También es necesario representar la información en un problema.

**Definición 1.1.3** Las **variables de decisión**  $\{x_1, \dots, x_n\}$  representan la información desconocida en un problema y éstas tienen un dominio de valores.

Como se trabaja sobre un conjunto de soluciones, es necesario definir el espacio en el que éstas se buscan.

**Definición 1.1.4** El **espacio de soluciones** de un problema de optimización, se define como un conjunto de soluciones  $S$  y un conjunto de acciones  $A = a_1, \dots, a_n$  tal que cada acción  $a_i : S \rightarrow S$  transforma una solución en otra.

Al tener una solución actual, es de interés el conjunto de soluciones alrededor de ésta, por lo que, es necesario la definición de vecindario de soluciones.

**Definición 1.1.5** El **vecindario**  $N(s)$  de una solución  $s \in S$  es una función  $N : S \rightarrow 2^S$  que asigna a cada solución un conjunto de soluciones; además, estas soluciones son obtenidas a través de una acción realizada sobre  $s$  y cada solución en  $N(s)$  se denomina **vecino**.

Es importante mencionar la localidad de un problema como una característica relevante al momento de buscar en el espacio de búsqueda.

**Definición 1.1.6** La **localidad** de un problema es la medida sobre que tan bien corresponde la distancia de dos soluciones  $x, y \in S$  con la diferencia de sus costos  $|f(x) - f(y)|$ .

Si la localidad de un problema es alta, significa que para soluciones vecinas al estar a una distancia corta, la diferencia entre sus costos también es corta. Por otro lado, si la localidad de un problema es baja, significa que soluciones vecinas aunque están a una distancia corta, tienen una amplia diferencia entre sus costos. Por lo que, lo mejor es tener una localidad alta para el problema a resolver, pues con una localidad baja, al moverse de entre soluciones vecinas, la búsqueda se puede dificultar al moverse bruscamente de entre regiones de soluciones con costos muy distintos, llegando a tener una búsqueda aleatoria. Para tener una localidad apropiada es importante tener una buena definición de la función de costo y como consecuencia, el espacio de búsqueda no se torna tan complicado de manejar.

Con esto, mientras se realiza una búsqueda sobre el espacio de soluciones, el objetivo es encontrar la mejor solución posible, que es el óptimo global, pero, cuando no se encuentra el global, por lo complicado que es, entonces la búsqueda fácilmente puede quedarse estancada en lo que se conoce como un óptimo local.

**Definición 1.1.7** Dada una solución  $s$  y su vecindario  $N(s)$ ,  $s$  es un **óptimo local** si tiene un mejor costo que todos sus vecinos, esto es:  $f(s) \leq f(s'), \forall s' \in N(s)$ .

En la Figura 1.1 se ejemplifica el óptimo global y local.

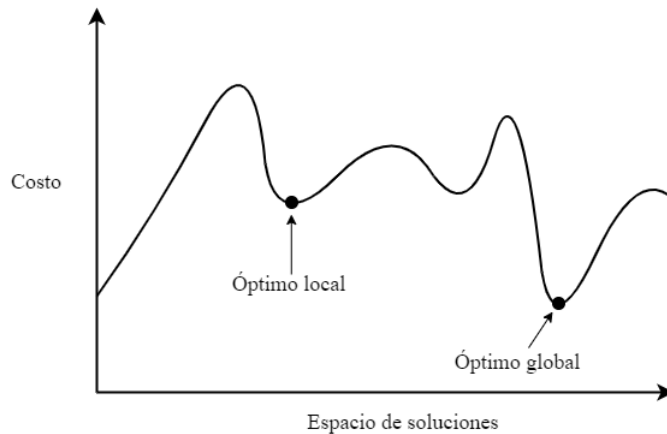


Figura 1.1: Óptimo local y global

Al tratar de obtener la solución para un problema, ésta debe estar bien construida; es decir, que cumple con las restricciones de un problema.

**Definición 1.1.8** Una solución  $s$  es **factible** si ésta satisface las restricciones, condiciones de un problema.

Aunque la solución final que se obtiene para un problema debe ser factible, no significa que únicamente se tenga que buscar sobre soluciones factibles, cuando en el espacio de búsqueda se contemplan soluciones no factibles, es posible diversificar la búsqueda y por tanto, en ocasiones escapar de óptimos locales, sin embargo, se agrega costo en tiempo al evaluar si cada solución es factible o no, pero lo más costoso es que el espacio de búsqueda aumenta de tamaño, como consecuencia, la búsqueda puede tardar en converger a soluciones factibles. Por lo que, la decisión de considerar soluciones no factibles depende de la experiencia en el problema.

## Algoritmos exactos y de aproximación

Los algoritmos exactos garantizan encontrar una solución óptima para cualquier ejemplar del problema, el tiempo de ejecución de éstos depende del tamaño del ejemplar; es decir, mientras más grande sea el ejemplar, más tiempo tarda la ejecución del algoritmo, lo cual resulta una desventaja enorme para problemas de optimización que son NP-Difíciles, pues el tamaño del espacio de soluciones resulta ser exponencial. Por ello, ante este tipo de problemas no se suelen usar algoritmos exactos, aunque, sólo se pueden usar algoritmos exactos para estos problemas cuando el tamaño de los ejemplares es pequeño.

Cuando se trata de resolver un problema de optimización que es NP-Difícil, utilizar un algoritmo exacto para éste es totalmente inadecuado por el tiempo exponencial que tomaría, ante esto, una recomendación es buscar una solución que se acerque a la solución óptima y tarde un tiempo polinomial en obtenerse, para el cual se utiliza un algoritmo de aproximación.

Los algoritmos de aproximación son algoritmos que en tiempo polinomial obtienen soluciones que están dentro de un radio de aproximación  $\epsilon$ , es decir, para cualquier solución  $x_a$  obtenida, su costo es menor o igual a  $\epsilon$  veces el costo de la solución óptima  $x^*$ .

$$\epsilon \geq \max \left( \frac{f(x_a)}{f(x^*)}, \frac{f(x^*)}{f(x_a)} \right)$$

Estos algoritmos se definen como  $\epsilon$ -aproximación. Es evidente que lo mejor es diseñar algoritmos de aproximación con  $\epsilon$  cada vez más pequeñas, con el fin de tener soluciones dentro de un rango más cercano a la solución óptima, pero diseñar algoritmos de aproximación con esta precisión es complicado, por lo que, se opta por usar otras estrategias de aproximación, como se muestra en las secciones de adelante, las heurísticas y metaheurísticas.

## 1.2. Heurísticas

Las heurísticas son técnicas de aproximación basadas en la experiencia y el conocimiento de un determinado problema, éstas proveen soluciones en tiempos relativamente buenos pero no está garantizado que se encuentren soluciones óptimas ni que sean de buena calidad. Las heurísticas son diseñadas según la información que se tiene del problema para explotar características particulares. De aquí que una gran diferencia entre las metaheurísticas y heurísticas es que mientras las heurísticas dependen del problema, las metaheurísticas son de mayor abstracción; es decir, son un marco general que se puede utilizar en distintos problemas.

En el diseño de heurísticas es importante tener un apropiado conocimiento del problema para distinguir los mecanismos que llevan a soluciones de buena calidad y cuales a soluciones de baja calidad.

Se mencionan a las heurísticas como técnicas de aproximación, algo distinto a un algoritmo de aproximación, esto es porque un algoritmo de aproximación tiene una cota sobre la calidad de soluciones, en cambio, como se ha mencionado anteriormente, las heurísticas no garantizan soluciones óptimas ni de buena calidad, por lo que, cuando en una heurística se puede determinar la calidad de las soluciones, ésta se vuelve un algoritmo de aproximación.

Existen dos tipos de heurísticas, las de construcción y las de mejora. Las heurísticas de construcción, como su nombre lo indica, construyen iterativamente una solución y una vez que ésta se generó, se detiene la heurística. Las heurísticas de mejora inician desde una solución e iterativamente aplican reglas de búsqueda que permiten moverse sobre el espacio de soluciones.

## Búsqueda local

La búsqueda local es una heurística clásica que inicia en una solución y en cada iteración actualiza la solución tomando al mejor vecino dentro del vecindario actual y se detiene hasta que encuentra un óptimo local. El objetivo de esta heurística, además de obtener un óptimo local, es tardar el menor tiempo posible a comparación de otras heurísticas, aunque la heurística podría ser lenta si el tamaño del vecindario a revisar en cada iteración es bastante grande, por lo que, un aspecto importante a controlar durante esta búsqueda es el tamaño de los vecindarios a explorar.

En el Algoritmo 1 se presenta un pseudo-código de la búsqueda local.

---

**Algoritmo 1** BÚSQUEDA LOCAL

---

**Entrada:** Solución inicial  $i$ .

**Salida:** Mejor solución encontrada.

```
1: while No se cumpla el criterio de término do
2:   Genera el vecindario  $N(i)$ 
3:   if no se encuentra un vecino que mejore el costo actual then
4:     return  $i$ 
5:   else
6:      $i'$  el mejor vecino en  $N(i)$ 
7:      $i \leftarrow i'$ 
8:   end if
9: end while
10: return  $i$ 
```

---

De lo anterior, se tienen las siguientes estrategias para seleccionar vecinos.

- **Mejor de todos.** En esta estrategia se selecciona al mejor vecino de cada vecindario, lo cual provoca que la búsqueda en cada vecindario sea exhaustiva y por tanto, esta exploración consume bastante tiempo.
- **El primero en mejorar.** En esta estrategia se toma al primer vecino que tenga mejor costo que la solución actual, aunque esta estrategia podría terminar evaluando todo el vecindario.
- **Selección aleatoria.** Consiste en tomar un conjunto aleatorio de vecinos y de entre estos tomar aquel que mejora el costo de la solución actual.

### 1.3. Metaheurísticas

Encontrar una solución óptima en los problemas de optimización combinatoria que son NP-Difíciles es hasta ahora algo imposible, pero en la práctica obtener buenas soluciones es lo ideal y esto se logra usando heurísticas y metaheurísticas.

El término metaheurística fue propuesto por Glover en 1986, las palabras ‘meta’ y ‘heurística’ tienen su origen en la literatura de la antigua Grecia, ‘meta’ significa ‘más allá’ y ‘heurística’ es ‘el arte de descubrir nuevas estrategias’.

Las metaheurísticas son metodologías de alto nivel que guían heurísticas subordinadas para escapar de óptimos locales aplicando fases de intensificación y diversificación en la búsqueda. El objetivo de la intensificación es mejorar la calidad de las soluciones obtenidas en una región del espacio de soluciones, mientras la diversificación es explorar diversas regiones del espacio de soluciones, estos dos conceptos son importantes para el diseño de las metaheurísticas. Es común que las metaheurísticas imiten estrategias exitosas que se encuentran en fenómenos físicos, de la naturaleza y sociales. Una característica relevante de las metaheurísticas es que son independientes del problema y por tanto, se pueden aplicar a diversos problemas. Las metaheurísticas realizan la búsqueda de soluciones mientras no se cumpla un criterio de término como pueden ser; un número de iteraciones, algún costo alcanzado en las soluciones, si las soluciones ya no mejoran durante un lapso de tiempo, entre otros. Se tienen dos principales clases de metaheurísticas, las metaheurísticas basadas en una solución y las metaheurísticas basadas en población.

El material de esta sección está basado en la literatura clásica del tema [16, 17, 20, 21].

#### Cuando usar una metaheurística

Al resolver un problema de optimización combinatoria, lo primero que se debe realizar es un análisis de su complejidad, muchos de los problemas de optimización combinatoria son NP-Difíciles, para estos problemas, la manera en la que es posible obtener soluciones buenas es mediante metaheurísticas, aunque para algunos problemas que pertenecen a la clase  $P$  es posible que algunos algoritmos de tiempo polinomial tomen tiempos considerablemente altos y por tanto, en la práctica no sea conveniente usarlos, por ejemplo, un algoritmo donde su complejidad en tiempo sea  $O(n^{4000})$ , por lo que, además de conocer la complejidad del problema, también es necesario tomar en cuenta el tiempo que puede tomar en la práctica resolver el problema aun si se tiene un algoritmo polinomial que lo resuelve.

Estas son algunas de las principales características en los problemas de optimización que justifican el uso de metaheurísticas:

- Para problemas en  $P$  donde existen ejemplares de un tamaño considerablemente grande para los cuales el tiempo que toma resolver estos usando un algoritmo polinomial toma un tiempo que en la práctica no es adecuado.
- Para problemas en  $P$  donde se requieren resultados al instante y con pocas decisiones, es decir, aunque se tiene un algoritmo polinomial, puede que una metaheurística tome menos tiempo.
- Para problemas que sean NP-Difíciles donde las soluciones obtenidas por algoritmos de aproximación no sean suficientes.
- Para problemas de optimización donde el costo computacional de evaluar las funciones de costo sea alto.

- Problemas para los cuales debido al tamaño de los ejemplares no se pueden resolver de manera exhaustiva.

## Elementos de las metaheurísticas

Para el diseño de las metaheurísticas es relevante especificar los elementos que son fundamentales en la estructura de éstas y estos elementos son:

- Representación de las soluciones.
- Función objetivo.
- Soluciones iniciales.
- Estrategias de búsqueda.
- Estrategias de intensificación y diversificación.

Adelante se describen estos elementos.

### Representación de las soluciones

Al utilizar una metaheurística, primero se necesita una representación o codificación de las soluciones del problema, donde las soluciones se representan mediante una estructura a la cual se le pueden aplicar operaciones de transformación y por tanto, tener nuevas soluciones.

De acuerdo a su estructura, existen dos clases principales de representaciones, las lineales y no lineales. A continuación se describen ambas.

#### Representaciones lineales

Estas representaciones están compuestas de cadenas de símbolos de un alfabeto dado, a continuación algunas representaciones lineales.

- Las representaciones **binarias** donde se utilizan valores binarios para las variables de decisión, donde las soluciones se suelen representar por vectores de valores binarios.
- Las representaciones que usan **permutaciones** de los elementos del problema y por tanto, las soluciones se representan en vectores de elementos del problema intercambiados.
- Las representaciones **numéricas**, donde las soluciones se representan en vectores de números.

#### Representaciones no lineales

Este tipo de representaciones utilizan estructuras más complejas y se basan bastante en gráficas como los árboles, los cuales se utilizan en los problemas de optimización donde existe una jerarquía entre los elementos de la solución.

### Función objetivo

La función objetivo  $f : S \rightarrow \mathbb{R}$  asocia cada solución con un número real que describe la calidad de la solución, además de que implica un orden total en las soluciones del espacio.

Cuando se define la función objetivo se debe asegurar que las soluciones óptimas deben tener los mejores resultados según la función, por otro lado, las soluciones de baja calidad deben tener un resultado bajo. Pero lo más importante de la función objetivo es que debe estar bien definida para que los métodos de búsqueda recorran eficientemente el espacio de soluciones.

## Soluciones iniciales

Se tienen dos casos para las distintas metaheurísticas, crear una solución inicial o una población de soluciones iniciales, para crear una solución inicial basta con un algoritmo exacto, de aproximación, una heurística, alguna manera trivial de crear soluciones o bien, si no se conoce suficiente información sobre la calidad de las buenas y malas soluciones, entonces se crean soluciones de manera aleatoria. Para crear una población de soluciones iniciales, lo recomendado es obtener una población de soluciones con suficiente diversidad, esto es, las soluciones sean lo suficientemente distintas unas de otras, pues si se cuenta con una diversidad baja, al generar una nueva población de soluciones, éstas pueden ser de baja calidad y similares a las soluciones anteriores, por lo que, la búsqueda se puede quedar estancada en una región con soluciones pobres, una manera de obtener diversidad en la población de soluciones es que las soluciones sean generadas de manera aleatoria, para tener mayor diversidad y abarcar más regiones del espacio de soluciones.

## Estrategias de búsqueda

Una vez que se tienen soluciones iniciales como punto de inicio para la búsqueda en el espacio de soluciones, es necesario establecer como se dirigen las metaheurísticas basadas en una solución y las que se basan en población. Existen dos estrategias principales para la búsqueda, la búsqueda basada en métodos de recombinación y la búsqueda basada en métodos de búsqueda local. A continuación se muestran las dos estrategias.

### Estrategia basada en métodos de búsqueda local

El uso de la búsqueda local es una de las principales características en bastantes metaheurísticas basadas en una solución, ésta se utiliza para que en cada iteración se modifique la solución actual hacia un nuevo vecino que no cambie tanto a comparación de ésta (es decir, es el resultado de una modificación a la solución actual), la consecuencia de usar búsqueda local es que la búsqueda sobre el espacio de soluciones se realiza mediante pasos cortos, lo cual tarda bastante, pero tiene la ventaja de que al acercarse la búsqueda en una región con soluciones de buena calidad, es posible que en algún momento se logre encontrar con alguna de estas soluciones, por otro lado, si se realizaran pasos largos, entonces la búsqueda empieza a moverse de una región a otra, incrementando la aleatoriedad, lo cual provoca que no se revisen detalladamente regiones con buenas soluciones. Para lograr diversificar la búsqueda en esta estrategia es usual que se recorran soluciones de distintas regiones del espacio de búsqueda utilizando los mecanismos de la búsqueda local para escapar de óptimos locales, como aceptar peores soluciones, soluciones aleatorias. En cuanto a la intensificación, se puede aceptar de entre las mejores soluciones de cada vecindario o una búsqueda exhaustiva, que claro, se toma el riesgo de caer tempranamente en óptimos locales.

### Estrategia basada en métodos de recombinación

En algunas metaheurísticas basadas en población se utilizan los métodos de recombinación, estos métodos funcionan asumiendo que los problemas se pueden descomponer en problemas más pequeños, tales que solucionarlos es más fácil y una vez que se tienen las soluciones, lo siguiente es unir sus características para construir una nueva solución. Al recombinar soluciones es importante tomar las características más importantes de la solución y que estas características puedan mantenerse en las soluciones descendientes, por lo que, más que poder descomponer los problemas, lo relevante es la correcta mezcla de soluciones. Cuando se utilizan métodos de recombinación, un criterio de término es cuando la población de soluciones converge hacia soluciones idénticas, por lo que, cuando se utiliza la recombinación y las soluciones convergen muy pronto es porque la búsqueda está en un óptimo local.

Algunas maneras para incrementar la diversidad de soluciones es teniendo mayor número de soluciones en la población y recombinar soluciones que sean lo más distintas posibles, otra opción es



reemplazar soluciones similares con peores soluciones o bien agregar a la población estas soluciones.

## Estrategias de intensificación y diversificación

En la búsqueda de soluciones, además de utilizar estrategias de búsqueda adecuadamente, es importante utilizar mecanismos que puedan modificar la búsqueda, con lo cual, se logra buscar de una manera más eficiente, para lograr esto, se utilizan estrategias de intensificación y diversificación.

### Intensificación

Las estrategias de intensificación se enfocan en tomar decisiones que guían la búsqueda sobre regiones del espacio con soluciones que puedan ser de buena calidad. En algunas estrategias de intensificación, se enfoca la búsqueda sobre las regiones con las mejores soluciones encontradas y de éstas, se toman sus características para crear nuevas soluciones. Sin embargo, el riesgo de enfocar la búsqueda sobre alguna región es encontrar óptimos locales, para esto, se utiliza la diversificación.

### Diversificación

A diferencia de la intensificación, la diversificación se utiliza para explorar regiones no visitadas de la búsqueda o bien, tomar características de algunas soluciones y crear nuevas soluciones que no hayan sido exploradas antes. Usando estas estrategias es cómo se intenta escapar de regiones con soluciones pobres y que por tanto, conducen a óptimos locales.

En las metaheurísticas se alternan las estrategias de intensificación y de diversificación, pues cuando la búsqueda se enfoca en alguna región con el objetivo de encontrar la mejor solución y cuando ya no mejora la búsqueda, lo siguiente es diversificar, tomando otra región del espacio para volver a buscar soluciones satisfactorias.

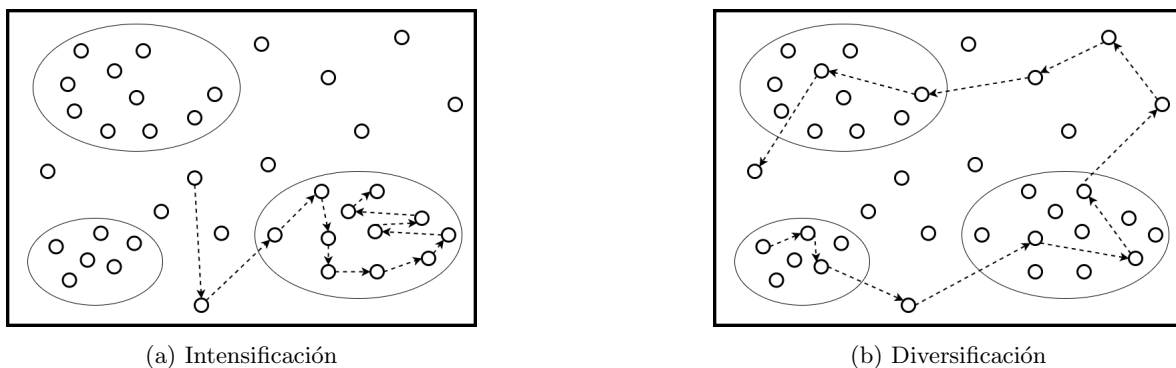


Figura 1.2: Representación de intensificación y diversificación

## Metaheurísticas basadas en una solución

Las metaheurísticas basadas en una solución inician con una solución inicial, la cual se va modificando, estas metaheurísticas también se les refiere como metaheurísticas de trayectoria, por la razón de recorrer el espacio de búsqueda modificando una solución a través de procesos iterativos, lo cual genera un camino entre una solución inicial y una solución final.

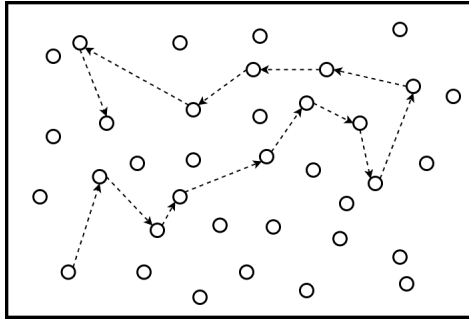


Figura 1.3: Representación de la trayectoria de una solución

### Esquema general

Estas metaheurísticas utilizan mecanismos de reemplazo y generación de soluciones, teniendo dos fases, la de generación y la de reemplazo.

En la fase de generación, dada una solución actual  $s$  se obtiene un conjunto  $C(s)$  de soluciones creadas a través de métodos locales.

En la fase de reemplazo, se selecciona una solución del conjunto  $C(s)$  para reemplazar a  $s$ .

Estas dos fases se repiten según el criterio de término, adicionalmente las metaheurísticas pueden tener o no memoria.

En el Algoritmo 2 se presenta un pseudo-código que ilustra el esquema general.

---

#### Algoritmo 2 ESQUEMA GENERAL DE LAS METAHEURÍSTICAS BASADAS EN UNA SOLUCIÓN

---

**Entrada:** Solución inicial  $s$ .

**Salida:** Mejor solución encontrada.

```

1:  $t \leftarrow 0$ 
2: while No se cumpla el criterio de término do
3:   Genera un conjunto de soluciones  $C(s)$ 
4:    $s \leftarrow \text{Selecciona}(C(s))$ 
5:    $t \leftarrow t + 1$ 
6:   //Si la metaheurística usa memoria
7:   Actualiza memoria
8: end while
9: return  $s$ 

```

---

Dado el esquema anterior, estas metaheurísticas se basan en moverse a través de vecindarios de soluciones modificando la solución actual, algo importante a considerar en los vecindarios es su tamaño, revisar vecindarios de un tamaño grande puede tomar bastante tiempo en cuanto a generar y evaluar cada solución vecina, además, dependiendo de la heurística subordinada utilizada, se puede caer en una región con óptimos locales, por tanto, es necesario diseñar buenas estrategias cuando el vecindario a buscar es grande.

## Metaheurísticas basadas en población

Las metaheurísticas basadas en población inician con una población de soluciones, de las cuales, seleccionan o con algunos métodos de recombinación logran unir las para obtener una nueva población de soluciones, aunque también pueden obtenerse nuevas poblaciones de soluciones realizando métodos de búsqueda local sobre cada solución de la población actual, todo esto se realiza mientras no se cumpla un criterio de término.

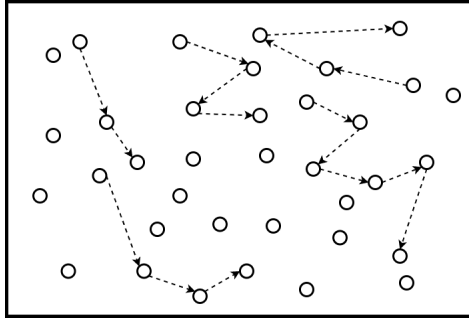


Figura 1.4: Representación de una población de soluciones

### Esquema general

Estas metaheurísticas al igual que las metaheurísticas basadas en una solución, tienen dos fases, la de generación y la de reemplazo.

Inicia con una población inicial de soluciones, en la fase de generación, crea una nueva población.

Para la fase de reemplazo, se seleccionan soluciones de entre la población actual y la nueva población obtenida en la fase de generación o bien, pueden recombinarse las soluciones para una nueva población o también, aplicar métodos de búsqueda local a cada solución de la población. Cual sea la manera en que se reemplaza la población, estas fases se repiten iterativamente hasta que se cumpla algún criterio de término, también, las metaheurísticas pueden tener o no memoria.

En el Algoritmo 3 se presenta un pseudo-código que ilustra el esquema general.

---

#### Algoritmo 3 ESQUEMA GENERAL DE LAS METAHEURÍSTICAS BASADAS EN POBLACIÓN

---

**Entrada:** Población inicial  $P_0$ .

**Salida:** Mejor solución encontrada.

```
1:  $t \leftarrow 0$ 
2:  $P_t = P_0$ 
3: while No se cumpla el criterio de término do
4:   Genera una población de soluciones  $P'_t$ 
5:    $P_{t+1} = \text{SeleccionaPoblacion}(P_t \cup P'_t)$ 
6:    $t \leftarrow t + 1$ 
7:   //Si la metaheurística usa memoria
8:   Actualiza memoria
9: end while
10: return MejorSolucion( $P_{t+1}$ )
```

---

## 1.4. Problemas de planeación

La planeación de cualquier actividad, operación o trabajo es importante en cualquier sector industrial, laboral, o cualquier ambiente donde sea necesario optimizar los tiempos o ganancias, ya que al realizar esto, se garantiza la sostenibilidad de estas industrias dentro del mercado competitivo. Además, es necesario utilizar los recursos disponibles de la manera más eficiente posible.

La planeación se enfoca en la correcta asignación de recursos para las actividades de un proyecto durante cualquier tiempo. Los problemas de planeación de proyectos están compuestos principalmente de actividades, recursos y relaciones de precedencia, para estos problemas se busca optimizar alguna función objetivo.

Para llegar a optimizar ciertos objetivos, se tienen problemas a enfrentar donde se involucra la planeación, es común que estos problemas mayormente se traten sobre máquinas, aunque, los problemas de planeación de máquinas están muy ligados a los problemas de planeación de proyectos.

### Planeación de proyectos

La planeación de proyectos está integrada por los componentes del proyecto, los objetivos de la planeación, la representación del proyecto, además de la representación de la planeación. A continuación se describen brevemente.

#### Componentes de un proyecto

Un proyecto se compone de actividades, que pueden ser tareas, operaciones, trabajos, entre otros. Cada actividad tiene un tiempo de duración, el cual es el tiempo que toma realizarla dentro del proyecto.

Las relaciones de precedencia entre actividades, definen el orden en el que las actividades se tienen que realizar dentro de la planeación del proyecto, por lo que, cada actividad debe iniciar antes que otras y cuando esta finalice, otras actividades pueden ser realizadas.

En cuanto a los recursos, estos son utilizados por las actividades, existen dos clasificaciones principales, los renovables y los no renovables.

Los recursos renovables pueden ser usados en cualquier tiempo de la planeación del proyecto siempre que no se exceda su disponibilidad en cada tiempo, algunos ejemplos son: máquinas, personas, procesadores, entre otros.

Por otro lado, los recursos no renovables, una vez que son completamente utilizados, ya no pueden volver a usarse, un ejemplo es el capital económico invertido en un proyecto.

#### Objetivos de la planeación de proyectos

Se tienen dos principales objetivos a optimizar dentro de la planeación de proyectos.

El objetivo más común a optimizar dentro de un proyecto es el tiempo entre el inicio y el final del proyecto, en el cual se trata de minimizar el tiempo en el que terminan las actividades en un proyecto.

Cuando existe flujo de dinero o capital en un proyecto, es necesario optimizar el *valor actual neto* (VAN), que indica la viabilidad económica del proyecto, éste es calculado mediante una fórmula ma-

temática donde se relaciona la inversión inicial, los ingresos y los egresos económicos. Para este tipo de proyectos, las actividades tienen un costo por iniciar y en cuanto finalizan, se obtiene alguna ganancia, por lo que se requiere planear las actividades obteniendo más ganancia que gasto.

### Representación de un proyecto

Existen dos maneras de representar un proyecto en una gráfica, las actividades en nodos AON<sup>1</sup> y las actividades en arcos AOA<sup>2</sup>.

Para las actividades en nodos AON, las actividades y toda la información referente a éstas, están almacenados en los nodos de la gráfica, en cuanto a los arcos, representan las relaciones de precedencia entre actividades.

Por ejemplo, en la Figura 1.5 se muestra una gráfica representando a las actividades en un proyecto, dentro del nodo se encuentra la actividad y arriba de ésta se encuentra la duración de la actividad y por debajo las unidades de recurso que utiliza la actividad.

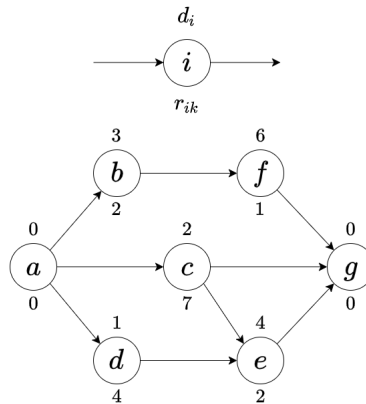


Figura 1.5: Representación AON

Para las actividades en arcos AOA, los nodos representan eventos y los arcos representan a las actividades. Esta representación es utilizada cuando se tienen flujos de dinero en un proyecto y por tanto, se tienen que representar los ingresos y egresos como eventos en los nodos de la gráfica, con lo cual, las actividades junto a su duración van en los arcos.

Por ejemplo, en la Figura 1.6 se muestra la representación AOA para un proyecto de actividades  $i = a, b, c, d, e$ , con duración  $p_i$  sobre las aristas,  $f_i^-$  representa el gasto económico y  $f_i^+$  la ganancia económica. Además, la gráfica contiene aristas punteadas que son ficticias; esto es, no contienen información de una actividad, pero son útiles para tener más eventos del proyecto y conectar otras actividades.

<sup>1</sup>Siglas en inglés de 'Activity-On-Node'

<sup>2</sup>Siglas en inglés de 'Activity-On-Arc'

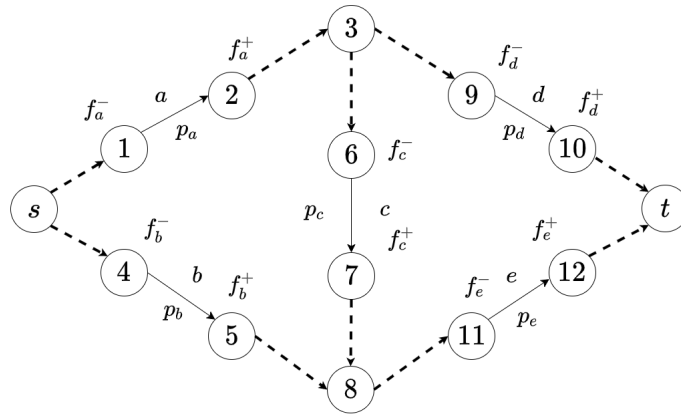


Figura 1.6: Representación AOA

### Representación de una planeación

Las planeaciones son representadas mediante diagramas de Gantt. Por ejemplo, en la Figura 1.7 se muestra una planeación de siete actividades  $i = 1, 2, 3, 4, 5, 6, 7$  usando tres máquinas  $M_j = 1, 2, 3$ , donde el tiempo que toma la planeación es de 10 unidades.

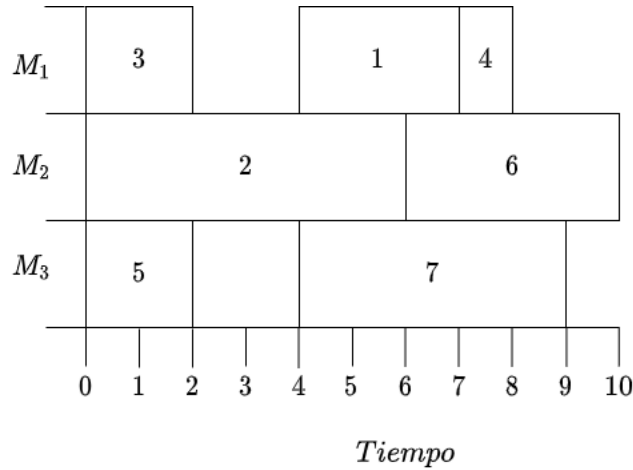


Figura 1.7: Representación de una planeación

### Problemas de planeación en máquinas

A continuación se muestran algunos de los problemas de planeación en máquinas más conocidos.

#### Problema de planeación de una máquina

El problema de planeación de una máquina<sup>3</sup>, es el más simple ejemplo de un problema de planeación, donde el único recurso que se tiene es una máquina, para  $n$  actividades  $j = 1, \dots, n$  o tareas a

<sup>3</sup>Del inglés 'Single-machine scheduling problem'

procesar, donde para cada actividad se tiene una duración  $p_j$ .

En la Figura 1.8 se muestra la planeación de cinco actividades en una máquina.

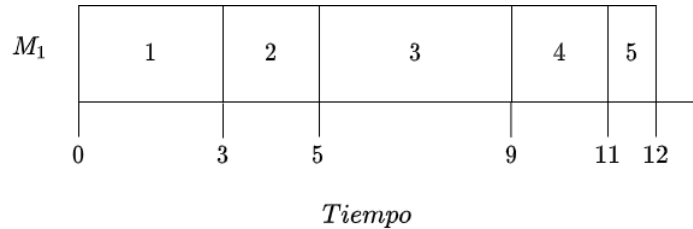


Figura 1.8: Planeación de una máquina

### Problema de planeación de máquinas en paralelo

Para el problema de planeación de máquinas en paralelo<sup>4</sup>, en lugar de tener una máquina como en la planeación anterior, se tienen  $m$  máquinas  $M_1, \dots, M_m$  que son idénticas; esto es, cualquier actividad dada puede ser realizada por cualquier máquina.

En la Figura 1.9 se muestra la planeación de doce actividades en tres máquinas.

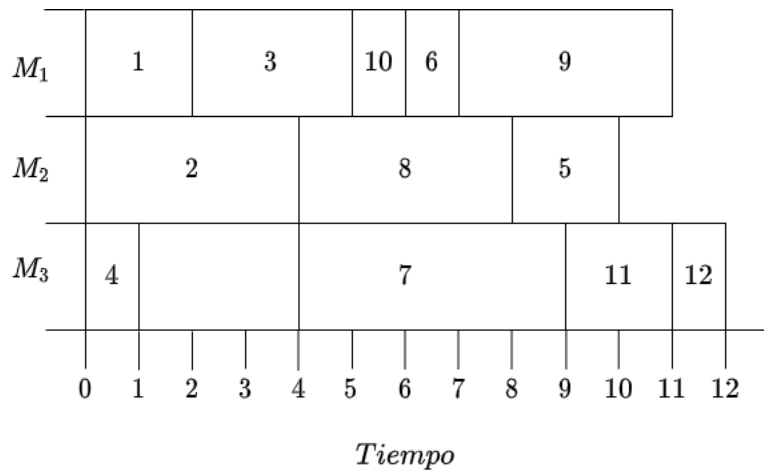


Figura 1.9: Planeación de máquinas en paralelo

### Problema de planeación de máquinas con operaciones generalizado

En el problema de planeación de máquinas con operaciones generalizado<sup>5</sup>, las actividades ahora consisten de múltiples operaciones, se tienen  $m$  máquinas  $M_1, \dots, M_m$ ,  $n$  actividades, donde para cada actividad  $j$  se tienen  $n_j$  operaciones  $O_{1j}, \dots, O_{n_j j}$ ; donde  $j = 1, \dots, n$ . Además, existe una relación de

<sup>4</sup>Del inglés 'Parallel machine scheduling problem'

<sup>5</sup>Del inglés 'Job-shop scheduling problem'

precedencia entre las operaciones de cada actividad, por lo que, dos operaciones de la misma actividad no pueden ser realizadas al mismo tiempo y las máquinas únicamente pueden realizar una operación en cada tiempo.

En la Figura 1.10 se muestra la planeación de las doce actividades de la planeación anterior, sólo que ahora se divide cada actividad en operaciones.

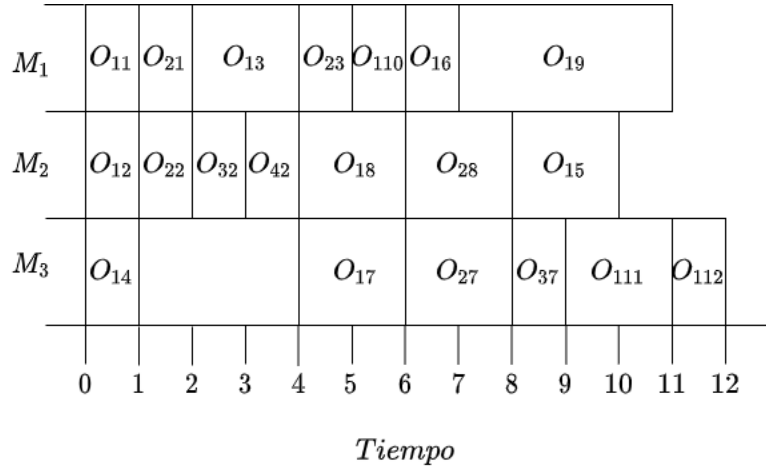


Figura 1.10: Planeación de máquinas con operaciones generalizado

### Problema de planeación de máquinas en flujo

En el problema de planeación de máquinas en flujo<sup>6</sup>, además de tener las características de la planeación anterior, se tienen tantas operaciones por cada actividad como máquinas  $n_j = m$ , además, cada operación  $O_{ij}$  de cada actividad  $j$  debe ser realizada específicamente por una máquina  $M_i$  y también existen relaciones de precedencia entre las operaciones de cada actividad.

En la Figura 1.11 se muestra una planeación donde las operaciones  $O_{11}, O_{12}, O_{13}, O_{14}$  son realizadas por la máquina  $M_1$ , en cuanto a la máquina  $M_2$ , esta realiza a las operaciones  $O_{21}, O_{22}, O_{23}, O_{24}$  y algo similar sucede para las máquinas  $M_3$  y  $M_4$ .

<sup>6</sup>Del inglés 'Flow-shop scheduling problem'



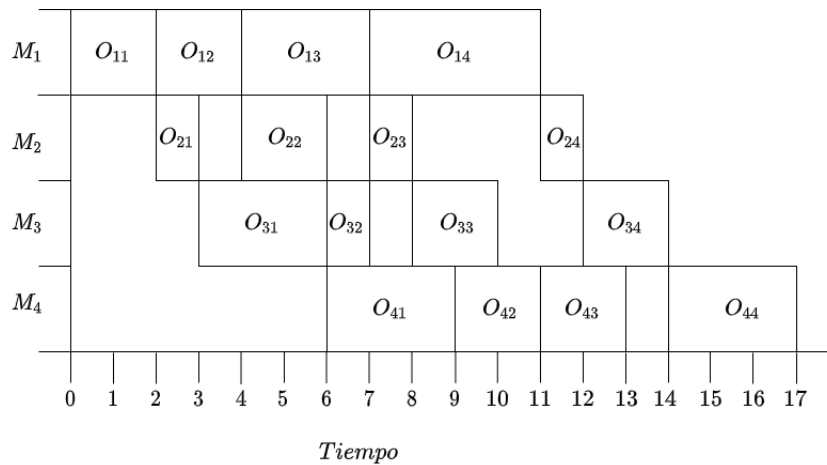


Figura 1.11: Planeación de máquinas en flujo

### Problema de planeación de máquinas abierto

El problema de planeación de máquinas abierto<sup>7</sup>, es similar a la planeación de máquinas en flujo, pero en este caso no existe una relación de precedencia entre las operaciones de las actividades.

En la Figura 1.12 se muestra un ejemplo de esta planeación tomando las operaciones de la planeación anterior, solo que, las operaciones de la actividad 1 están en un orden distinto, que es  $O_{11}, O_{31}, O_{21}, O_{41}$ , lo mismo sucede con las otras actividades, de esta manera se puede elegir en que orden planear las operaciones. Dada la libertad de no tener relaciones de precedencia entre las operaciones, se puede tener una planeación con bastantes tiempos en los que no se realizan algunas operaciones, teniendo recursos sin utilizar.

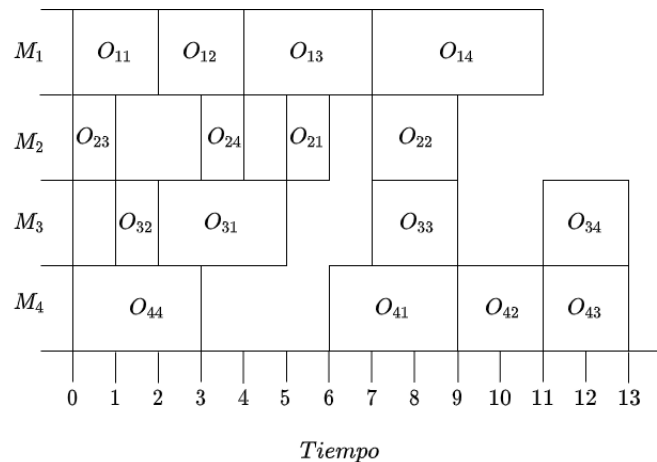


Figura 1.12: Planeación de máquinas abierto

<sup>7</sup>Del inglés ‘Open-shop scheduling problem’

### Problema de planeación de tareas en múltiples máquinas

En el problema de planeación de tareas en múltiples máquinas<sup>8</sup>, se tienen  $n$  actividades  $j = 1, \dots, n$  y  $m$  máquinas  $M_1, \dots, M_m$ , para cada actividad, ésta puede ser realizada al mismo tiempo por un subconjunto de máquinas.

En la Figura 1.13 se muestra la planeación de ocho actividades realizadas en múltiples máquinas, por ejemplo, la Actividad 1 está siendo realizada por las máquinas  $M_3$  y  $M_4$  a partir del tiempo cero. De la misma forma para la Actividad 5, que es realizada por las máquinas  $M_2$  y  $M_3$  a partir del tiempo dos. Este tipo de planeación es conveniente para las actividades que pueden ser realizadas al mismo tiempo por más de una máquina.

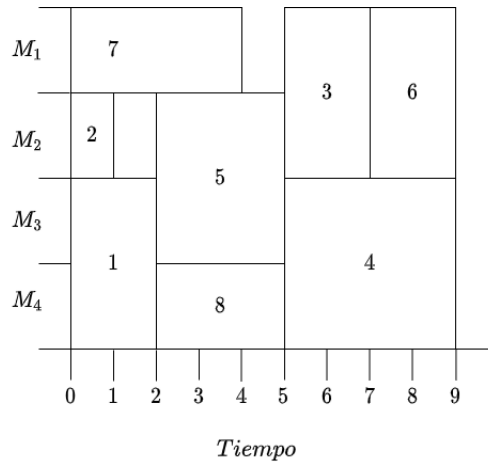


Figura 1.13: Planeación de tareas en múltiples máquinas

<sup>8</sup>Del inglés ‘*Multi-processor task scheduling problem*’

## 1.5. Planeación de proyectos sujeto a recursos (RCPSP)

El problema clásico de planeación de proyectos sujeto a recursos (RCPSP), consiste en minimizar el tiempo que toma terminar un conjunto de actividades que están sujetas a la utilización de recursos renovables y estas actividades no se pueden interrumpir.

De acuerdo a Blazewicz [1], este problema es NP-Completo. El material que se presenta en esta sección está basado en el artículo presentado por Demeuleester, Herroelen [2].

### Descripción del problema

Dado un proyecto compuesto de un conjunto  $K$  de recursos renovables, un conjunto  $H$  de restricciones y  $n$  actividades, tal que existen dos actividades ficticias, la Actividad 1 y la Actividad  $n$  que indican el inicio y el término del proyecto, respectivamente; la Actividad 1 debe ser predecesora de las actividades iniciales del proyecto, en cuanto a la Actividad  $n$ , debe ser sucesora de las últimas actividades del proyecto. Se tiene que cada actividad está restringida a iniciar después de que las actividades anteriores, según el conjunto  $H$ , hayan concluido.

En el conjunto  $K$  se encuentran los recursos renovables del proyecto, donde cada recurso renovable  $k$  puede ser utilizado en cualquier tiempo de la planeación siempre y cuando la demanda por parte de las actividades no exceda su disponibilidad, donde las unidades que se pueden ocupar de un recurso renovable  $k$  se denotan como  $a_k$ .

Para cada actividad se define:

- La función de costo  $f_i$  que indica el tiempo en el que finaliza la actividad.
- La duración  $d_i$  de la actividad  $i$ .
- Las unidades  $r_{ik}$  de un recurso renovable  $k \in K$  que la actividad  $i$  ocupa durante un intervalo de tiempo.
- Si la actividad  $i$  debe realizarse antes de la actividad  $j$ , se tiene que  $(i, j) \in H$ .

Se tienen las siguientes condiciones que debe cumplir una planeación factible.

- Ninguna actividad puede ser interrumpida una vez que ha iniciado.
- Para cada restricción  $(i, j) \in H$ , el tiempo en el que termina la actividad  $i$  debe ser a lo más el tiempo en el que empieza la actividad  $j$ .

$$f_i \leq f_j - d_j, \forall (i, j) \in H$$

- Dado un conjunto  $S_t$  de actividades que se realizan en el intervalo de tiempo  $(t - 1, t]$ , se debe cumplir que la cantidad de unidades a utilizar de un recurso renovable  $k$  no sea mayor a su disponibilidad, esto es:

$$\sum_{i \in S_t} r_{ik} \leq a_k, \forall k \in K, t = 1, \dots, f_n$$

Claramente, las actividades ficticias 1 y  $n$  no requieren alguna duración ni recursos.

El objetivo principal es minimizar el tiempo que toma realizar las actividades del proyecto, esto es, minimizar el tiempo en que finaliza la última actividad, lo cual es  $f_n$ .

Formalmente, el problema se puede formular como:

$$\text{Minimizar } f_n \tag{1.1}$$

Sujeto a

$$f_i \leq f_j - d_j, \forall (i, j) \in H \tag{1.2}$$

$$f_1 = 0 \tag{1.3}$$

$$\sum_{i \in S_t} r_{ik} \leq a_k, \forall k \in K, t = 1, \dots, f_n \tag{1.4}$$

Donde:

$n$  : el número de actividades en el proyecto.

$K$  : el conjunto de recursos renovables.

$f_i$  : el tiempo en el que la actividad  $i$  finaliza.

$d_i$  : la duración de la actividad  $i$ .

$H$  : el conjunto de pares que indican la precedencia de actividades.

$S_t$  : el conjunto de actividades en progreso en el intervalo  $(t - 1, t]$ .

$r_{ik}$  : la cantidad de unidades que la actividad  $i$  utiliza del recurso  $k$ .

$a_k$  : las unidades que se pueden ocupar del recurso  $k$ .

La función objetivo, de acuerdo a 1.1, representa la minimización del tiempo en que finaliza la última actividad.

El conjunto de restricciones 1.2 asegura que toda actividad debe ser planeada después de que todas las actividades que le preceden sean planeadas.

De acuerdo a 1.3, la actividad ficticia 1 finaliza en el tiempo 0.

El conjunto de restricciones 1.4 indica que en cualquier intervalo de tiempo  $(t - 1, t]$ , la demanda de unidades por parte de las actividades que utilizan un recurso  $k$  no debe superar su disponibilidad.

En la Figura 1.14 se tiene un ejemplo de ocho actividades (recordando que las actividades 1 y 8 son ficticias), con un recurso renovable cuya disponibilidad es de dos unidades.

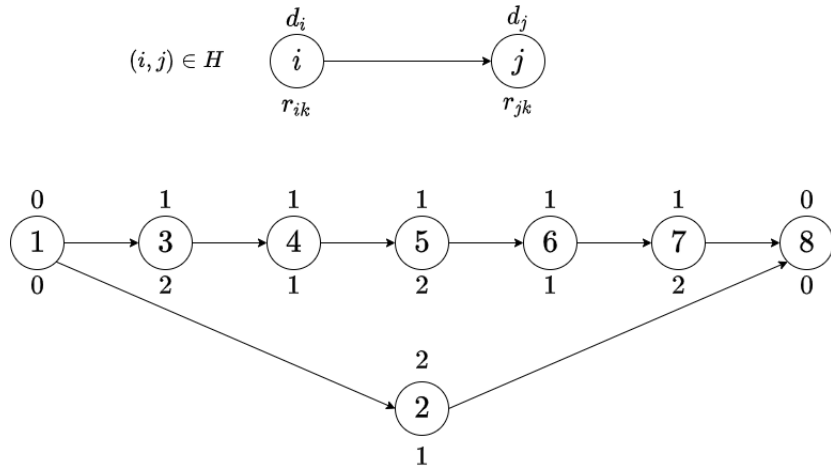


Figura 1.14: Proyecto de ocho actividades.

Ahora, en la Figura 1.15 se presenta una planeación óptima de las actividades sin interrupciones, en el eje X se representa el tiempo y en el eje Y la cantidad de recursos usados por las actividades a través del tiempo, se puede ver que las actividades ficticias 1 y 8 no aparecen porque no consumen recursos ni tiempo, en cambio, la actividad 3 se planea en el tiempo 0 utilizando dos unidades de recursos, después, le sigue la actividad 4 en el tiempo 1 utilizando una unidad de recurso y continuando así hasta la última actividad en planearse que es la actividad 2 que tarda dos unidades de tiempo y utiliza una unidad de recurso. Finalmente, la actividad ficticia 8 aunque no aparece, tiene como tiempo de planeación el tiempo en que termina la última actividad, este tiempo es 7 de acuerdo a la actividad 2, por lo que, esta actividad ficticia denota la duración del proyecto.

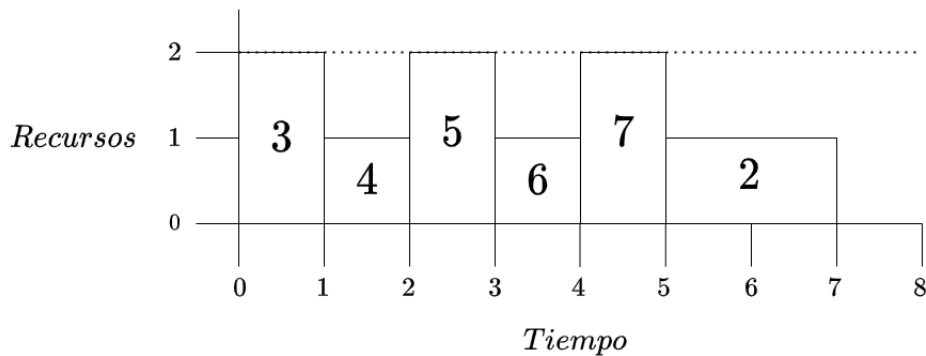


Figura 1.15: Planeación óptima sin interrupciones.

## Capítulo 2

# Planeación de proyectos sujeto a recursos con interrupciones (PRCPSP)

El problema de planeación de proyectos sujeto a recursos con interrupciones (PRCPSP), consiste en minimizar el tiempo que toma terminar un conjunto de actividades que están sujetas a la utilización de recursos renovables y estas actividades, a diferencia del RCPSP, se pueden interrumpir.

El problema RCPSP es un subcaso de PRCPSP en el que el número de interrupciones es 0, por lo que PRCPSP es NP-Completo.

El material que se presenta en este capítulo está basado en el artículo presentado por Demeuleester, Herroelen [2].

### 2.1. Descripción del problema

Dado un proyecto compuesto de un conjunto  $K$  de recursos renovables, un conjunto  $H$  de restricciones y  $n$  actividades, tal que existen dos actividades ficticias, la Actividad 1 y la Actividad  $n$  que indican el inicio y el término del proyecto, respectivamente; la Actividad 1 debe ser predecesora de las actividades iniciales del proyecto, en cuanto a la Actividad  $n$ , debe ser sucesora de las últimas actividades del proyecto. Se tiene que cada actividad está restringida a iniciar después de que las actividades anteriores, según el conjunto  $H$ , hayan concluido y las actividades pueden ser interrumpidas un número entero  $m$  de veces.

Para cada actividad, además, de las definiciones en el problema anterior, se tiene que:

- La duración  $d_i$  se divide en a lo más  $m + 1$  partes, cada una con una duración entera, pero, si esto no se puede realizar, entonces la duración se divide en partes de duración uno, a cada parte se le denomina subactividad, de tal manera que  $f_{i,j}$  es el tiempo en que finaliza la  $j$ -ésima subactividad con  $j = 1, \dots, m + 1$ , por lo que, el tiempo en el que la actividad finaliza es  $f_{i,m+1}$ .

Se tienen las siguientes condiciones que debe cumplir una planeación factible:

- Para cada restricción  $(i, j) \in H$ , el tiempo en el que finaliza la última subactividad que conforma la actividad  $i$  debe ser a lo más el tiempo en el que empieza la primer subactividad de la actividad  $j$ ; esto es:

$$f_{i,m+1} \leq f_{j,0}, \forall (i,j) \in H$$

- Las subactividades deben terminar en orden, lo que significa:

$$f_{i,j-1} + 1 \leq f_{i,j}, \quad j = 1, \dots, m + 1$$

- Dado un conjunto  $S_t$  de actividades que se realizan en el intervalo de tiempo  $(t - 1, t]$ , se debe cumplir que la cantidad de unidades que se utilizan de un recurso renovable  $k$  no sea mayor a su disponibilidad, esto es:

$$\sum_{i \in S_t} r_{ik} \leq a_k, \quad \forall k \in K, \quad t = 1, \dots, f_{n,0}$$

El objetivo principal es minimizar el tiempo que toma realizar todas las subactividades de las actividades en el proyecto, esto es, minimizar el tiempo en que termina la última actividad que se compone de una única subactividad, el cual es  $f_{n,0}$ .

Formalmente, el problema se puede formular como:

$$\text{Minimizar } f_{n,0} \tag{2.1}$$

$$f_{i,m+1} \leq f_{j,0}, \quad \forall (i,j) \in H \tag{2.2}$$

$$f_{i,j-1} + 1 \leq f_{i,j}, \quad j = 1, \dots, m + 1 \tag{2.3}$$

$$f_{1,0} = 0 \tag{2.4}$$

$$\sum_{i \in S_t} r_{ik} \leq a_k, \quad \forall k \in K, \quad t = 1, \dots, f_{n,0} \tag{2.5}$$

La función objetivo, de acuerdo a 2.1, consiste en minimizar el tiempo en que finaliza la única subactividad de la última actividad.

El conjunto de restricciones 2.2 indica que las subactividades de cada actividad deben ser planeadas después de que todas las subactividades de las actividades que la preceden sean planeadas.

De acuerdo a 2.3, las subactividades de una actividad deben ser planeadas en orden. Según la ecuación 2.4, la única subactividad que compone la actividad ficticia inicial finaliza en el tiempo 0.

El conjunto de restricciones 2.5 indica que en cualquier intervalo de tiempo  $(t - 1, t]$ , la demanda de unidades por parte de las actividades que utilizan un recurso  $k$  no debe superar su disponibilidad.

Retomando el ejemplo de la Figura 1.14, del capítulo anterior, se puede notar que toma siete unidades de tiempo terminar las actividades sin interrumpirlas, pero si se interrumpe la Actividad 2 habrá un ahorro en el tiempo total como se muestra en la Figura 2.1.

Al interrumpir la Actividad 2, se ahorran dos unidades de tiempo 'rellenando' los espacios que estaban sin ocupar en la primer planeación. Ésta es la razón por la cual se busca relajar la condición de interrupciones en el problema RCPSP.

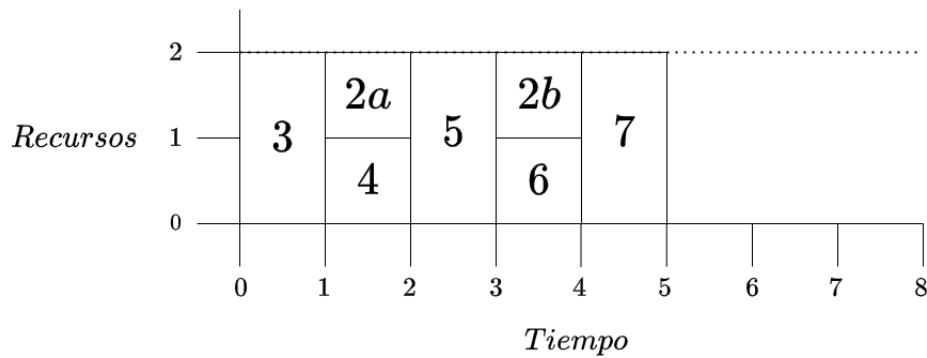


Figura 2.1: Planeación óptima con interrupciones.

## 2.2. Estrategia de solución

A partir de las actividades descritas en el proyecto, se diseña un nuevo proyecto que contenga subactividades, las actividades ficticias 1 y  $n$ , pasan a ser subactividades ficticias de duración cero. Cada actividad se separa en subactividades de duración uno, por lo que, por cada actividad el número de subactividades es la duración de la actividad.

**Definición 2.2.1** Una **planeación parcial**  $PS_t$  en un tiempo  $t$  es el conjunto de subactividades planeadas hasta el tiempo  $t$ , tal que su planeación respeta la precedencia de las actividades y las restricciones de recursos.

**Definición 2.2.2** Una **actividad sin terminar** en un tiempo  $t$  es una actividad para la cual no todas sus subactividades han sido planeadas.

**Definición 2.2.3** Un **conjunto elegible**  $E_t$  en un tiempo  $t$  son las actividades para las cuales alguna de sus subactividades se pueden planear respetando la precedencia de actividades y las restricciones de recursos.

### Reglas de dominio

Las reglas de dominio están dadas por los teoremas que adelante se definen, estas reglas disminuyen considerablemente el número de planificaciones factibles a considerar cuando se buscan soluciones. Por lo que cada teorema define una regla de dominio.

**Definición 2.2.4** Una planeación parcial es **semi-activa** si para cualquier planeación de cualquier subactividad ya no es posible planearla en un tiempo menor a la actual respetando la precedencia de actividades y las restricciones de recursos.

De la definición anterior, como se busca minimizar el tiempo en la planeación de subactividades, es necesario que sólo se consideren planeaciones semi-activas, pues esto ayuda a reducir el número de planeaciones factibles a considerar.



**Teorema 2.2.1** Para resolver el PRCPSP es suficiente construir las planeaciones parciales semi-activas al nivel de subactividades.

**Definición 2.2.5** Una actividad es **planeada inmediatamente** en un tiempo  $t$  si todas sus  $z$  subactividades que no han sido planeadas, denotadas como  $i_x$  ( $x = 1, 2, ..z$ ), son planeadas inmediatamente en el tiempo  $t$  de tal forma que  $f_{i_x} = t + x$ .

**Teorema 2.2.2** Dada una planeación parcial  $PS_t$  en un tiempo  $t$ , si existe una única actividad  $i$  que es elegible y que además no puede ser planeada en cualquier otro tiempo mayor o igual a  $t$  junto con alguna otra actividad sin terminar, respetando la precedencia de actividades y restricciones de recursos, entonces existe una continuación óptima de  $PS_t$ , de tal forma que las subactividades de  $i$  que aún no han sido planeadas, se planean inmediatamente a partir del tiempo  $t$ .

El teorema indica que si se encuentra una actividad que no se pueda planear al mismo tiempo con alguna otra, entonces no queda otra opción que planearla inmediatamente.

**Teorema 2.2.3** Dada una planeación parcial  $PS_t$  en un tiempo  $t$ , si existe una actividad  $i$  tal que es elegible y además puede ser planeada en cualquier otro tiempo mayor o igual a  $t$  junto con una única actividad  $j$  sin terminar y respetando la precedencia de actividades y las restricciones de recursos, entonces existe una continuación óptima de  $PS_t$ , de tal forma que las subactividades tanto de  $i$  como de  $j$  que no han sido planeadas, se planean al mismo tiempo.

De acuerdo al teorema esta vez se van a planear al mismo tiempo dos actividades  $i, j$  pero sólo si son las únicas dos que se puedan planear al mismo tiempo, sin importar la duración, pues si la actividad  $j$  es de mayor duración, se habrán planeado tantas subactividades de  $j$  como subactividades de  $i$ , por otro lado, si la duración es menor o igual, la actividad  $j$  se planea totalmente.

Aunque ambos teoremas difieren, tienen la similitud de que ambos presentan casos en los que directamente se planean las actividades que cumplen algunas características únicas en un tiempo dado y no es necesario elegir entre varias actividades como se ve en los siguientes teoremas.

**Definición 2.2.6** Si no es posible planear todas las subactividades de las actividades elegibles en un tiempo  $t$ , entonces ocurre un **conflicto de recursos**.

**Definición 2.2.7** Cuando sucede un conflicto de recursos en la planeación actual, se define a una **rama** como una nueva planeación a partir de la actual, donde se resuelve el conflicto de alguna manera.

Claramente, a partir de una planeación, al surgir un conflicto de recursos se crean distintas ramas donde cada una utiliza una distinta planeación y debido a esto, intuitivamente se puede ver que se recorre un árbol, donde la raíz es la planeación inicial y de cada nodo surgen ramas que dan a lugar a distintas planeaciones.

**Definición 2.2.8** Una **Alternativa con retraso**  $D_q$  es un conjunto que contiene las actividades elegibles que resuelven un conflicto de recursos.

**Definición 2.2.9** Un **Conjunto de retraso**  $D(p)$  consiste en conjuntos  $D_q$  de Alternativas con retraso.

**Definición 2.2.10** Una Alternativa con retraso  $D_q$  es **mínima** si no contiene alguna otra Alternativa con retraso  $D_v \in D(p)$ .

**Teorema 2.2.4** Para resolver un conflicto de recursos es suficiente considerar sólo Alternativas con retraso que sean mínimas.

**Teorema 2.2.5** Dada una planeación parcial  $PS_t$  en un tiempo  $t$ , si existe una planeación parcial  $PS'_t$  que fue obtenida antes de la actual en el mismo tiempo y la planeación parcial  $PS_t$  está contenida en  $PS'_t$ , entonces se dice que la planeación parcial  $PS_t$  es **dominada**.

**Definición 2.2.11** La **ruta crítica restante** de una subactividad es la serie más larga de subactividades que desde la subactividad actual deben ser planeadas para terminar el proyecto, incluyendo la subactividad actual y sin contar las subactividades ficticias.

**Definición 2.2.12** Para cada Alternativa con retraso  $D_q$  construida en el tiempo  $t$ , se define la **cota inferior de la ruta crítica restante** como  $l_q = t + 1 + \max_{i \in D_q} \{q_i\}$ , donde  $q_i$  es la longitud de la ruta crítica restante de la subactividad  $i$ .

### 2.2.1. Proceso Branch and Bound

El proceso BRANCH AND BOUND<sup>1</sup> realiza una búsqueda sobre un árbol donde cada nodo representa una planeación parcial y haciendo uso de las reglas de dominio logra disminuir la cantidad de planeaciones a considerar. A continuación los pasos del proceso.

#### Paso 1: Inicialización

Dado un proyecto, las actividades se dividen en subactividades de duración uno, donde el número de subactividades por actividad depende de su duración  $d_i$ , así mismo, cada subactividad requiere la misma cantidad de unidades de recurso  $r_{ik}$  de la actividad a la que pertenece, las actividades ficticias 1 y  $n$  pasan a ser subactividades de duración cero.

La cota superior del proyecto se inicializa en  $T = 9999$ .

Se define el nivel del árbol en  $p = 0$ .

Se define el punto de decisión  $t = -1$  que indica el tiempo en la planeación.

Para cada subactividad  $i$  se calcula la longitud de su ruta crítica restante  $q_i$ .

En este paso lo único que se debe planear es la subactividad ficticia 1; al ser la única en ser planeada, es necesario asignar la cota inferior de la ruta crítica en el nivel cero como la ruta crítica restante de la subactividad ficticia 1.

Además de calcular el conjunto de subactividades elegibles como las subactividades que según la precedencia de actividades se pueden realizar después de que la subactividad ficticia 1 se haya planeado.

El Algoritmo 4 presenta un pseudo-código para el paso de Inicialización.

#### Paso 2: Aumento

Se revisa si la subactividad ficticia  $n$  ya fue planeada, si lo es, entonces según las restricciones del proyecto, todas las subactividades del proyecto han sido planeadas, por lo que se asigna como duración del proyecto el tiempo en que termina la actividad ficticia  $n$ , con esto se revisa si el tiempo en que finaliza la subactividad ficticia  $n$  es igual a  $lb_0$ , esto es porque el algoritmo va a buscar soluciones donde el tiempo esté acotado inferiormente por  $lb_0$ , por lo que la mejor duración de un proyecto que se puede encontrar con el algoritmo es  $lb_0$ . Por otro lado, si no se cumple que  $T = lb_0$ , entonces es necesario regresar al nivel anterior en el árbol de búsqueda realizando un RETROCESO (Paso 7).

---

<sup>1</sup>Dado que es una técnica muy conocida, se ha decidido dejar el término en inglés.

---

**Algoritmo 4** INICIALIZACIÓN

---

**Entrada:** Conjunto de actividades  $P$ .  
Conjunto  $K$  de recursos renovables.  
Conjunto  $H$  de restricciones.

**Salida:** Planeación mejor encontrada.

```
1: subactividades  $\leftarrow \emptyset$ 
2:  $PS_t \leftarrow \emptyset$ 
3:  $S_t \leftarrow \emptyset$ 
4: //Divide cada actividad en subactividades
5: for all actividad  $\in P$  do
6:   subactividades.agregar(divide_subactividades(actividad))
7: end for
8:  $T \leftarrow 9999$  //Cota superior para la duración del proyecto
9:  $p \leftarrow 0$  //Nivel del árbol
10:  $t \leftarrow -1$  //Tiempo de la planeación actual
11:  $PS_t \leftarrow \{subactividad_1\}$  //Planea la subactividad ficticia inicial
12: subactividad_1.tiempo  $\leftarrow 0$ 
13:  $S_t \leftarrow \{subactividad_1\}$ 
14: //Calcula la cota inferior de la ruta crítica del nivel cero como la ruta crítica restante de la
    subactividad ficticia inicial
15:  $lb_0 \leftarrow subactividad_1.ruta\_critica\_restante$ 
16:  $E_t \leftarrow \{i \mid \text{la subactividad } i \text{ tiene como único predecesor a la subactividad } 1 \}$ 
```

---

Si la subactividad ficticia  $n$  no fue planeada, se aumenta el tiempo actual y se revisa si la planeación actual es dominada por alguna anteriormente obtenida, pues si esto sucede, entonces es necesario regresar a una mejor planeación realizando un RETROCESO (Paso 7).

El Algoritmo 5 presenta un pseudo-código para el paso de Aumento.

---

**Algoritmo 5** AUMENTO

---

```
1: if subactividadn ya está planeada then
2:    $T \leftarrow subactividad_n.tiempo$ 
3:   if  $T = lb_0$  then
4:     TERMINA
5:   end if
6:    $p \leftarrow p - 1$ 
7:   RETROCESO
8: end if
9:  $t \leftarrow t + 1$ 
10: if  $PS_t$  es dominada por alguna planeación anterior then
11:   RETROCESO
12: end if
```

---

**Paso 3: Separación**

En este paso se analizan los casos en los que existe alguna actividad elegible tal que sus subactividades restantes se pueden planear al mismo tiempo con una única actividad o bien puede ser planeada únicamente.

Para cada actividad elegible  $i$  calcula las actividades sin terminar que se pueden planear al mismo tiempo con la actividad  $i$  en cualquier tiempo  $t' \geq t$ , respetando la precedencia de actividades y las restricciones de recursos.

Una vez hecho esto, se tienen dos casos.

- Ninguna actividad se puede planear al mismo tiempo con la actividad  $i$ .

Se aplica el Teorema 2.2.2, se planean inmediatamente las  $z$  subactividades restantes sin planear,  $i_x$  ( $x = 1..z$ ), por lo que  $PS_t = PS_t + \{i_x\}$ .

Asigna el tiempo de término a cada subactividad,  $f_{i_x} = t + x$ .

Actualiza el tiempo actual  $t = f_{i_z} - 1$ .

Como la última subactividad es  $i_z$ , entonces  $S_t = \{i_z\}$ .

Actualiza el conjunto de actividades elegibles  $E = E - \{i_1\} + \{y \mid \text{la subactividad } y \text{ es sucesora de } i_z, \text{ además todas las subactividades predecesoras han sido planeadas}\}$ .

Por último, si la subactividad ficticia  $n$  se planea durante este paso, entonces directamente se le asigna  $f_n = t$ .

- Sólo una actividad  $j$  se puede planear al mismo tiempo con la actividad  $i$ , además  $j$  es elegible.

Se aplica el Teorema 2.2.3, se planean al mismo tiempo las subactividades tanto de  $i$  como  $j$ , tantas como haya subactividades de  $i$ .

Se planean inmediatamente las  $z$  subactividades restantes sin planear de  $i$ ,  $i_x$  ( $x = 1..z$ ), por lo que  $PS_t = PS_t + \{i_x\}$ .

Asigna el tiempo de término a cada subactividad de  $i$ ,  $f_{i_x} = t + x$ .

Como la última subactividad es  $i_z$ , entonces  $S_t = \{i_z\}$ .

Sea  $z'$  el número de subactividades de  $j$  sin planear, se obtiene el mínimo de las subactividades sin planear entre  $i, j$  con  $z'' = \min\{z, z'\}$ .

De la misma manera que  $i$ , se planean las  $z''$  subactividades de  $j$  sin planear,  $j_x$  ( $x = 1..z''$ ), por lo que  $PS_t = PS_t + \{j_x\}$ .

Como las últimas subactividades son  $i_z$  y  $j_{z''}$ , entonces  $S_t = S_t + \{j_{z''}\}$ .

Asigna el tiempo de término a cada subactividad de  $j$ ,  $f_{j_x} = t + x$ .

Actualiza el tiempo actual  $t = f_{i_z} - 1$ .

Actualiza el conjunto de actividades elegibles  $E = E - \{i_1, j_1\} + \{y \mid \text{la subactividad } y \text{ es sucesora de } i_z \text{ o } j_{z''}, \text{ además todas las subactividades predecesoras han sido planeadas}\}$ .

Si no se pudo planear alguna actividad, realizar una PLANEACIÓN (Paso 4), en otro caso revisar si las planeaciones han incrementado la ruta crítica de la mejor solución factible hasta el momento, por lo que si  $t + 1 + \max_{i \in E_t} \{q_i\} \geq T$  realizar un RETROCESO (Paso 7), en otro caso hacer un AUMENTO (Paso 2).

El Algoritmo 6 presenta un pseudo-código para el paso de Separación.

---

**Algoritmo 6** SEPARACIÓN

---

```

1: for all actividadi elegible do
2:   //Calcula las actividades elegibles que se pueden planear al mismo tiempo con la actividad i
3:    $A \leftarrow \text{obtener\_actividades\_mismo\_tiempo}(\text{actividad}_i)$ 
4:   //Ninguna actividad se puede planear al mismo tiempo con la actividad i
5:   if  $|A| = 0$  then
6:     for all subactividadx  $\in$  actividadi.subactividades_restantes do
7:        $PS_t \leftarrow PS_t + \{\text{subactividad}_x\}$ 
8:       subactividadx.tiempo  $\leftarrow t + x$ 
9:     end for
10:    //Usa el tiempo de la última subactividad
11:     $t \leftarrow \text{actividad}_i.\text{subactividad}_z.\text{tiempo} - 1$ 
12:     $S_t \leftarrow \{\text{actividad}_i.\text{subactividad}_z\}$ 
13:     $E_t \leftarrow E_t - \{i_1\} + \{y \mid \text{la subactividad } y \text{ es sucesora de la última subactividad planeada, además todas las subactividades predecesoras han sido planeadas}\}$ 
14:    if subactividadn  $\in PS_t$  then
15:      subactividadn.tiempo =  $t$ 
16:    end if
17:  end if
18:  //Solo una actividad se puede planear al mismo tiempo con la actividad i
19:  if  $|A| = 1$  and es_elegible( $A$ ) then
20:    actividadj  $\leftarrow A[0]$ 
21:     $z' \leftarrow \min\{\text{actividad}_j.\text{subactividades\_sin\_planear}, \text{actividad}_i.\text{subactividades\_sin\_planear}\}$ 
22:    for all subactividadx  $\in$  actividadi.subactividades_restantes do
23:       $PS_t \leftarrow PS_t + \{\text{subactividad}_x\}$ 
24:      subactividadx.tiempo  $\leftarrow t + x$ 
25:    end for
26:     $S_t \leftarrow \{\text{actividad}_i.\text{subactividad}_{z'}\}$ 
27:    for all subactividadx  $\in$  actividadj.subactividades_restantes( $z'$ ) do
28:       $PS_t \leftarrow PS_t + \{\text{subactividad}_x\}$ 
29:      subactividadx.tiempo  $\leftarrow t + x$ 
30:    end for
31:     $S_t \leftarrow S_t + \{\text{actividad}_j.\text{subactividad}_{z'}\}$ 
32:     $t \leftarrow \text{actividad}_i.\text{subactividad}_{z'}.\text{tiempo} - 1$ 
33:     $E_t \leftarrow E_t - \{i_1, j_1\} + \{y \mid \text{la subactividad } y \text{ es sucesora de las últimas subactividades planeadas, además todas las subactividades predecesoras han sido planeadas}\}$ 
34:  end if
35: end for
36: if ninguna actividad se planeo then
37:   PLANEACIÓN
38: else
39:   //Revisa si las planeaciones incrementaron la ruta crítica de la solución actual
40:   if  $t + 1 + \max_{\text{subactividad} \in E_t} \{q_i\} \geq T$  then
41:     RETROCESO
42:   else
43:     AUMENTO
44:   end if
45: end if

```

---

#### Paso 4: Planeación

En este paso se planean las subactividades elegibles que pueden ser planeadas siempre que se respete el uso de recursos.

- Para cada recurso  $k \in K$ , revisar si  $\sum_{i \in E_t} r_{ik} \leq a_k$ .
- Si existe algún recurso para el cual la suma de lo requerido por las actividades es mayor a la disponibilidad del recurso, entonces existe un conflicto de recursos, por lo que se hace el Paso 5.
- Si no ocurre algún conflicto, entonces se planean las subactividades elegibles,  $PS_t = PS_t + E_t$ .  
Como se realiza en el mismo tiempo,  $S_t = E_t$ .

Por lo que se les asigna el mismo tiempo de término,  $\forall i \in E_t, f_i = t + 1$ .

Finalmente calcula el conjunto de actividades elegibles  $E_t = \{ j \mid \text{la subactividad } j \text{ es sucesora de } i \in S_t \text{ y todas las subactividades predecesoras de } j \text{ han sido planeadas} \}$  y realizar un AUMENTO (Paso 2).

El Algoritmo 7 presenta un pseudo-código para el paso de Planeación.

---

#### Algoritmo 7 PLANEACIÓN

---

```
1: if Algún recurso no cumple la demanda de las actividades then
2:   CONFLICTO DE RECURSOS
3: else
4:    $PS_t \leftarrow PS_t + E_t$ 
5:    $S_t \leftarrow E_t$ 
6:   for all actividad  $\in E_t$  do
7:     actividad.tiempo  $\leftarrow t + 1$ 
8:   end for
9:    $E_t = \{ j \mid \text{la subactividad } j \text{ es sucesora de } i \in S_t \text{ y todas las subactividades predecesoras de } j \text{ han sido planeadas} \}$ 
10:  AUMENTO
11: end if
```

---

#### Paso 5: Conflicto de recursos

En este paso, para solucionar un conflicto de recursos lo que se realiza es quitar de las subactividades a planear un subconjunto de actividades tal que las unidades del recurso que utilizan sean al menos la cantidad de unidades que faltan del recurso y que causó el conflicto.

Una vez que se obtienen los subconjuntos que pueden resolver el conflicto, se descarta del Conjunto de retraso al subconjunto que tiene la subactividad más cercana a la subactividad final del proyecto, esto es para dar más prioridad de planeación a las subactividades más lejanas de la subactividad final.

Por lo que se describen los pasos:

- Incrementa el nivel del árbol,  $p = p + 1$ .
- Calcula para cada recurso  $k \in K$ , cuántas unidades faltan para cumplir con las unidades requeridas por las subactividades,  $c_k = \sum_{i \in E_t} r_{ik} - a_k$ .
- Calcula el Conjunto de retraso  $D(p) = \{ D_q \mid D_q \text{ es un subconjunto de } E_t, \text{ además se cumple que } \sum_{i \in D_q} r_{ik} \geq c_k \text{ para cada recurso } k \text{ y } D_q \text{ no contiene a otro conjunto } D_u \in D(p) \}$ .
- Para cada  $D_q \in D(p)$ , calcula la cota inferior,  $l_q = t + 1 + \max_{i \in D_q} \{q_i\}$ .

- Sea  $D_b$  el conjunto  $D_q \in D(p)$  con el valor más pequeño de  $l_q$ .
- Actualiza el Conjunto de retraso  $D(p) = D(p) - D_b$ .
- Calcula  $lb_p = \max\{lb_{p-1}, l_b\}$ .  
Si  $lb_p \geq T$  decrementa el nivel del árbol a  $p = p - 1$  y realiza un RETROCESO (Paso 7).
- Guarda la planeación parcial  $PS_t$  y el conjunto de actividades elegibles  $E_t$ .

El Algoritmo 8 presenta un pseudo-código para el paso de Conflicto de recursos.

---

**Algoritmo 8** CONFLICTO DE RECURSOS

---

```

1:  $p \leftarrow p + 1$ 
2: for all  $recurso_i \in K$  do
3:    $recurso_i.calcula\_unidades\_faltantes()$ 
4: end for
5:  $D(p) \leftarrow \{D_q | D_q \text{ es un subconjunto de } E_t, \text{ además se cumple que } \sum_{i \in D_q} r_{ik} \geq c_k \text{ para cada recurso } k \text{ y } D_q \text{ no contiene a otro conjunto } D_u \in D(p)\}$ 
6: Para cada  $D_q \in D(p)$  calcula la cota inferior,  $l_q = t + 1 + \max_{i \in D_q} \{q_i\}$ .
7:  $D_b \leftarrow$  el conjunto  $D_q \in D(p)$  con el valor más pequeño de  $l_q$ .
8:  $D(p) \leftarrow D(p) - D_b$ 
9:  $lb_p \leftarrow \max\{lb_{p-1}, l_b\}$ 
10: if  $lb_p \geq T$  then
11:    $p \leftarrow p - 1$ 
12:   RETROCESO
13: end if
14: Guarda la planeación parcial  $PS_t$  y el conjunto de actividades elegibles  $E_t$ 

```

---

**Paso 6: Retraso**

En este paso se planean las subactividades elegibles que no pertenecen al subconjunto que resuelve el conflicto de recursos.

- Del conjunto  $D_b$  escogido en el paso anterior, planea todas las subactividades elegibles que no pertenecen a este conjunto,  $PS_t = PS_t + E - D_b$ .
- Actualiza el tiempo de término de las subactividades planeadas que no pertenecen a  $D_b$ ,  $\forall i \in (E - D_b), f_i = t + 1$ .
- Actualiza el conjunto de subactividades  $S_t = E_t - D_b$ .
- Actualiza el conjunto de actividades elegibles  $E_t = D_b + \{ j \mid \text{la subactividad } j \text{ es sucesora de cada actividad } i \in S_t \text{ y todas las subactividades predecesoras a } j \text{ han sido planeadas} \}$ .
- Realiza un AUMENTO (Paso 2).

El Algoritmo 9 presenta un pseudo-código para el paso de Retraso.

---

**Algoritmo 9** RETRASO

---

```

1: //Del conjunto  $D_b$  escogido en el paso anterior, planea todas las subactividades elegibles que no
   pertenecen a este conjunto
2:  $PS_t \leftarrow PS_t + E - D_b$ 
3: //Actualiza el tiempo de termino de las subactividades planeadas que no pertenecen a  $D_b$ 
4: for all subactividad  $\in (E - D_b)$  do
5:   subactividad.tiempo  $\leftarrow t + 1$ 
6: end for
7:  $S_t \leftarrow E_t - D_b$ 
8:  $E_t \leftarrow D_b + \{ j \mid \text{la subactividad } j \text{ es sucesora de cada actividad } i \in S_t \text{ y todas las subactividades}
   \text{ predecesoras a } j \text{ han sido planeadas} \}$ 
9: AUMENTO

```

---

**Paso 7: Retroceso**

- Si el nivel del árbol  $p \leq 0$ , entonces termina.
- Si  $D(p) = \emptyset$ , entonces  $p = p - 1$  y realiza un RETROCESO (Paso 7).
- Selecciona la Alternativa con retraso  $D_b \in D(p)$  con la cota inferior más pequeña  $l_b$  y actualiza el Conjunto de retraso  $D(p) = D(p) - D_b$ .
- Calcula  $lb_p = \max\{lb_{p-1}, l_b\}$ , si  $lb_p \geq T$ , actualiza el nivel del árbol  $p = p - 1$  y realiza un RETROCESO (Paso 7).
- Recupera la planeación parcial y el conjunto elegible y realiza un RETRASO (Paso 6).

El Algoritmo 10 presenta un pseudo-código para el paso de Retroceso.

---

**Algoritmo 10** RETROCESO

---

```

1: if  $p \leq 0$  then
2:   TERMINA
3: end if
4: if  $D(p) = \emptyset$  then
5:    $p \leftarrow p - 1$ 
6:   RETROCESO
7: end if
8: Selecciona la Alternativa con retraso  $D_b \in D(p)$  con la cota inferior mas pequeña  $l_b$ 
9:  $D(p) \leftarrow D(p) - D_b$ 
10:  $lb_p \leftarrow \max\{lb_{p-1}, l_b\}$ 
11: if  $lb_p \geq T$  then
12:    $p \leftarrow p - 1$ 
13:   RETROCESO
14: end if
15: Recupera la planeación parcial y el conjunto elegible.
16: RETRASO

```

---

A continuación se describe el flujo de los pasos en el proceso BRANCH AND BOUND.



Para la INICIALIZACIÓN (Paso 1) al terminar se realiza un AUMENTO (Paso 2).

Para el AUMENTO (Paso 2), se puede terminar la ejecución del proceso o realizar un RETROCESO (Paso 7) o bien seguir con la SEPARACIÓN (Paso 3).

Para la SEPARACIÓN (Paso 3), se puede realizar un RETROCESO (Paso 7) o un AUMENTO (Paso 2), si no se cumplen las condiciones para realizar alguno de esos pasos, entonces se sigue con la PLANEACIÓN (Paso 4).

Para la PLANEACIÓN (Paso 4), se puede realizar un AUMENTO (Paso 2) o seguir con el CONFLICTO DE RECURSOS (Paso 5).

Para el CONFLICTO DE RECURSOS (Paso 5), se puede realizar un RETROCESO (Paso 7) o seguir con el RETRASO (Paso 6).

Para el RETRASO (Paso 6), se puede realizar un AUMENTO (Paso 2) o seguir con el RETROCESO (Paso 7).

Para el RETROCESO (Paso 7), se puede terminar la ejecución del proceso o repetir el paso, pero si no se cumplen las condiciones para repetir el paso, entonces se puede realizar un RETRASO (Paso 6).

Por lo observado en cada paso, estos se ejecutan en orden secuencial y sólo se realizan saltos a otros pasos cuando se cumplen condiciones en cada paso, la Figura 3.2 muestra el orden en el que se realizan los pasos.

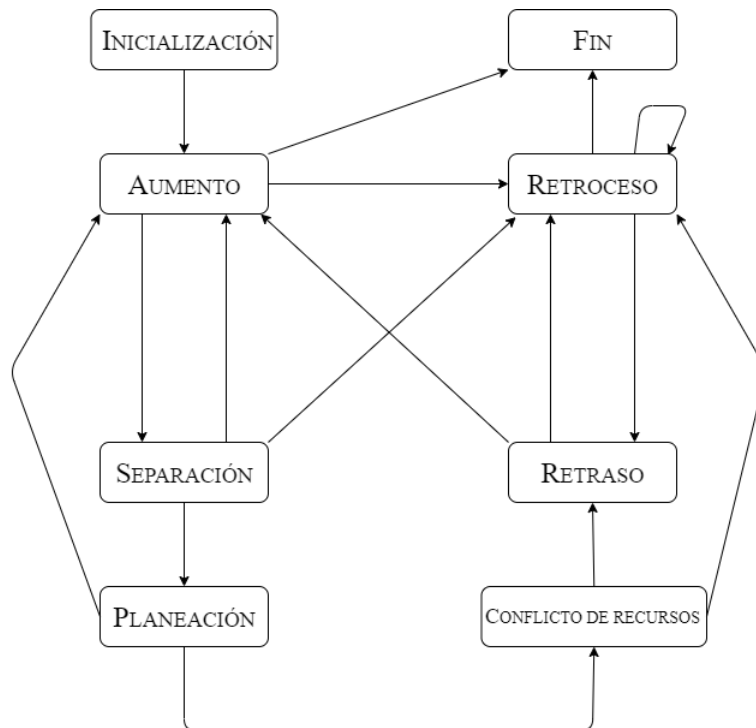


Figura 2.2: Orden de ejecución de los pasos

### Ejemplo

Se tiene un proyecto de 9 actividades, recordando que las actividades 1 y 9 son ficticias y un recurso con capacidad de cinco unidades.

Para cada nodo que representa una actividad, su duración se encuentra arriba del nodo y las unidades de recurso que utiliza se encuentran debajo del nodo.

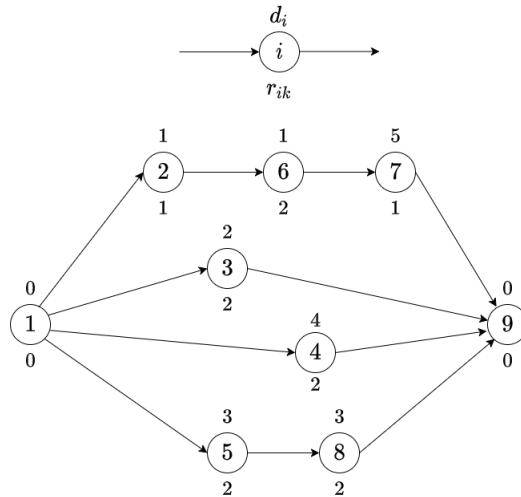


Figura 2.3: Proyecto de 9 actividades

De acuerdo al proceso BRANCH AND BOUND se describen los pasos obtenidos.

Paso 1 Construye un nuevo proyecto a partir de la entrada dividiendo las actividades en subactividades, se puede ver que las subactividades 5, 6, 7, 8 corresponden a la actividad 4 en el proyecto original, en cuanto a la actividad 2 se divide en una única subactividad 2, la actividad 6 se divide en una única subactividad 12, la actividad 7 se divide en las subactividades 13, 14, 15, 16, 17, para la actividad 3 se divide en las subactividades 3, 4, la actividad 5 se divide en subactividades 9, 10, 11 y la actividad 8 se divide en subactividades 18, 19, 20.

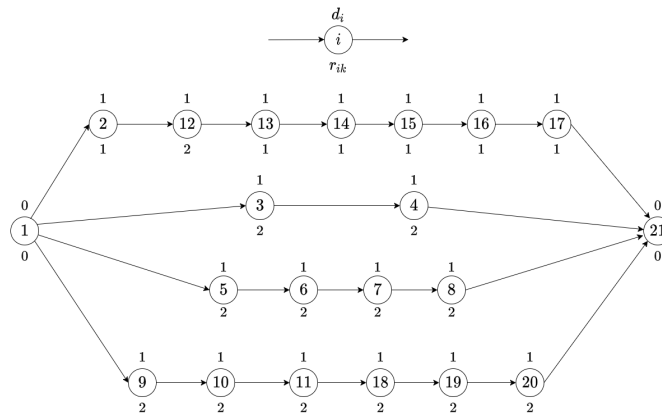


Figura 2.4: Proyecto con subactividades

Inicia la cota superior  $T$  del proyecto en  $T = 9999$ .

El nivel del árbol en  $p = 0$ .

El punto de decisión o el tiempo actual en  $t = -1$ .

Planea la subactividad ficticia 1 en el tiempo 0,  $f_1 = 0$ .

$$PS = \{1\}, S = \{1\}$$

Determina la ruta crítica restante de la subactividad 1 y actualiza  $lb_0 = q_1 = 7$ .

El conjunto elegible serán las subactividades que son sucesoras a la subactividad inicial 1, lo cual es:  $E_t = \{2, 3, 5, 9\}$ .

Paso 2 Actualiza  $t = 0$ .

Como la planeación actual es la única, entonces no está dominada, por lo que guarda la planeación actual y el tiempo actual.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{2, 3, 5, 9\}$ , cualquiera se puede planea de manera concurrente con al menos otras dos del mismo conjunto, como la subactividad 2 con las subactividades 3 y 5 o bien, la subactividad 3 con las subactividades 9 y 2, por lo que no se pueden planea subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 7, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 1$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 2$ .

Determina el conjunto  $D(1)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 2; por simplicidad, los subconjuntos elegidos son de tamaño 1, de manera que:  $D_1 = \{3\}$ ,  $D_2 = \{5\}$  y  $D_3 = \{9\}$ , donde  $D(1) = \{D_1, D_2, D_3\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned}l_1 &= 0 + 1 + \max_{i \in D_1} \{q_3\} = 3, \\l_2 &= 0 + 1 + \max_{i \in D_2} \{q_5\} = 5, \\l_3 &= 0 + 1 + \max_{i \in D_3} \{q_9\} = 7.\end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_1$  por tener la cota inferior mínima de 3, pues a la subactividad 3 sólo le hace falta planearse además de la subactividad 4 para llegar a la subactividad ficticia final, lo cual hace que  $q_3 = 2$  y por lo tanto  $l_1 = 3$ .

Se actualiza  $D(1) = D(1) - D_1 = \{\{5\}, \{9\}\}$ .

Determina  $lb_1 = \max\{lb_0, l_1\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_1 = \{1, 2, 5, 9\}$ .

$$S = E - D_1 = \{2, 5, 9\}.$$

Asigna las fechas de término a las subactividades planeadas,  $f_2 = f_5 = f_9 = 1$ .

Determina el conjunto de actividades elegibles  $E_t = \{12, 3, 6, 10\}$ .

Paso 2 Actualiza  $t = 1$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{12, 3, 6, 10\}$ , cualquiera se puede planear de manera concurrente con al menos otra del mismo conjunto, como la subactividad 12 con la subactividad 3 o 6, la subactividad 3 con la subactividad 6 o 10, por lo que no se pueden planear subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 8, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 2$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 3$ .

Determina el conjunto  $D(2)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 3; de manera que:  $D_4 = \{3, 6\}$ ,  $D_5 = \{3, 10\}$ ,  $D_6 = \{3, 12\}$ ,  $D_7 = \{6, 10\}$ ,  $D_8 = \{6, 12\}$ ,  $D_9 = \{10, 12\}$ , donde  $D(2) = \{D_4, D_5, D_6, D_7, D_8, D_9\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned} l_4 &= 1 + 1 + \max_{i \in D_4} \{q_3, q_6\} = 1 + 1 + \max_{i \in D_4} \{2, 3\} = 5, \\ l_5 &= 1 + 1 + \max_{i \in D_5} \{q_3, q_{10}\} = 1 + 1 + \max_{i \in D_5} \{2, 5\} = 7, \\ l_6 &= 1 + 1 + \max_{i \in D_6} \{q_3, q_{12}\} = 1 + 1 + \max_{i \in D_6} \{2, 6\} = 8, \\ l_7 &= 1 + 1 + \max_{i \in D_7} \{q_6, q_{10}\} = 1 + 1 + \max_{i \in D_7} \{3, 5\} = 7, \\ l_8 &= 1 + 1 + \max_{i \in D_8} \{q_6, q_{12}\} = 1 + 1 + \max_{i \in D_8} \{3, 6\} = 8, \\ l_9 &= 1 + 1 + \max_{i \in D_9} \{q_{10}, q_{12}\} = 1 + 1 + \max_{i \in D_9} \{5, 6\} = 8. \end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_4$  por tener una cota inferior mínima de 5.

Se actualiza  $D(2) = D(2) - D_4 = \{\{3, 10\}, \{3, 12\}, \{6, 10\}, \{6, 12\}, \{10, 12\}\}$ .

Determina  $lb_2 = \max\{lb_1, l_4\} = \max\{7, 5\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_4 = \{1, 2, 5, 9, 10, 12\}$ .

$$S = E - D_4 = \{10, 12\}.$$

Asigna las fechas de término a las subactividades planeadas,  $f_{10} = f_{12} = 2$ .

Determina el conjunto de actividades elegibles  $E_t = \{13, 3, 6, 11\}$ .

Paso 2 Actualiza  $t = 2$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{13, 3, 6, 11\}$ , cualquiera se puede planear de manera concurrente con al menos dos del mismo conjunto, como la subactividad 13 con las subactividades 3 y 11, la subactividad 3 con la subactividad 6 o 11, por lo que no se pueden planear subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 7, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 3$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 2$ .

Determina el conjunto  $D(3)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 2; de manera que:  $D_{10} = \{13\}, D_{11} = \{3\}, D_{12} = \{6\}, D_{13} = \{11\}$ , donde  $D(3) = \{D_{10}, D_{11}, D_{12}, D_{13}\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned} l_{10} &= 2 + 1 + \max_{i \in D_{10}} \{q_{13}\} = 2 + 1 + \max_{i \in D_{10}} \{5\} = 8, \\ l_{11} &= 2 + 1 + \max_{i \in D_{11}} \{q_3\} = 2 + 1 + \max_{i \in D_{11}} \{2\} = 5, \\ l_{12} &= 2 + 1 + \max_{i \in D_{12}} \{q_6\} = 2 + 1 + \max_{i \in D_{12}} \{3\} = 6, \\ l_{13} &= 2 + 1 + \max_{i \in D_{13}} \{q_{11}\} = 2 + 1 + \max_{i \in D_{13}} \{4\} = 7. \end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_{11}$  por tener una cota inferior mínima de 5.

Se actualiza  $D(3) = D(3) - D_{11} = \{\{13\}, \{6\}, \{11\}\}$ .

Determina  $lb_3 = \max\{lb_2, l_{11}\} = \max\{7, 5\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_{11} = \{1, 2, 5, 9, 10, 12, 13, 6, 11\}$ .

$S = E - D_{11} = \{13, 6, 11\}$ .

Asigna las fechas de término a las subactividades planeadas,  $f_{13} = f_6 = f_{11} = 3$ .

Determina el conjunto de actividades elegibles  $E_t = \{14, 3, 7, 18\}$ .

Paso 2 Actualiza  $t = 3$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{14, 3, 7, 18\}$ , cualquiera se puede planear de manera concurrente con alguna otra del mismo conjunto, como la subactividad 14 con las subactividades 3 y 7, la subactividad 18 con la subactividad 3 o 7, por lo que no se pueden planear subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 7, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 4$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 2$ .

Determina el conjunto  $D(4)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 2; de manera que:  $D_{14} = \{14\}, D_{15} = \{3\}, D_{16} = \{7\}, D_{17} = \{18\}$ , donde  $D(4) = \{D_{14}, D_{15}, D_{16}, D_{17}\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned} l_{14} &= 3 + 1 + \max_{i \in D_{14}} \{q_{14}\} = 3 + 1 + \max_{i \in D_{14}} \{4\} = 8, \\ l_{15} &= 3 + 1 + \max_{i \in D_{15}} \{q_3\} = 3 + 1 + \max_{i \in D_{15}} \{2\} = 6, \\ l_{16} &= 3 + 1 + \max_{i \in D_{16}} \{q_7\} = 3 + 1 + \max_{i \in D_{16}} \{2\} = 6, \\ l_{17} &= 3 + 1 + \max_{i \in D_{17}} \{q_{18}\} = 3 + 1 + \max_{i \in D_{17}} \{3\} = 7. \end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_{15}$  por tener una cota inferior mínima de 6.

Se actualiza  $D(4) = D(4) - D_{15} = \{\{14\}, \{7\}, \{18\}\}$ .

Determina  $lb_4 = \max\{lb_3, l_{15}\} = \max\{7, 6\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_{15} = \{1, 2, 5, 9, 10, 12, 13, 6, 11, 14, 7, 18\}$ .

$$S = E - D_{15} = \{14, 7, 18\}.$$

Asigna las fechas de término a las subactividades planeadas,  $f_{14} = f_7 = f_{18} = 4$ .

Determina el conjunto de actividades elegibles  $E_t = \{15, 3, 8, 19\}$ .

Paso 2 Actualiza  $t = 4$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{15, 3, 8, 19\}$ , cualquiera se puede planear de manera concurrente con alguna otra del mismo conjunto, como la subactividad 15 con las subactividades 3 y 8, la subactividad 19 con la subactividad 8 o 3, por lo que no se pueden planear subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 7, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 5$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 2$ .

Determina el conjunto  $D(5)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 2; de manera que:  $D_{18} = \{15\}, D_{19} = \{3\}, D_{20} = \{8\}, D_{21} = \{19\}$ , donde  $D(5) = \{D_{18}, D_{19}, D_{20}, D_{21}\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned} l_{18} &= 4 + 1 + \max_{i \in D_{18}} \{q_{15}\} = 4 + 1 + \max_{i \in D_{18}} \{3\} = 8, \\ l_{19} &= 4 + 1 + \max_{i \in D_{19}} \{q_3\} = 4 + 1 + \max_{i \in D_{19}} \{2\} = 7, \\ l_{20} &= 4 + 1 + \max_{i \in D_{20}} \{q_8\} = 4 + 1 + \max_{i \in D_{20}} \{1\} = 6, \\ l_{21} &= 4 + 1 + \max_{i \in D_{21}} \{q_{19}\} = 4 + 1 + \max_{i \in D_{21}} \{2\} = 7. \end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_{20}$  por tener una cota inferior mínima de 6.

Se actualiza  $D(5) = D(5) - D_{20} = \{\{15\}, \{3\}, \{19\}\}$ .

Determina  $lb_5 = \max\{lb_4, l_{20}\} = \max\{7, 6\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_{20} = \{1, 2, 5, 9, 10, 12, 13, 6, 11, 14, 7, 18, 15, 3, 19\}$ .

$$S = E - D_{20} = \{15, 3, 19\}.$$

Asigna las fechas de término a las subactividades planeadas,  $f_{15} = f_3 = f_{19} = 5$ .

Determina el conjunto de actividades elegibles  $E_t = \{16, 4, 8, 20\}$ .

Paso 2 Actualiza  $t = 5$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 De las subactividades en el conjunto de elegibles  $E_t = \{16, 4, 8, 20\}$ , cualquiera se puede planear de manera concurrente con alguna otra del mismo conjunto, como la subactividad 16 con las subactividades 4 y 8, la subactividad 20 con la subactividad 8 o 4, por lo que no se pueden planear subactividades en este paso.

Paso 4 Determina la cantidad de unidades que las subactividades en  $E_t$  requieren del único recurso, tal cantidad es 7, como la disponibilidad del recurso es de sólo 5 unidades, ocurre un conflicto de recursos.

Paso 5 Actualiza el nivel del árbol en  $p = 6$ .

Determina la cantidad de unidades necesarias requeridas por las subactividades elegibles, lo cual es  $c = 2$ .

Determina el conjunto  $D(6)$  como los subconjuntos de actividades elegibles tal que la suma de unidades que requieren del único recurso es al menos 2; de manera que:  $D_{22} = \{16\}, D_{23} = \{4\}, D_{24} = \{8\}, D_{25} = \{20\}$ , donde  $D(6) = \{D_{22}, D_{23}, D_{24}, D_{25}\}$ .

Para cada conjunto se calcula la cota inferior mínima:

$$\begin{aligned} l_{22} &= 5 + 1 + \max_{i \in D_{22}} \{q_{16}\} = 5 + 1 + \max_{i \in D_{22}} \{2\} = 8, \\ l_{23} &= 5 + 1 + \max_{i \in D_{23}} \{q_4\} = 5 + 1 + \max_{i \in D_{23}} \{1\} = 7, \\ l_{24} &= 5 + 1 + \max_{i \in D_{24}} \{q_8\} = 5 + 1 + \max_{i \in D_{24}} \{1\} = 7, \\ l_{25} &= 5 + 1 + \max_{i \in D_{25}} \{q_{20}\} = 5 + 1 + \max_{i \in D_{25}} \{1\} = 7. \end{aligned}$$

De acuerdo al algoritmo se elige el conjunto  $D_{23}$  por tener una cota inferior mínima de 7.

Se actualiza  $D(6) = D(6) - D_{23} = \{\{16\}, \{8\}, \{20\}\}$ .

Determina  $lb_6 = \max\{lb_5, l_{23}\} = \max\{7, 7\} = 7$ .

Guarda la planeación actual y el conjunto elegible.

Paso 6 Actualiza la planeación actual:  $PS_t = PS_t + E_t - D_{23} = \{1, 2, 5, 9, 10, 12, 13, 6, 11, 14, 7, 18, 15, 3, 19, 16, 8, 20\}$ .

$$S = E - D_{23} = \{16, 8, 20\}.$$

Asigna las fechas de término a las subactividades planeadas,  $f_{16} = f_8 = f_{20} = 6$ .

Determina el conjunto de actividades elegibles  $E_t = \{17, 4\}$ .

Paso 2 Actualiza  $t = 6$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 Sólo restan dos subactividades que se pueden planear de manera concurrente, las subactividades 4 y 17, como la subactividad 4 pertenece a la actividad 3 y de acuerdo al algoritmo, sólo se planean tantas subactividades de 7 (la actividad a la que pertenece la subactividad 17) como hayan subactividades de 3, de donde sólo se planean las únicas subactividades restantes,  $PS_t = \{1, 2, 5, 9, 10, 12, 13, 6, 11, 14, 7, 18, 15, 3, 19, 16, 8, 20, 4, 17\}$ ,  $S = \{4, 17\}$ .

Asigna las fechas de término a las subactividades planeadas,  $f_4 = f_{17} = 7$ .

Determina el conjunto de actividades elegibles  $E_t = \{21\}$ .

Como no se cumple que  $t + 1 + \max_{i \in E} \{4, 17\} = 8 \geq T$ , ir al *Paso 2*.

Paso 2 Actualiza  $t = 7$ .

La planeación actual no es dominada por alguna anterior, por lo que se guarda.

Paso 3 La subactividad 21 no se puede planear al mismo tiempo con alguna otra subactividad, por lo que se planea inmediatamente,  $PS_t = \{1, 2, 5, 9, 10, 12, 13, 6, 11, 14, 7, 18, 15, 3, 19, 16, 8, 20, 4, 17, 21\}$ ,  $S = \{21\}$ .

Asigna la fecha de término a la subactividad planeada,  $f_{21} = 7$ .

Determina el conjunto de actividades elegibles  $E_t = \emptyset$ .

Como  $7 < T$ , ir al *Paso 2*.

Paso 2 La subactividad ficticia 21 ya fue planeada, por lo que  $T = f_{21} = 7$ .

Como  $T = lb_0 = 7$ , entonces el algoritmo termina, con la planeación de subactividades obtenida.

En la Figura 3.5 se presenta, esquemáticamente, la planeación final de las subactividades.

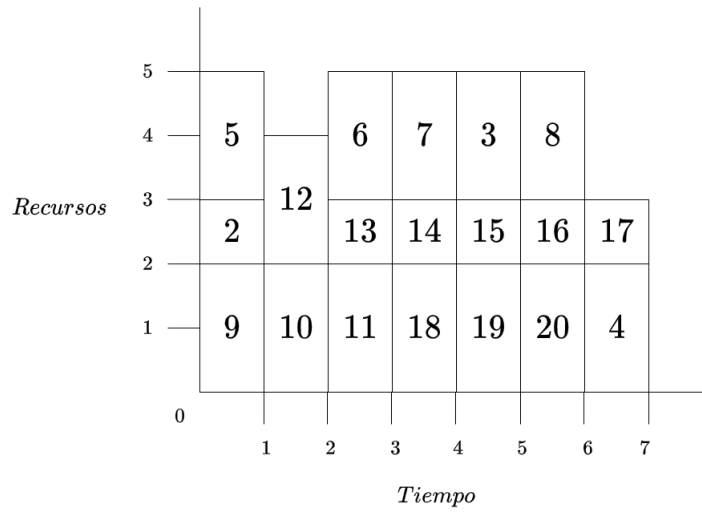


Figura 2.5: Planeación al nivel de subactividades

Y en la Figura 3.6 se presenta la planeación considerando las actividades originales (completas).

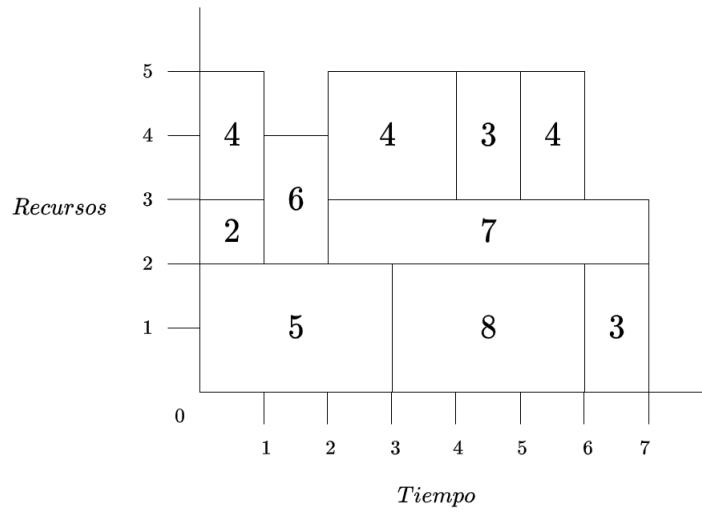


Figura 2.6: Planeación al nivel de actividades



### 2.2.2. Conclusiones del proceso Branch and Bound

Este proceso se enfoca en encontrar planeaciones cuyo tiempo de finalización es el número más grande de actividades entre las dos actividades ficticias (esto de acuerdo al *Paso 2: AUMENTO* ) o bien, ir retrocediendo en el árbol de planeaciones (mientras se pueda) buscando alguna planeación , con la cual se pueda obtener un tiempo distinto realizando los pasos anteriores.

La dificultad de realizar este proceso se encuentra en el *Paso 5: CONFLICTO DE RECURSOS*, al calcular el Conjunto de retraso  $D(p)$ , calcularlo de manera óptima es lo mas difícil del proceso, porque lo ideal es encontrar subconjuntos (alternativas de retraso) del menor tamaño posible, así realizando el *Paso 6: RETRASO*, se descartarán de la planeación actual a las subactividades de un cierto subconjunto en  $D(p)$ , por lo que si se descarta un subconjunto pequeño, entonces es posible planear más actividades en cada tiempo.

Por lo que en este paso se deben encontrar subconjuntos de subactividades  $D_q$  de tamaño mínimo tal que  $\sum_{i \in D_q} r_{ik} \geq c_k$ .

Además de que ésta es la parte no determinista del algoritmo, pues es posible utilizar varios subconjuntos de subactividades elegibles que puedan resolver un conflicto de recursos.

Por lo que obtener resultados satisfactorios utilizando este algoritmo, depende de la implementación del *Paso 5: CONFLICTO DE RECURSOS*.

## Capítulo 3

# Recocido Simulado

La metaheurística de Recocido Simulado creada por Scott Kirkpatrick, Daniel Gelatt y Mario P. Vecchi en 1983, además de Cerny en 1985, se inspira en el proceso de recocido de metales, donde el objetivo es obtener un material con buena estructura física, el proceso del recocido consiste en calentar a altas temperaturas el material, hasta el punto de fusión donde el material alcanza un estado líquido, con lo cual, sus átomos se desplazan fácilmente hacia nuevas posiciones reordenando la estructura del material, una vez alcanzada la temperatura planeada, el material se debe enfriar, si el enfriamiento es lento, entonces los átomos logran obtener una posición correcta, por otro lado, si el enfriamiento es rápido, entonces el material presenta inconsistencias y deformaciones. De la simulación de este proceso, es que recibe el nombre esta metaheurística.

Esta metaheurística usa como heurística subordinada a la búsqueda local, de manera que mientras se realiza esta búsqueda, se puede salir de un óptimo local aceptando peores soluciones, esto de acuerdo a la temperatura actual, pues cuando ésta es considerablemente alta, la probabilidad de salir de un óptimo local también es alta, por lo que, al tener temperaturas altas, es fácil moverse entre soluciones con distinto costo, aludiendo al movimiento de los átomos en el proceso del recocido de metales.

### 3.1. Descripción

Sea  $S$  el conjunto de todas las soluciones posibles para un problema de optimización  $\mathcal{P}$ .

El objetivo de la metaheurística es encontrar un óptimo global  $i^* \in S$ , tal que  $f(i^*) \leq f(j)$ ,  $\forall j \in S$ .

La metaheurística empieza con una solución inicial  $i \in S$  que puede ser obtenida de manera simple, además de una temperatura inicial  $T$ .

Después, revisa aleatoriamente un número determinado de vecinos  $N_T$ , donde para cada vecino  $j$ , obtiene la diferencia entre su función de costo  $f(j)$  con  $f(i)$ . Esto es:

$$\Delta = f(j) - f(i)$$

Se tienen dos casos.

- Si  $\Delta \leq 0$ .

El costo de  $j$  es mejor y por lo tanto, se acepta a  $j$  como solución y se asigna a  $i$  como  $j$ , con lo cual,  $j$  es la mejor solución hasta el momento.

- En otro caso, se acepta a  $j$  con probabilidad  $e^{-\Delta/T}$ .

De la evaluación anterior es como esta metaheurística buscar salir de posibles óptimos locales, aceptando peores soluciones de acuerdo a una probabilidad.

Una vez que termina de revisar un número determinado de vecinos, decrementa la temperatura actual usando una constante  $\alpha$ ,  $0 < \alpha < 1$ , que indica un porcentaje de decremento, con la cual en cada decremento la temperatura  $T$  pasa a  $\alpha T$ .

Esta serie de pasos se repiten hasta que la temperatura disminuye a un valor  $\epsilon$  o menor, con  $\epsilon > 0$  que es una constante especial, si esta constante es muy baja la metaheurística podría tardar bastante en terminar.

De acuerdo al comportamiento de la metaheurística, para lograr diversificar la búsqueda acepta peores soluciones, que conducen hacia regiones poco exploradas y cuando es necesario intensificar la búsqueda, toma a la mejor solución obtenida en el vecindario de acuerdo a la estrategia usada en la búsqueda local.

En el Algoritmo 11 se presenta un pseudo-código para la metaheurística descrita.

---

**Algoritmo 11** RECOCIDO SIMULADO

---

**Entrada:** Solución inicial  $i$ .

Temperatura inicial  $T$ ,  $T > 0$ .

Cantidad de vecinos a revisar por nivel de temperatura  $N_T$ ,  $0 < N_T$ .

Constante de decremento  $\alpha$ ,  $0 < \alpha < 1$ .

Constante de temperatura mínima  $\epsilon$ ,  $0 < \epsilon$ .

**Salida:** Mejor solución encontrada.

```
1: while  $T > \epsilon$  do
2:    $n \leftarrow 0$ 
3:   while  $n < N_T$  do
4:     Genera aleatoriamente un vecino  $j$  de  $i$ 
5:      $\Delta \leftarrow f(j) - f(i)$ 
6:     if  $\Delta \leq 0$  then
7:        $i \leftarrow j$ 
8:     else
9:       if  $\text{random}(0, 1) < e^{-\Delta/T}$  then
10:         $i \leftarrow j$ 
11:       end if
12:     end if
13:      $n \leftarrow n + 1$ 
14:   end while
15:    $T \leftarrow \alpha T$ 
16: end while
17: return  $i$ 
```

---

### 3.2. Aplicado a PRCPSP

Dado un proyecto de actividades, es necesario crear un nuevo proyecto a nivel de subactividades, donde cada actividad se interrumpe a los más  $m$  veces, si  $m$  es mayor o igual a la duración de la actividad, entonces la actividad se divide en subactividades de duración uno. Las subactividades resultantes utilizan la misma cantidad de recursos que la actividad original.

Para utilizar la metaheurística en el PRCPSP, es necesario definir cuatro cosas:

- Representación de la solución.
- Generación de soluciones vecinas.
- La función de costo.
- Solución inicial.

#### Representación de la solución

Una solución se representa mediante dos arreglos, el primero es un arreglo  $L$  de subactividades, donde las únicas subactividades que permanecen en el mismo lugar para cualquier solución son las dos ficticias, la subactividad 1 en la primer posición y la subactividad  $n$  en la última posición, para la  $i$ -ésima subactividad  $u$ , el índice  $j$  de la subactividad antecesora más cercana a  $u$  es menor y por el otro lado, el índice  $k$  de la subactividad sucesora más cercana a  $u$  es mayor, todo esto según el conjunto  $H$  de restricciones. En este arreglo, solo se representa el orden en el que son planeadas las subactividades.

El segundo arreglo  $L'$  tiene los tiempos en que se inicia cada subactividad, de manera que la  $i$ -ésima subactividad en  $L$  se planea en el  $i$ -ésimo tiempo en  $L'$ .

Se presenta ejemplo de una planeación y su solución en la Figura 3.1 (a) y (b), respectivamente.

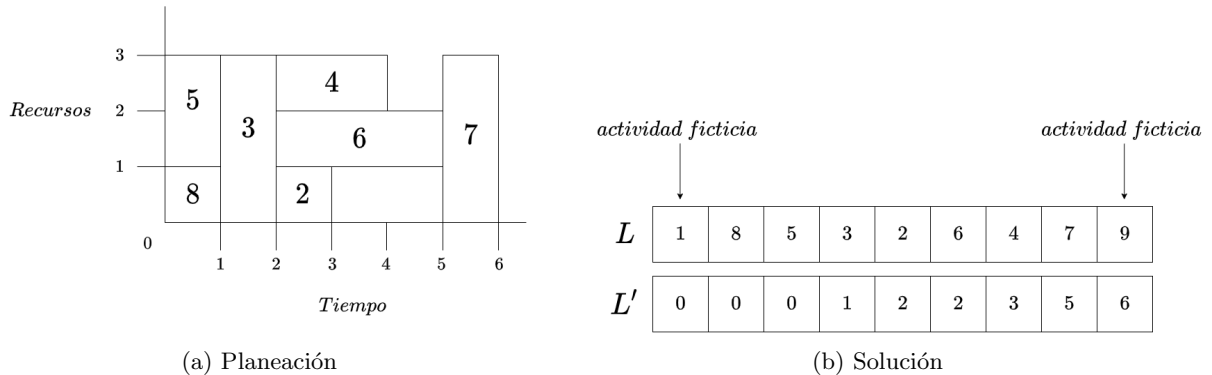


Figura 3.1: Representación de una planeación

#### Generación de soluciones vecinas

Dada una solución, una solución vecina se genera de la siguiente forma:

Dado el arreglo  $L$  de longitud  $n$ , se toma una posición  $i$  del subarreglo  $L[2..n-3]$  pues en las posiciones 0 y  $n-1$  se encuentran las subactividades ficticias que no se pueden mover de posición,

por otro lado, las posiciones 1 y  $n - 2$  tampoco se pueden escoger por lo que se presenta a continuación.

Sea  $v$  la subactividad en la posición  $i$  escogida, sean  $u, w$  las subactividades en las posiciones  $i - 1, i + 1$ , respectivamente, tal que estas subactividades no son predecesoras ni sucesoras a  $u$  de acuerdo a las restricciones en  $H$  y entre estas subactividades no se tiene una relación de precedencia, entonces se intercambian de posiciones rotando estos tres elementos un lugar a la derecha (únicamente sobre las posiciones  $i - 1, i, i + 1$ ), obteniendo así una planeación correspondiente a la solución vecina.

$$\begin{aligned} L[i - 1] &= w \\ L[i] &= u \\ L[i + 1] &= v \end{aligned}$$

Regresando a porque no se pueden escoger las posiciones 1 y  $n - 2$ , si éstas se escogen, entonces se pueden rotar de posición las dos subactividades ficticias porque son adyacentes en  $L$  a las subactividades correspondientes en las posiciones ya mencionadas.

En la Figura 3.2 se representa una solución aún sin rotar sus elementos, tomando al elemento en la posición  $i$ .

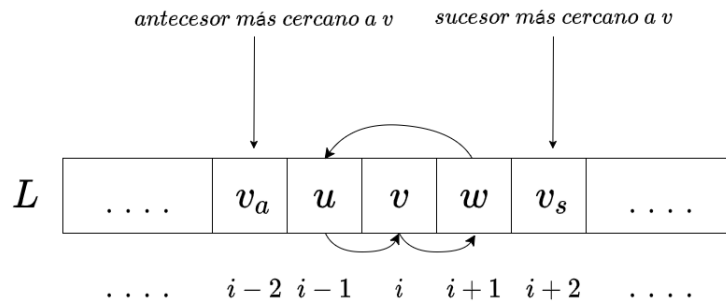


Figura 3.2: Solución inicial

En la Figura 3.3 se tiene una solución vecina después de rotar a los elementos en la figura anterior.

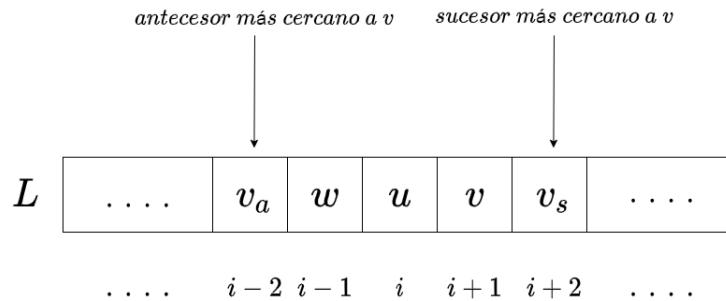


Figura 3.3: Nueva solución vecina

Retomando la planeación mostrada en la Figura 3.1 (a), si se toma la posición 4 y se realiza una rotación de sus elementos se obtiene la planeación mostrada en la Figura 3.4, la cual tarda una unidad más de tiempo.

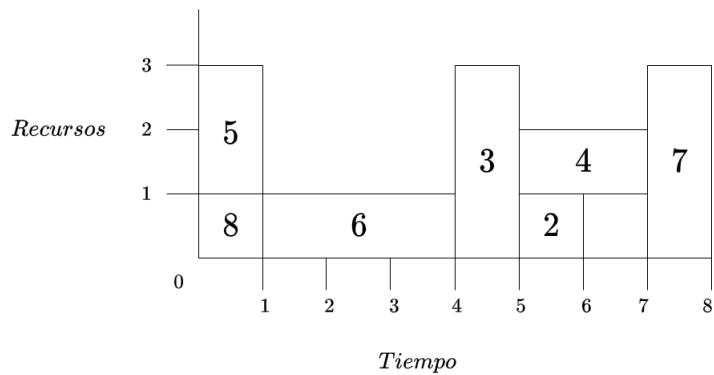


Figura 3.4: Nueva planeación

En la Figura 3.5 se tienen los arreglos  $L, L'$  que representan a la solución vecina de la solución en la Figura 3.1 (b).

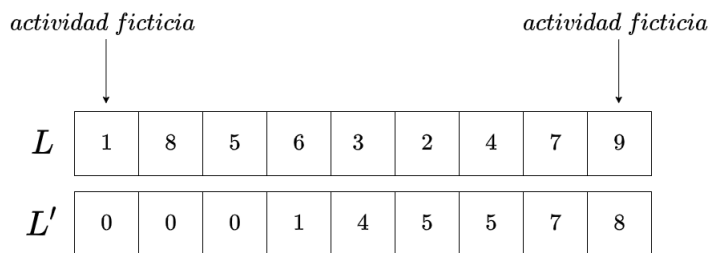


Figura 3.5: Arreglos modificados

### Función de costo

La función de costo  $f(i)$  para una solución  $i$  está dada por el tiempo en el que se termina la subactividad ficticia  $n$  que es la última subactividad en planearse de acuerdo a la definición del problema. Por lo que, la función a optimizar es el tiempo en que se planea la subactividad ficticia  $n$ , esto es  $f_{n,0}$ .

Para obtener este tiempo es necesario calcular la arreglo  $L'$  en la representación de la solución ya que contiene los tiempos en que se planea cada subactividad.

El Algoritmo 12 calcula el arreglo  $L'$ .

---

**Algoritmo 12** CALCULA  $L'$ 

---

**Entrada:** Arreglo  $L$  con las subactividades ordenadas según el orden en que se planean.

**Salida:** Arreglo  $L'$  con los tiempos en que se planea cada actividad.

```
1:  $n \leftarrow |L|$ 
2:  $L' \leftarrow []$ 
3:  $L'[0] \leftarrow 0$ 
4:  $k \leftarrow 0$ 
5: for  $i = 1$  to  $n - 2$  do
6:    $subactividad\_actual \leftarrow L[i]$ 
7:   if Se puede planear la  $subactividad\_actual$  en el tiempo  $k$  then
8:      $L'[i] = k$ 
9:   else
10:    // De entre las actividades predecesoras a la actual, toma el mayor de los tiempos en que finalizan
11:     $k' \leftarrow maximo\_tiempo\_antecesoras\_planeadas(i)$ 
12:     $k \leftarrow \max\{k, k'\}$ 
13:    // Revisa si en el tiempo  $k$  existe un conflicto de recursos al planear la actividad actual
14:    while  $conflicto\_planear(subactividad\_actual, k)$  do
15:       $k \leftarrow k + 1$ 
16:    end while
17:     $L'[i] = k$ 
18:  end if
19: end for
20:  $subactividad\_final \leftarrow L[n - 2]$ 
21:  $L'[n - 1] = k + subactividad\_final.duracion$ 
22: return  $L'$ 
```

---

Una vez obtenido el arreglo  $L'$  de una solución  $i$ , su función de costo se obtiene como:

$$f(i) = L'[n - 1]$$

## Soluciones iniciales

Dado un proyecto de actividades, una planeación inicial se obtiene realizando un recorrido BFS sobre la gráfica del proyecto. Cada vez que se itera sobre una actividad, se revisa si las actividades que le preceden a ésta, se hayan planeado, si esto se cumple, se toma el mayor de entre el tiempo actual y el máximo tiempo de término de sus actividades antecesoras, una vez obtenido el tiempo, se calcula a partir de éste, un tiempo en el que no exista un conflicto de recursos al intentar planear la actividad actual con las actividades que ya han sido planeadas, finalmente se guardan las actividades sucesoras que no han sido planeadas y para las cuales, sus actividades predecesoras hayan sido planeadas. En el Algoritmo 13 se muestra el pseudo-código para obtener soluciones iniciales.

---

**Algoritmo 13** PLANEACIÓN INICIAL
 

---

```

1:  $t \leftarrow 0$ 
2:  $actividad\_inicial.tiempo \leftarrow t$ 
3:  $Q \leftarrow \emptyset$ 
4:  $Q.push(actividad\_inicial)$ 
5: while not  $Q.empty()$  do
6:    $u \leftarrow Q.first()$ 
7:    $t' \leftarrow \text{maximo\_tiempo\_antecesoras\_planeadas}(u)$ 
8:    $t \leftarrow \text{máx}\{t, t'\}$ 
9:   while  $\text{conflicto\_planear}(u, t)$  do
10:     $t \leftarrow t + 1$ 
11:   end while
12:   // Planea la actividad actual en el tiempo  $t$ 
13:    $u.tiempo \leftarrow t$ 
14:   // Agrega las actividades sucesoras para las cuales sus actividades antecesoras han sido planeadas
15:   for all  $v$  in  $u.subactividades\_sucesoras$  do
16:     if not  $v.planeada()$   $\&\&$   $v.antecesoras\_planeadas()$  then
17:        $Q.insert(v)$ 
18:     end if
19:   end for
20: end while
21:  $actividad\_final.tiempo \leftarrow t$ 

```

---

Por ejemplo, para el proyecto mostrado en la Figura 3.6, se describe adelante la manera en la que funciona el algoritmo para obtener una solución inicial.

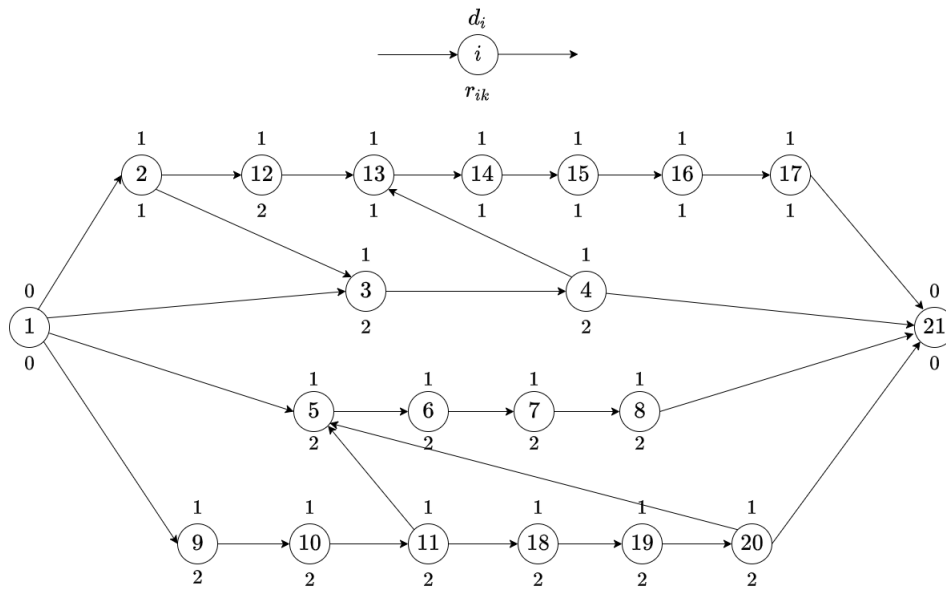


Figura 3.6: Gráfica de proyecto



Iteración 1 Se revisa la subactividad ficticia inicial 1, planeándose en el tiempo 0, además de agregar a las subactividades 2 y 9 a  $Q$ . En la Figura 3.7 se muestra el recorrido hasta esta iteración.

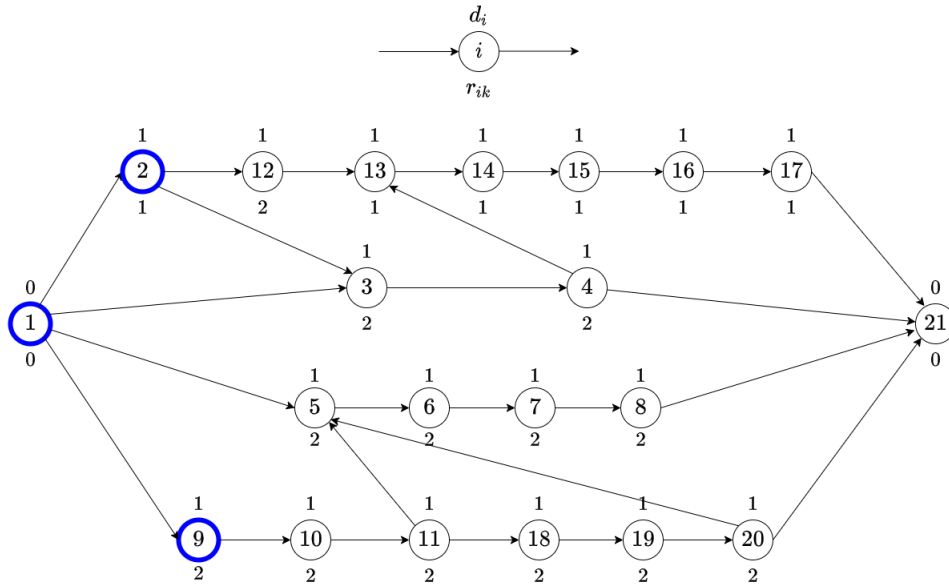


Figura 3.7: Iteración 1 del recorrido BFS

Iteración 2 Se revisa la subactividad 2, planeándose en el tiempo 0, además de agregar a las subactividades 12 y 3 a  $Q$ . En la Figura 3.8 se muestra el recorrido hasta esta iteración.

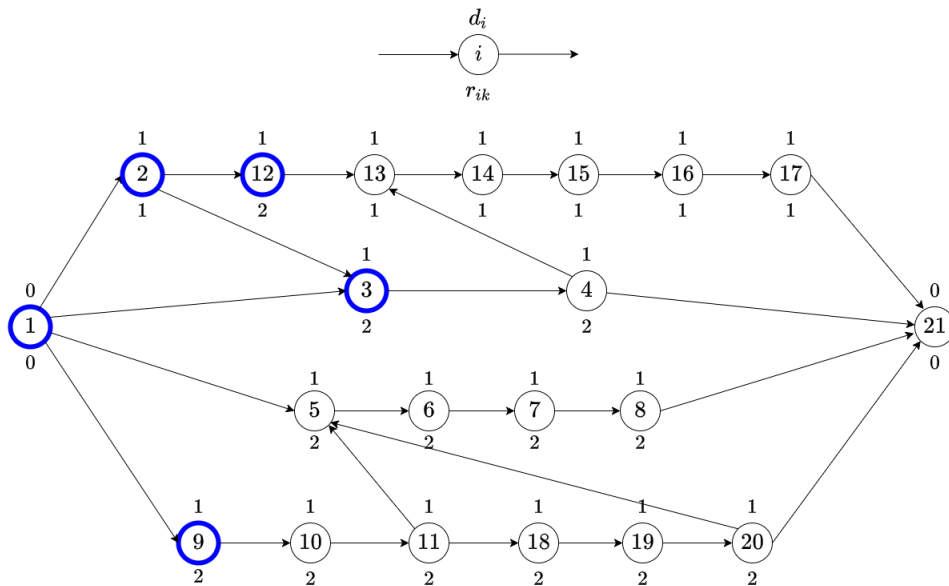


Figura 3.8: Iteración 2 del recorrido BFS

Iteración 3 Se revisa la subactividad 9, planeándose en el tiempo 0, además de agregar a la subactividad 10 a  $Q$ . En la Figura 3.9 se muestra el recorrido hasta esta iteración.

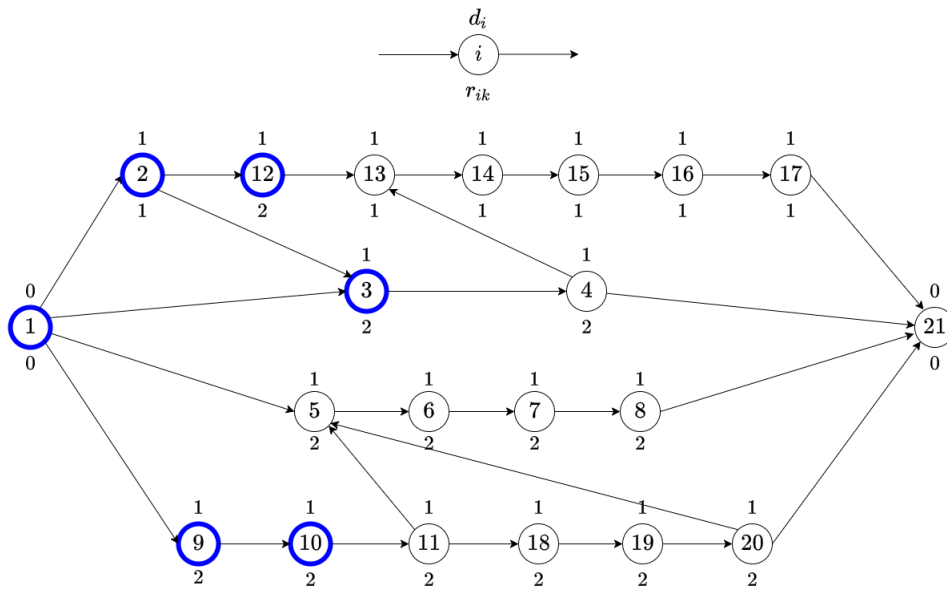


Figura 3.9: Iteración 3 del recorrido BFS

Iteración 4 Se revisa la subactividad 12, planeándose en el tiempo 1, la única subactividad sucesora a ésta, es la subactividad 13, pero aún no se puede agregar a  $Q$  porque la subactividad 4 que es antecesora a ésta, aún no se ha planeado, por lo que no se agregan más subactividades a  $Q$ .

Iteración 5 Se revisa la subactividad 3, planeándose en el tiempo 1, además de agregar a la subactividad 4 a  $Q$ . En la Figura 3.10 se muestra el recorrido hasta esta iteración.

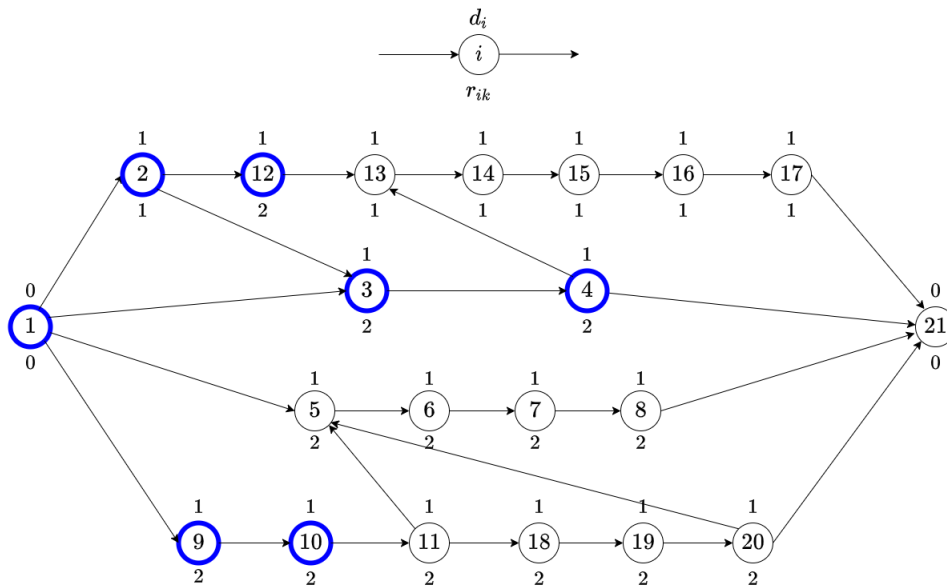


Figura 3.10: Iteración 5 del recorrido BFS

Iteración 6 Se revisa la subactividad 10, planeándose en el tiempo 2, además de agregar a la subactividad 11 a  $Q$ . En la Figura 3.11 se muestra el recorrido hasta esta iteración.

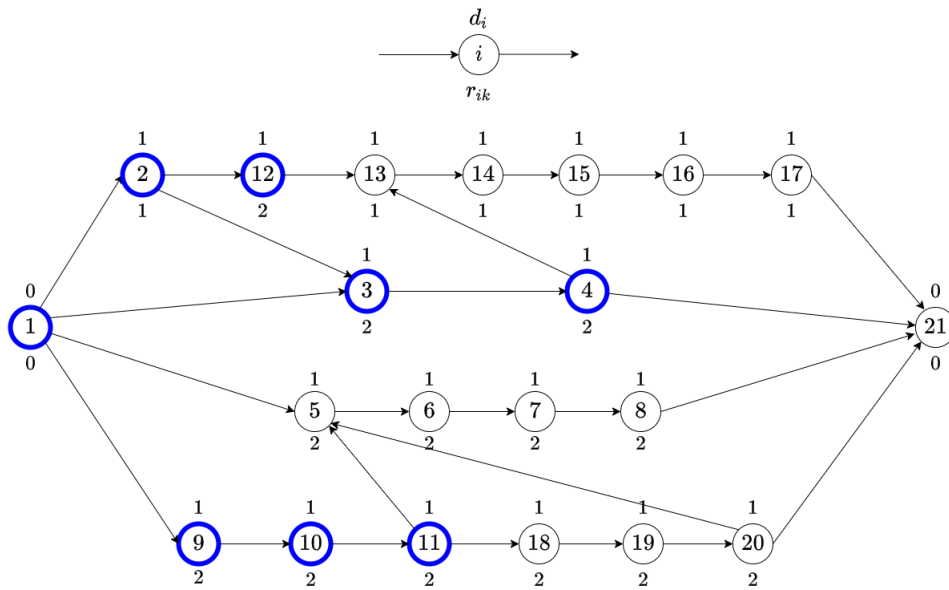


Figura 3.11: Iteración 6 del recorrido BFS

Iteración 7 Se revisa la subactividad 4, planeándose en el tiempo 2, además de agregar a la subactividad 13 a  $Q$ . En la Figura 3.12 se muestra el recorrido hasta esta iteración.

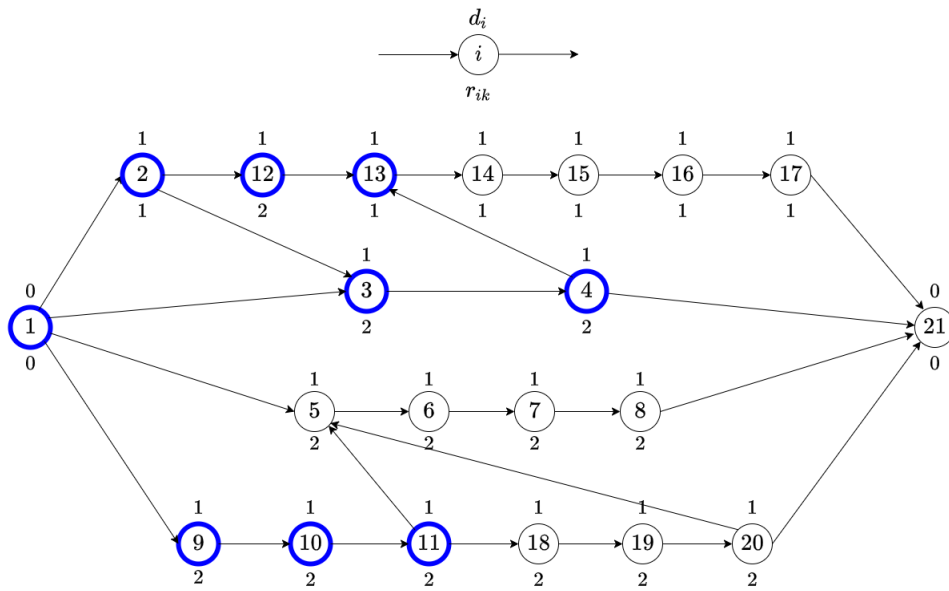


Figura 3.12: Iteración 7 del recorrido BFS

Iteración 8 Se revisa la subactividad 11, planeándose en el tiempo 3, además de agregar a la subactividad 18 a  $Q$ . En la Figura 3.13 se muestra el recorrido hasta esta iteración.

Iteración 9 Se revisa la subactividad 13, planeándose en el tiempo 3, además de agregar a la subactividad 14 a  $Q$ . En la Figura 3.14 se muestra el recorrido hasta esta iteración.

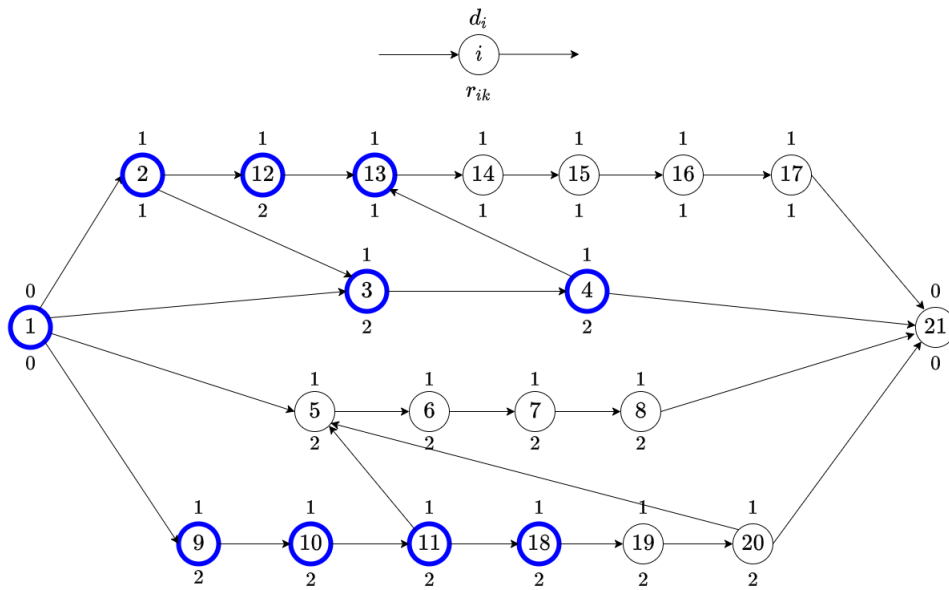


Figura 3.13: Iteración 8 del recorrido BFS

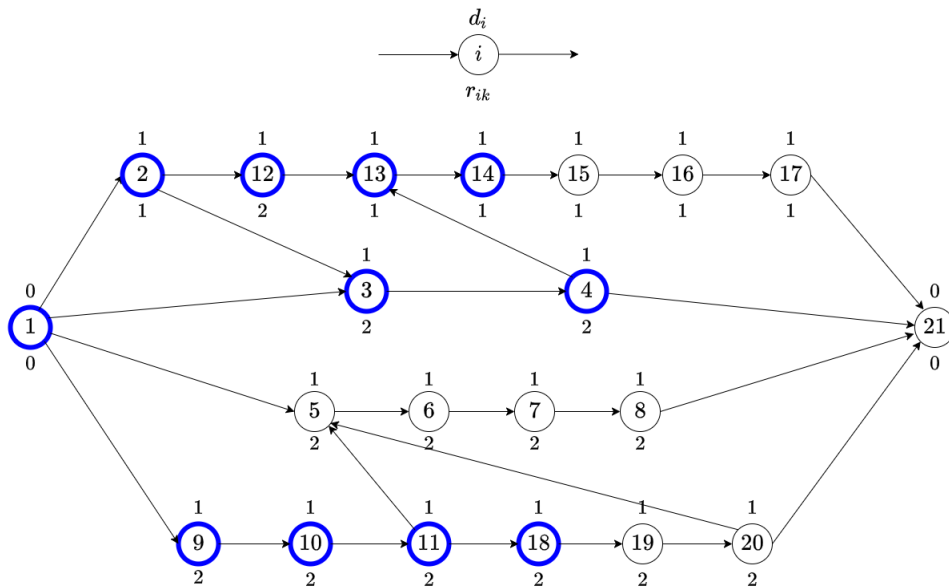


Figura 3.14: Iteración 9 del recorrido BFS

Iteración 10 Se revisa la subactividad 18, planeándose en el tiempo 4, además de agregar a la subactividad 19 a  $Q$ . En la Figura 3.15 se muestra el recorrido hasta esta iteración.

Iteración 11 Se revisa la subactividad 14, planeándose en el tiempo 4, además de agregar a la subactividad 15 a  $Q$ . En la Figura 3.16 se muestra el recorrido hasta esta iteración.

Iteración 12 Se revisa la subactividad 19, planeándose en el tiempo 5, además de agregar a la subactividad 20 a  $Q$ . En la Figura 3.17 se muestra el recorrido hasta esta iteración.

Iteración 13 Se revisa la subactividad 15, planeándose en el tiempo 5, además de agregar a la subactividad

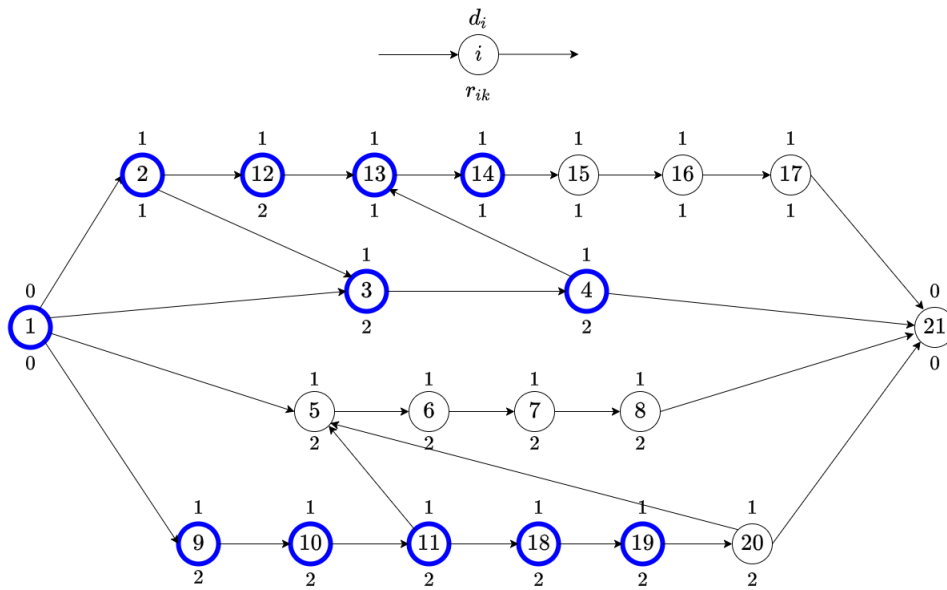


Figura 3.15: Iteración 10 del recorrido BFS

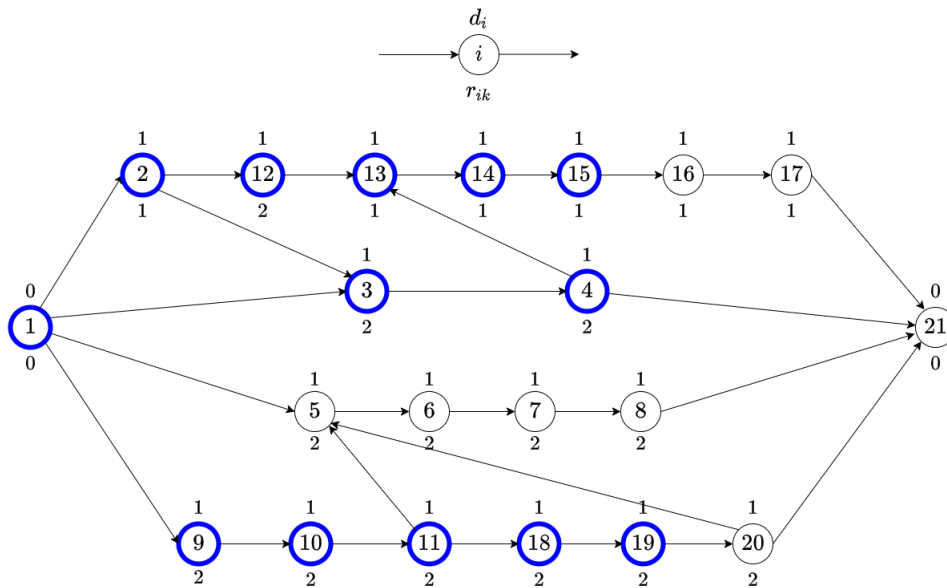


Figura 3.16: Iteración 11 del recorrido BFS

16 a  $Q$ . En la Figura 3.18 se muestra el recorrido hasta esta iteración.

Iteración 14 Se revisa la subactividad 20, planeándose en el tiempo 6, además de agregar a la subactividad 5 a  $Q$ . En la Figura 3.19 se muestra el recorrido hasta esta iteración.

Iteración 15 Se revisa la subactividad 16, planeándose en el tiempo 6, además de agregar a la subactividad 17 a  $Q$ . En la Figura 3.20 se muestra el recorrido hasta esta iteración.

Iteración 16 Se revisa la subactividad 5, planeándose en el tiempo 7, además de agregar a la subactividad 6 a  $Q$ . En la Figura 3.21 se muestra el recorrido hasta esta iteración.

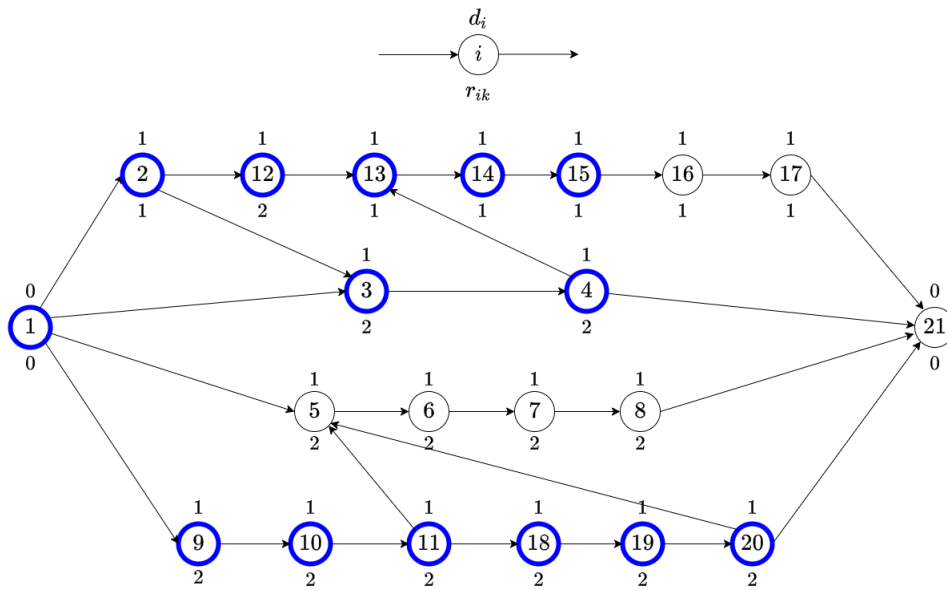


Figura 3.17: Iteración 12 del recorrido BFS

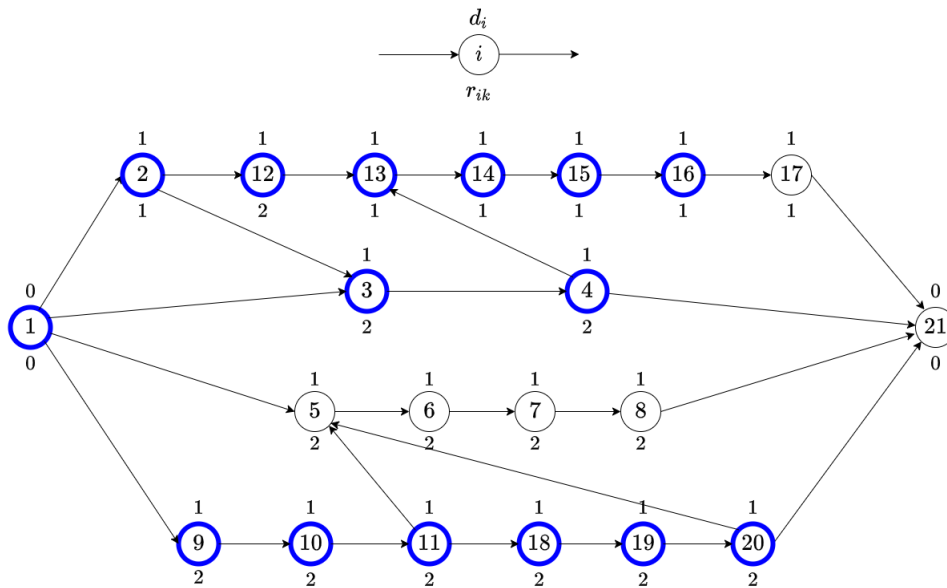


Figura 3.18: Iteración 13 del recorrido BFS

Iteración 17 Se revisa la subactividad 17, planeándose en el tiempo 7, la única subactividad sucesora a ésta, es la subactividad 21, pero aún no se puede agregar a  $Q$  porque la subactividad 8 que es antecesora a ésta, aún no se ha planeado, por lo que no se agregan más subactividades a  $Q$ .

Iteración 18 Se revisa la subactividad 6, planeándose en el tiempo 8, además de agregar a la subactividad 7 a  $Q$ . En la Figura 3.22 se muestra el recorrido hasta esta iteración.

Iteración 19 Se revisa la subactividad 7, planeándose en el tiempo 9, además de agregar a la subactividad 8 a  $Q$ . En la Figura 3.23 se muestra el recorrido hasta esta iteración.

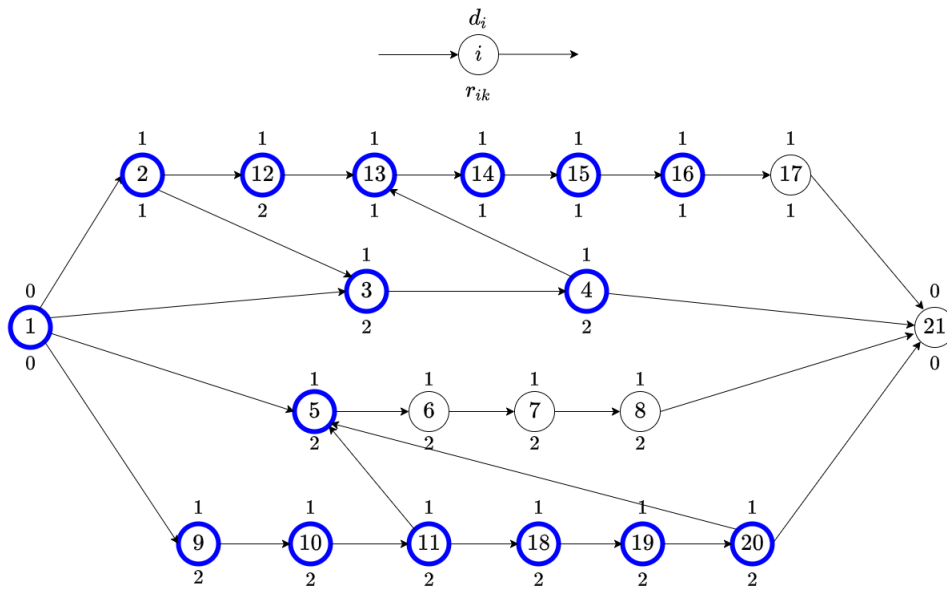


Figura 3.19: Iteración 14 del recorrido BFS

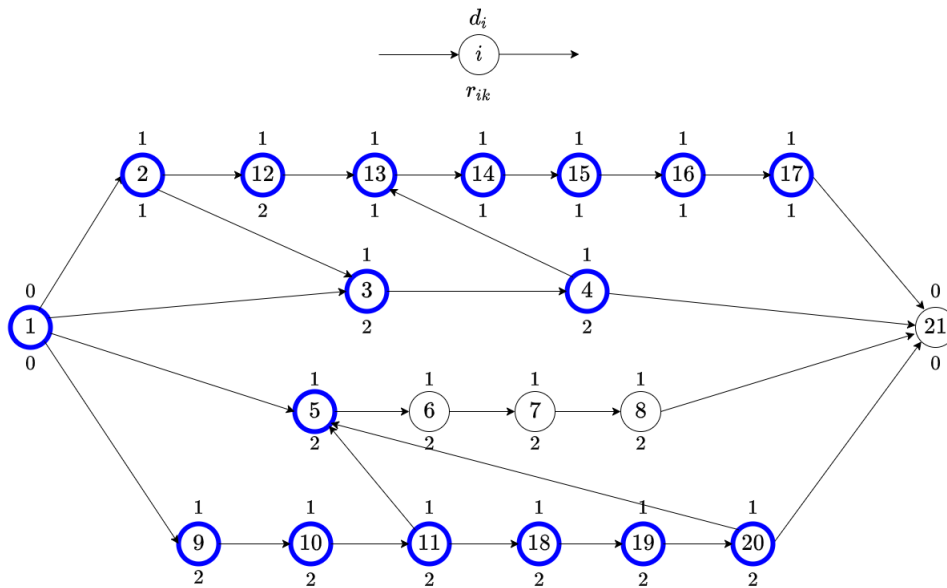


Figura 3.20: Iteración 15 del recorrido BFS

Iteración 20 Se revisa la subactividad 8, planeándose en el tiempo 10, además de agregar a la subactividad 21 a  $Q$ . En la Figura 3.24 se muestra el recorrido hasta esta iteración.

Iteración 21 Se revisa la subactividad 21, planeándose en el tiempo 11, como es la subactividad ficticia final, significa que todas las actividades del proyecto ya fueron planeadas, por tanto, se obtiene la planeación mostrada en la Figura 3.25.

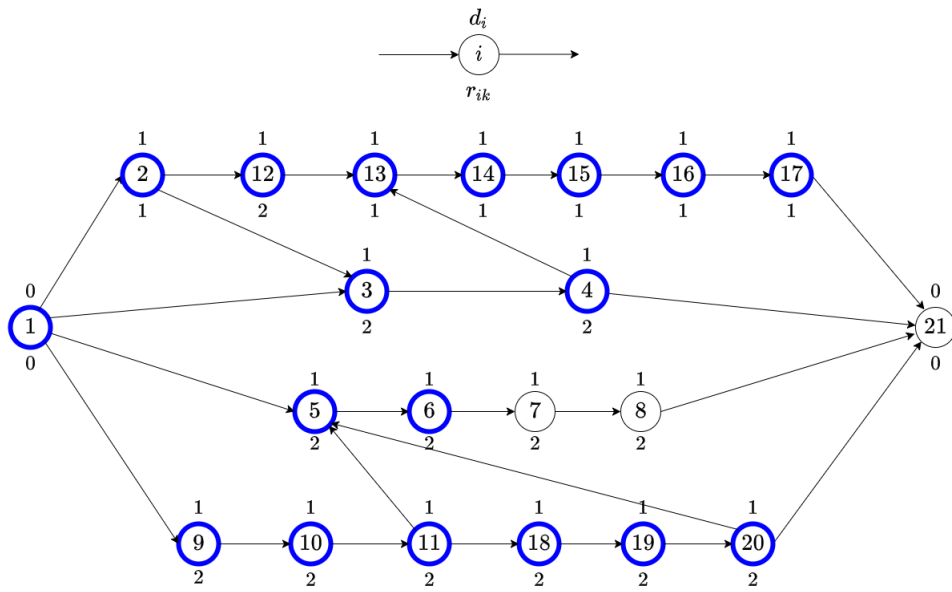


Figura 3.21: Iteración 16 del recorrido BFS

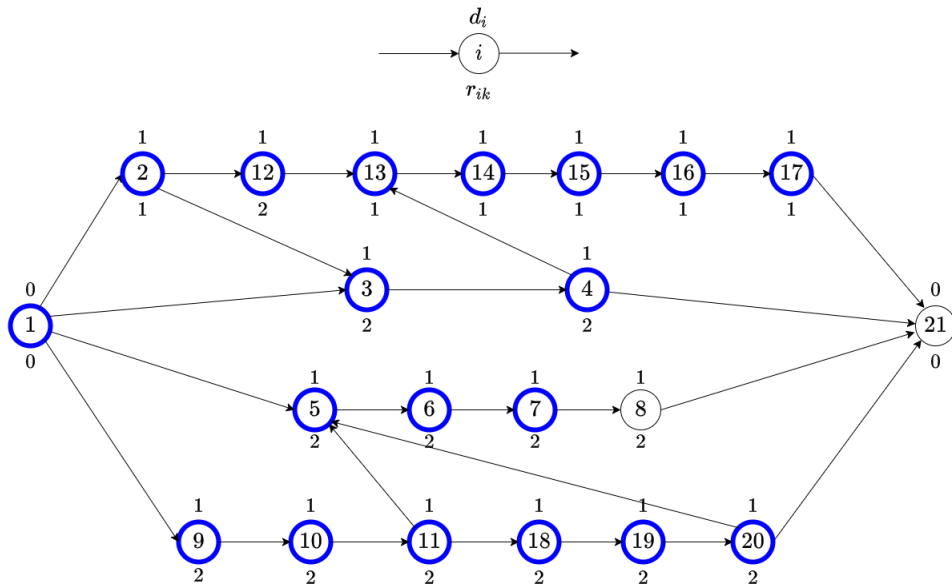


Figura 3.22: Iteración 18 del recorrido BFS



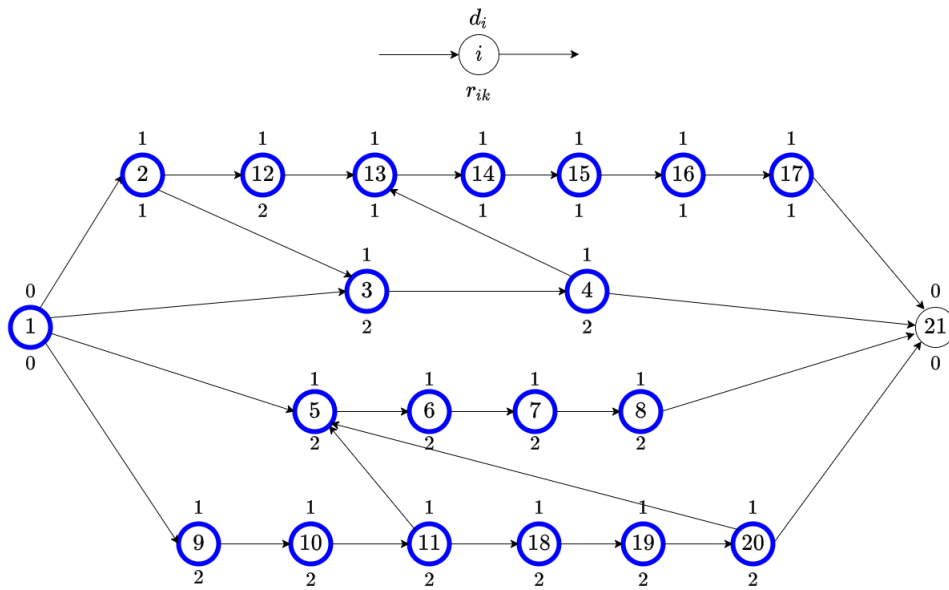


Figura 3.23: Iteración 19 del recorrido BFS

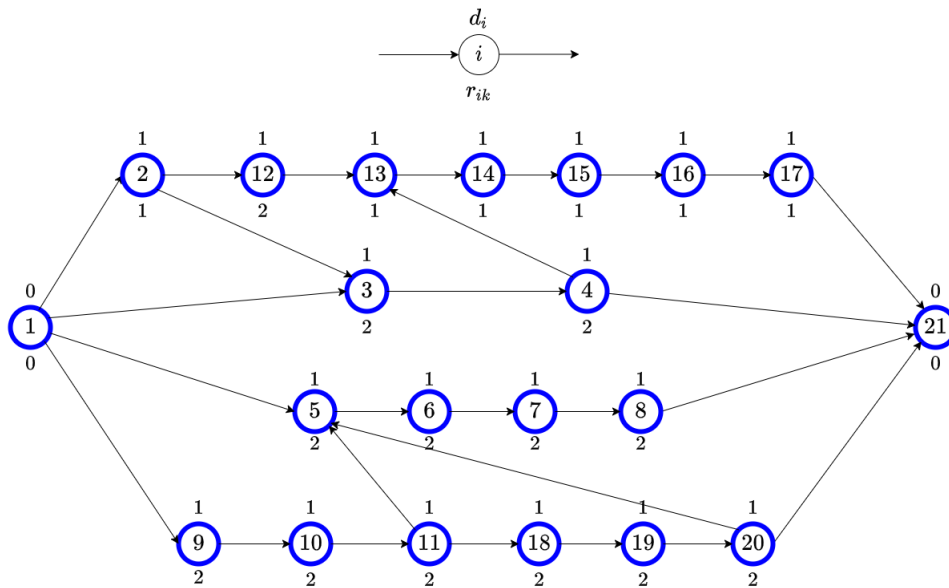


Figura 3.24: Iteración 20 del recorrido BFS

### Ejemplo del funcionamiento

Para mostrar el funcionamiento detallado de la metaheurística, se tiene el proyecto mostrado en la Figura 3.26 de 15 actividades (sin contar las dos ficticias 1 y 17), con un costo óptimo de 14 unidades, con 4 recursos de 3 unidades cada uno. Además de la planeación óptima conocida, mostrada en la Figura 3.27.

Para el proyecto mostrado anteriormente se utilizan los siguientes parámetros en la metaheurística:

- Temperatura inicial: 100

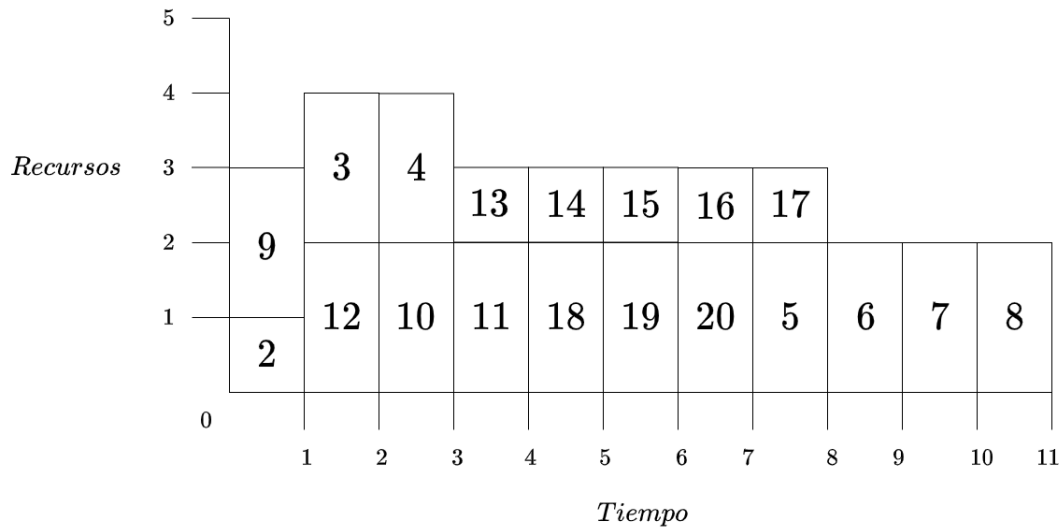


Figura 3.25: Planeación inicial obtenida

- Número de interrupciones: 1
- Vecinos a revisar por nivel de temperatura: 4
- Decremento de temperatura: 0.8
- Temperatura mínima: 0.1

A continuación la ejecución de la metaheurística, mostrando cada iteración en donde se modifica la solución actual después de revisar un conjunto de vecinos dado. En total se muestran 31 iteraciones.

Iteración 1 **Temperatura actual:** 100

**Solución actual:**

Al ser la primer iteración, primero se obtiene el proyecto a nivel de subactividades del proyecto original, el cual se muestra en la Figura 3.28. Después se utiliza el Algoritmo 13 sobre este proyecto obteniendo así una planeación inicial con costo 20, la cual se muestra en la Figura 3.29 (a), además de la representación de la solución en la Figura 3.29 (b).

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

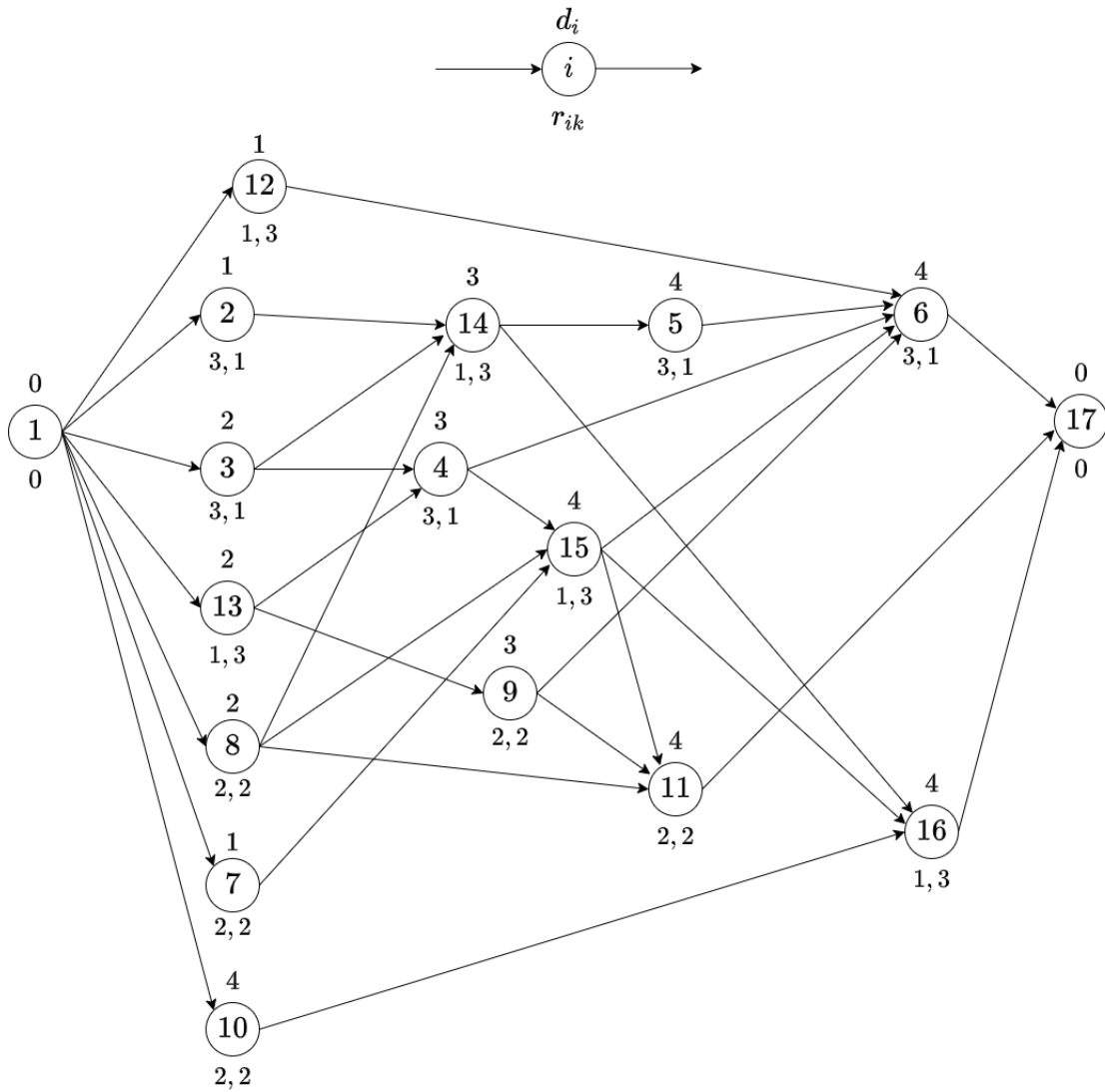


Figura 3.26: Proyecto de 15 actividades

- Vecino 3. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.990, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 2

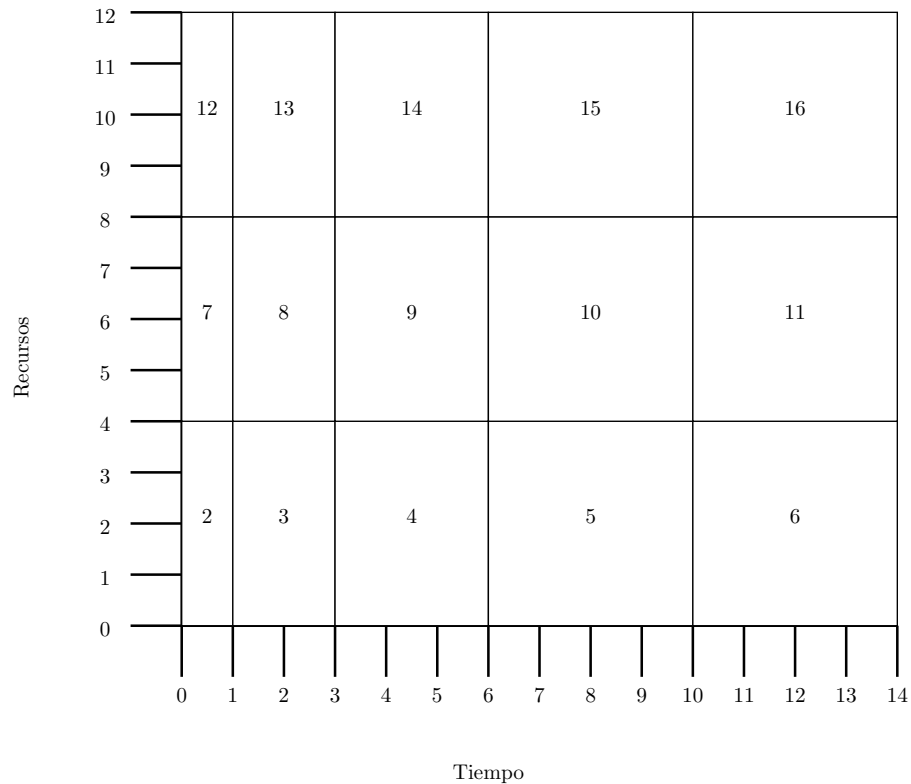


Figura 3.27: Planeación óptima del proyecto en la Figura 3.26

**Temperatura actual:** 80

**Solución actual:**

La solución actual al iniciar la iteración 2 es la solución vecina 4 con costo 21 de la iteración anterior y que se muestra en la Figura 3.30.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 23 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.975, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

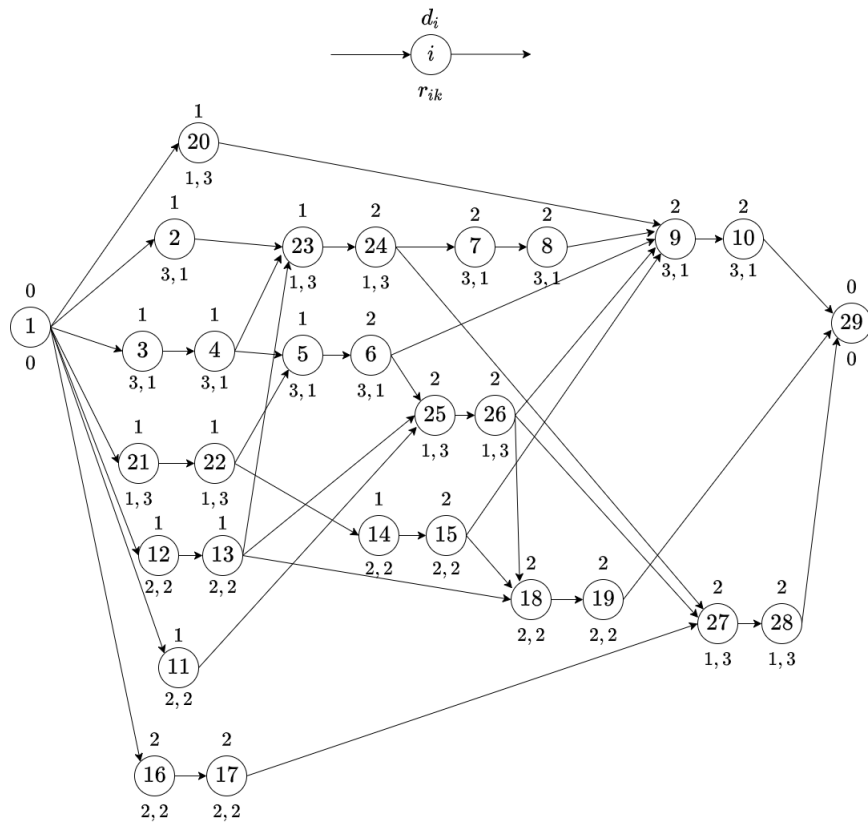


Figura 3.28: Proyecto a nivel de subactividades

Vecino 4. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 3 **Temperatura actual:** 64

**Solución actual:**

La solución actual al iniciar la iteración 3 es la solución vecina 4 con costo 23 de la iteración anterior y que se muestra en la Figura 3.31.

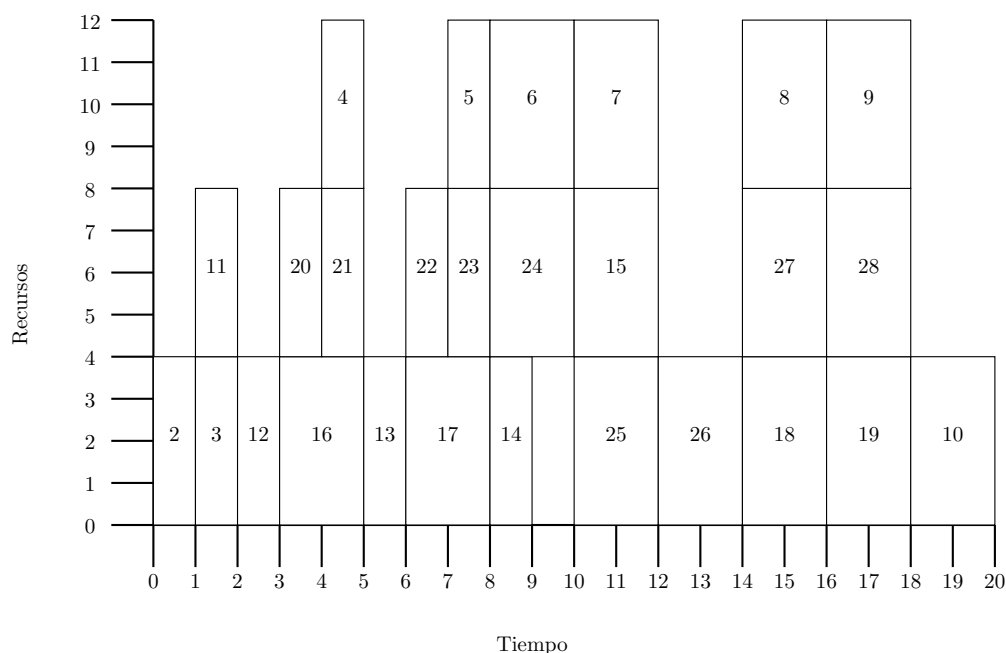
**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 26. Esta solución



(a) Planeación actual en la iteración 1

L	1	2	3	11	12	16	20	21	4	13	17	22	23	5	14	24	6	25	15	7	26	18	27	8	19	28	9	10	29
L'	0	0	1	1	2	3	3	4	4	5	6	6	7	7	8	8	8	10	10	10	12	14	14	14	16	16	16	18	20

(b) Representación de la solución actual en la iteración 1

Figura 3.29: Planeación y representación de la solución actual en la iteración 1

no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.954, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.

Iteración 4 **Temperatura actual:** 51.2

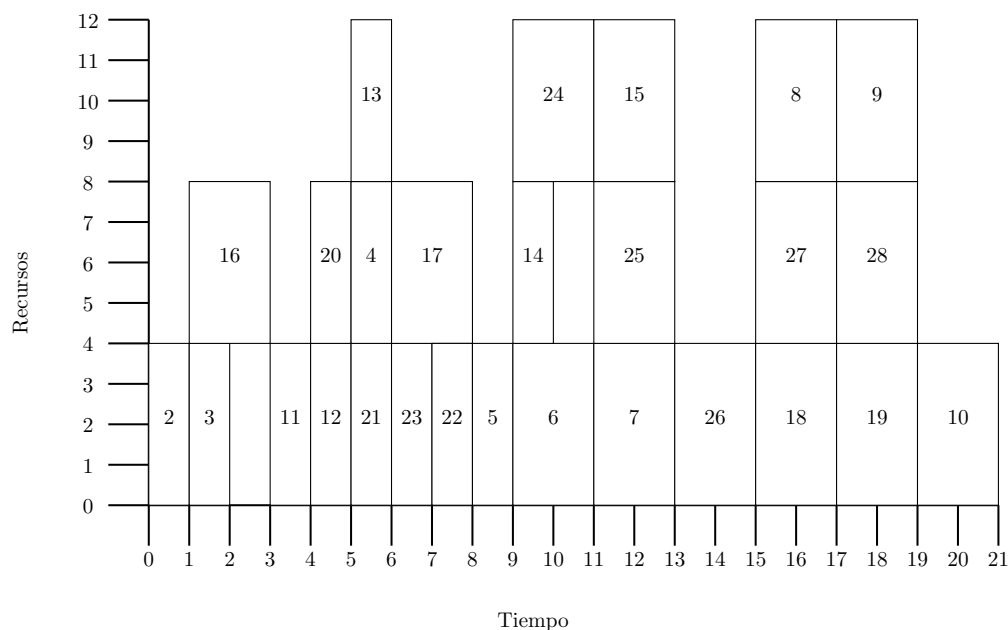
**Solución actual:**

La solución actual al iniciar la iteración 4 es la solución vecina 4 con costo 25 de la iteración anterior y que se muestra en la Figura 3.32.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución



(a) Planeación actual en la iteración 2

L	1	2	3	16	11	12	20	21	4	13	23	17	22	5	6	14	24	7	25	15	26	18	27	8	19	28	9	10	29
L'	0	0	1	1	3	4	4	5	5	5	6	6	7	8	9	9	9	11	11	11	13	15	15	15	17	17	17	19	21

(b) Representación de la solución actual en la iteración 2

Figura 3.30: Planeación y representación de la solución actual en la iteración 2

tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 23 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

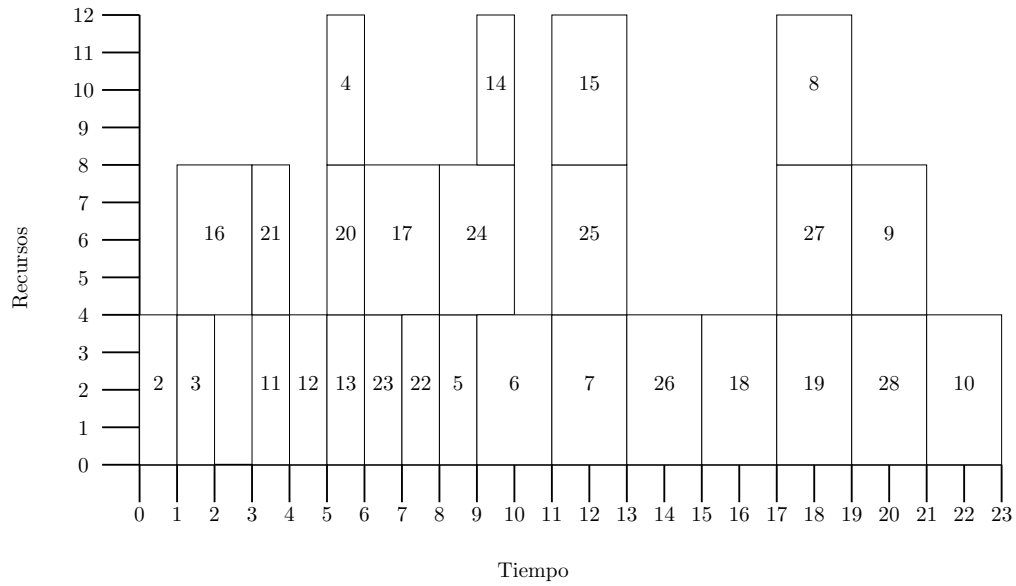
Vecino 3. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 5 **Temperatura actual:** 40.96

**Solución actual:**

La solución actual al iniciar la iteración 5 es la solución vecina 4 con costo 25 de la iteración anterior y que se muestra en la Figura 3.33.



(a) Planeación actual en la iteración 3

L	1	2	3	16	11	21	12	13	20	4	23	17	22	5	24	6	14	7	25	15	26	18	19	27	8	28	9	10	29
L'	0	0	1	1	3	3	4	5	5	5	6	6	7	8	8	9	9	11	11	11	13	15	17	17	17	19	19	21	23

(b) Representación de la solución actual en la iteración 3

Figura 3.31: Planeación y representación de la solución actual en la iteración 3

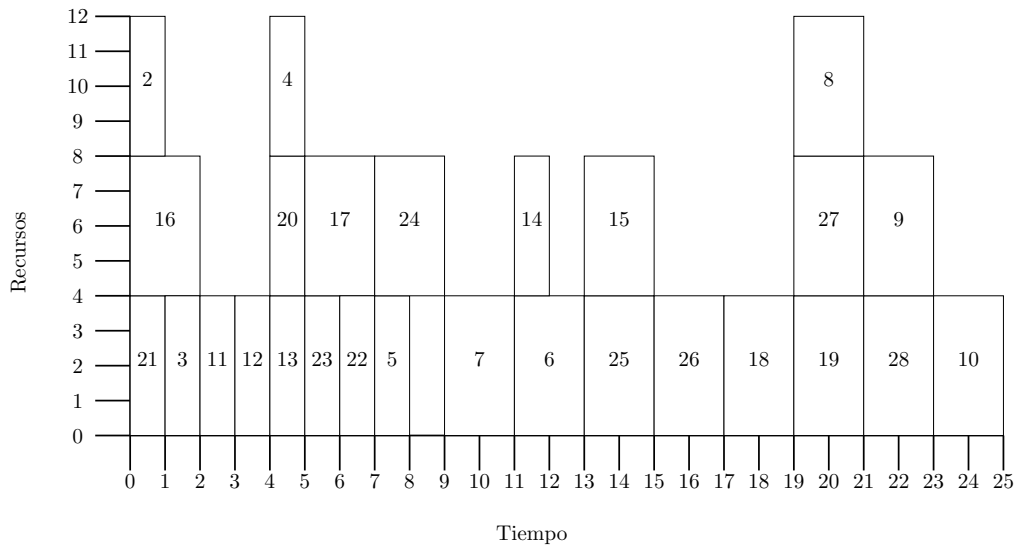
**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 25. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 6 **Temperatura actual:** 32.76





(a) Planeación actual en la iteración 4

L	1	21	16	2	3	11	12	13	20	4	23	17	22	5	24	7	6	14	25	15	26	18	19	27	8	28	9	10	29
L'	0	0	0	0	1	2	3	4	4	4	5	5	6	7	7	9	11	11	13	13	15	17	19	19	19	21	21	23	25

(b) Representación de la solución actual en la iteración 4

Figura 3.32: Planeación y representación de la solución actual en la iteración 4

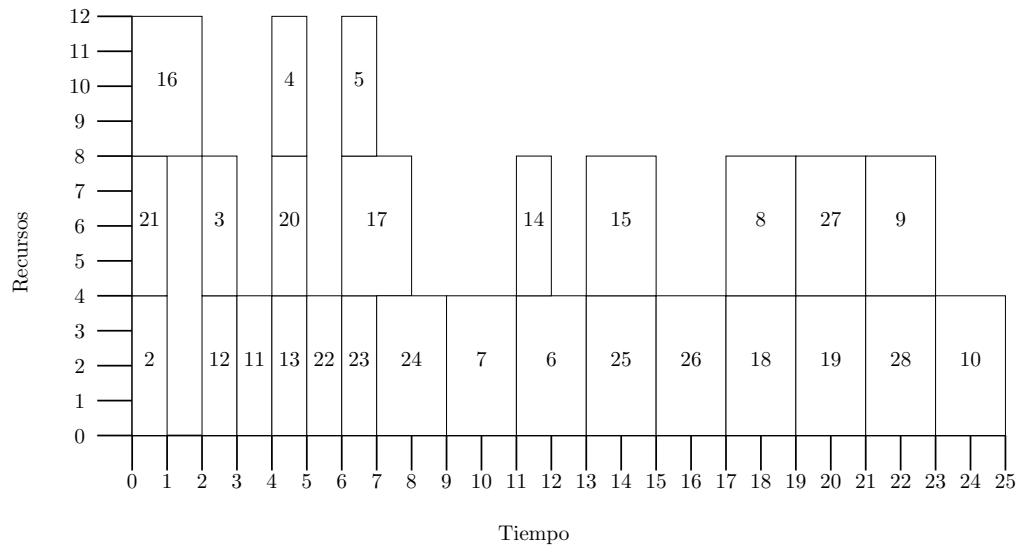
**Solución actual:**

La solución actual al iniciar la iteración 6 es la solución vecina 4 con costo 24 de la iteración anterior y que se muestra en la Figura 3.34.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.



(a) Planeación actual en la iteración 5

L	1	2	21	16	12	3	11	13	20	4	22	23	17	5	24	7	6	14	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	2	2	3	4	4	4	5	6	6	6	7	9	11	11	13	13	15	17	17	19	19	21	21	23	25

(b) Representación de la solución actual en la iteración 5

Figura 3.33: Planeación y representación de la solución actual en la iteración 5

Iteración 7 **Temperatura actual:** 26.21

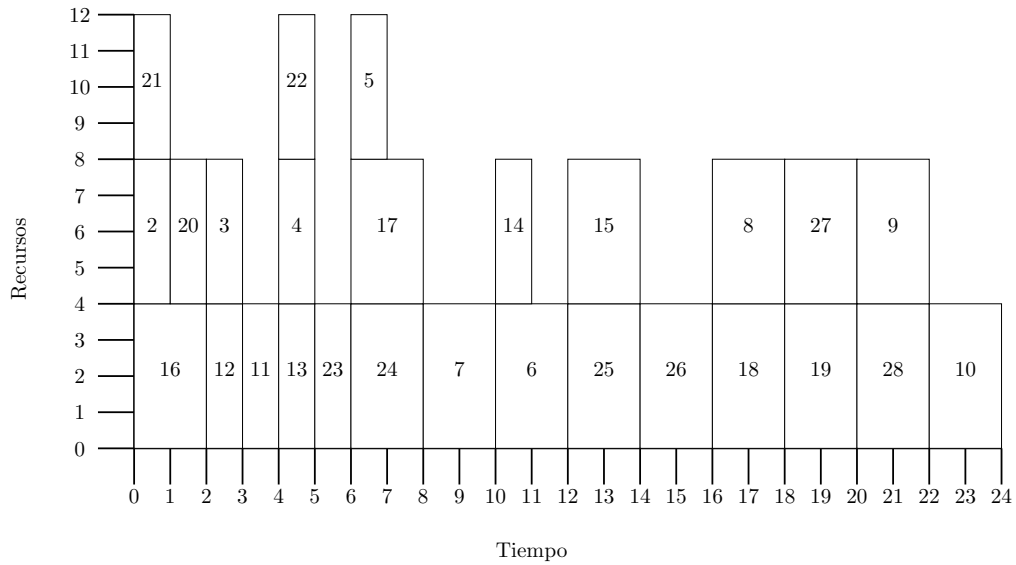
**Solución actual:**

La solución actual al iniciar la iteración 7 es la solución vecina 4 con costo 24 de la iteración anterior y que se muestra en la Figura 3.35.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución



(a) Planeación actual en la iteración 6

L	1	16	2	21	20	12	3	11	13	4	22	23	24	17	5	7	6	14	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	2	2	3	4	4	4	5	6	6	6	8	10	10	12	12	14	16	16	18	18	20	20	22	24

(b) Representación de la solución actual en la iteración 6

Figura 3.34: Planeación y representación de la solución actual en la iteración 6

tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 8 **Temperatura actual:** 20.97

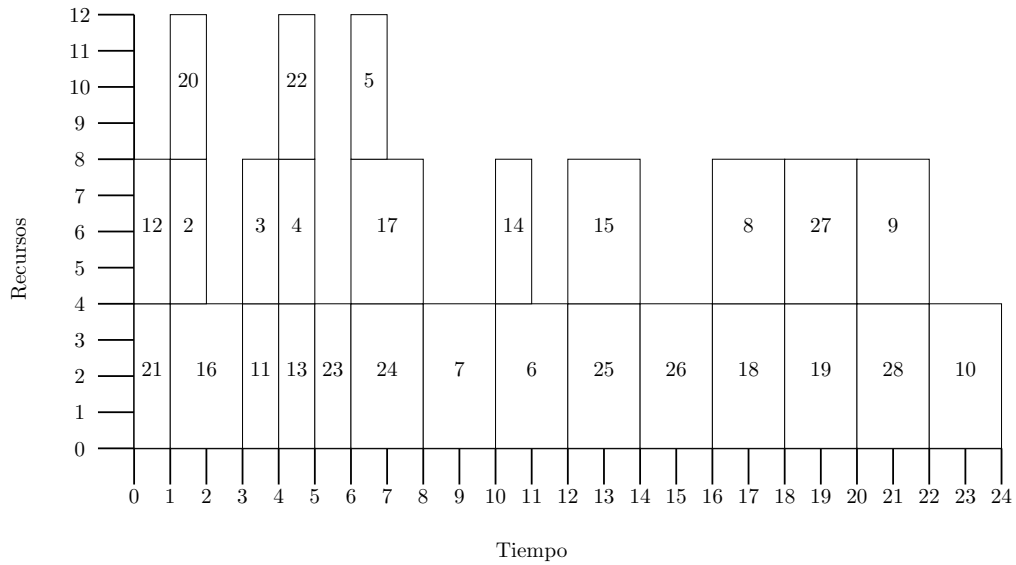
**Solución actual:**

La solución actual al iniciar la iteración 8 es la solución vecina 4 con costo 24 de la iteración anterior y que se muestra en la Figura 3.36.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.



(a) Planeación actual en la iteración 7

L	1	21	12	16	2	20	11	3	13	4	22	23	24	17	5	7	6	14	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	1	1	1	3	3	4	4	4	5	6	6	6	8	10	10	12	12	14	16	16	18	18	20	20	22	24

(b) Representación de la solución actual en la iteración 7

Figura 3.35: Planeación y representación de la solución actual en la iteración 7

Vecino 4. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 9 **Temperatura actual:** 16.77

**Solución actual:**

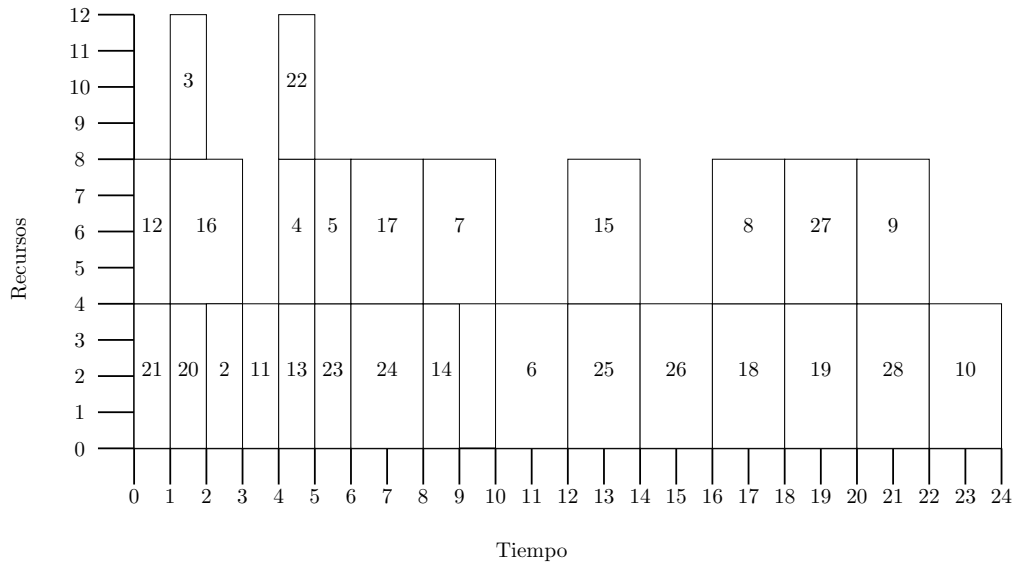
La solución actual al iniciar la iteración 9 es la solución vecina 4 con costo 24 de la iteración anterior y que se muestra en la Figura 3.37.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.



(a) Planeación actual en la iteración 8

L	1	21	12	20	16	3	2	11	13	4	22	23	5	24	17	14	7	6	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	1	1	1	2	3	4	4	4	5	5	6	6	8	8	10	12	12	14	16	16	18	18	20	20	22	24

(b) Representación de la solución actual en la iteración 8

Figura 3.36: Planeación y representación de la solución actual en la iteración 8

- Vecino 3. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.942, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.

Iteración 10 **Temperatura actual:** 13.42

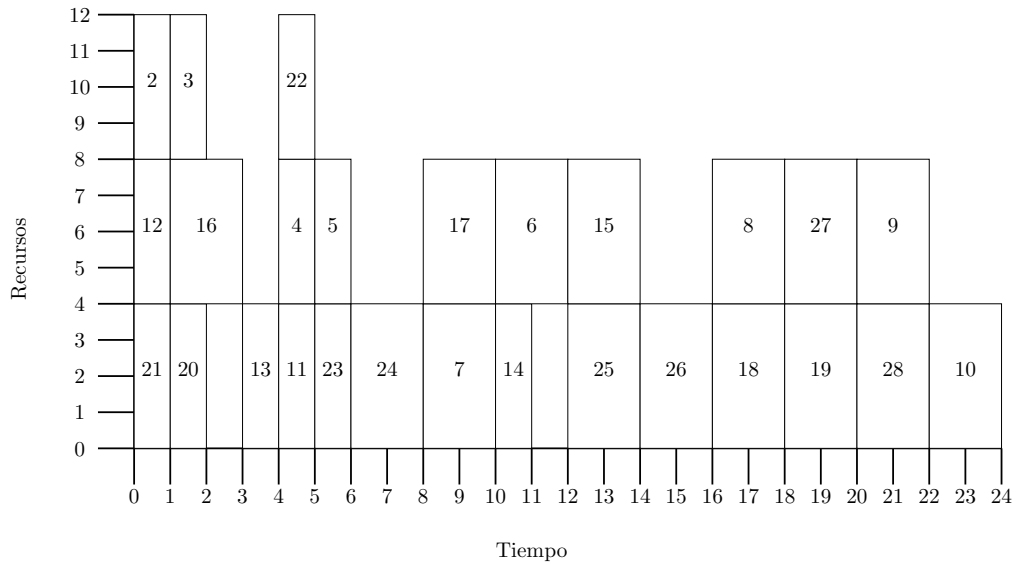
**Solución actual:**

La solución actual al iniciar la iteración 10 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 3.38.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.



(a) Planeación actual en la iteración 9

L	1	21	12	2	20	16	3	13	11	4	22	23	5	24	7	17	14	6	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	1	1	3	4	4	4	5	5	6	8	8	10	10	12	12	14	16	16	18	18	20	20	22	24

(b) Representación de la solución actual en la iteración 9

Figura 3.37: Planeación y representación de la solución actual en la iteración 9

- Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 9 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 11 **Temperatura actual:** 10.73

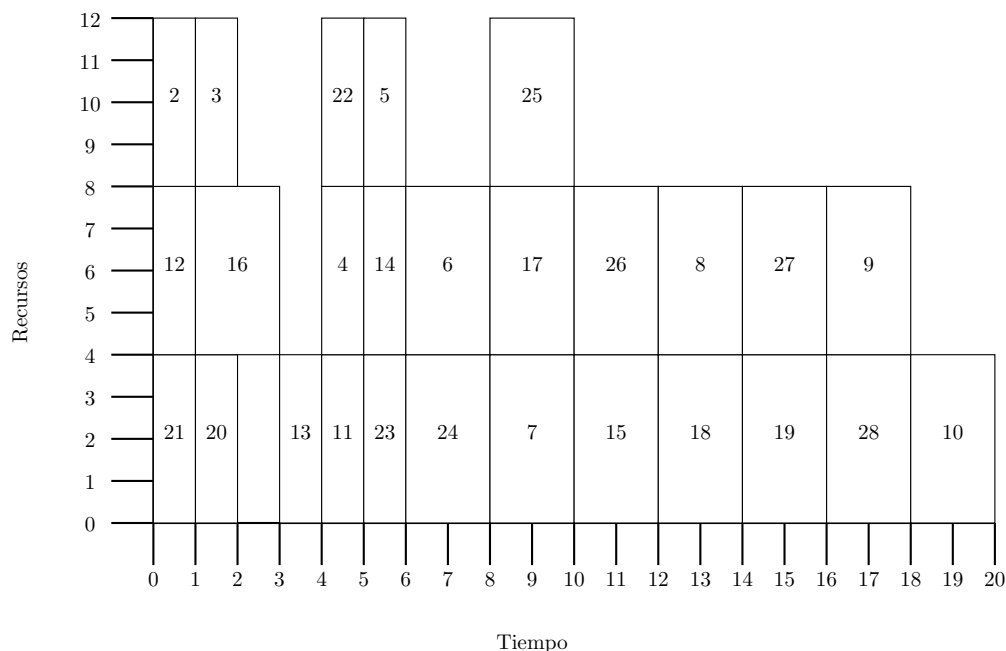
**Solución actual:**

La solución actual al iniciar la iteración 11 es la solución vecina 4 con costo 20 de la iteración anterior y que se muestra en la Figura 3.39.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta



(a) Planeación actual en la iteración 10

L	1	21	12	2	20	16	3	13	11	4	22	23	14	5	24	6	7	17	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	1	1	3	4	4	4	5	5	5	6	6	8	8	8	10	10	12	12	14	14	16	16	18	20

(b) Representación de la solución actual en la iteración 10

Figura 3.38: Planeación y representación de la solución actual en la iteración 10

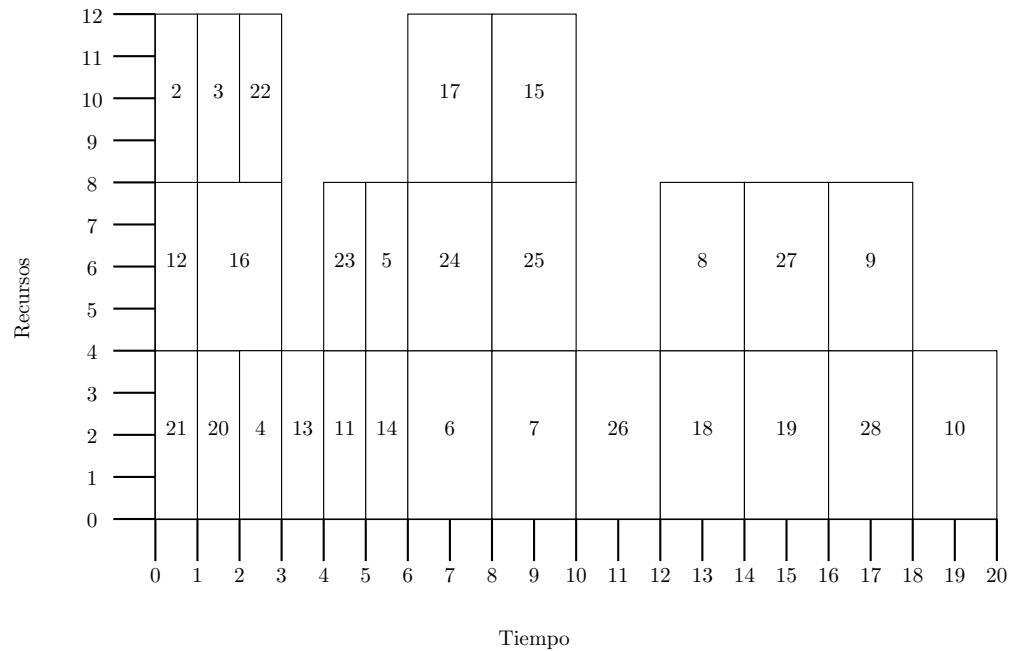
solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

- Vecino 2. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 12 **Temperatura actual:** 8.58

**Solución actual:**

La solución actual al iniciar la iteración 12 es la solución vecina 4 con costo 20 de la iteración anterior y que se muestra en la Figura 3.40.



(a) Planeación actual en la iteración 11

L	1	21	12	2	20	16	3	4	22	13	11	23	14	5	6	24	17	7	25	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	1	1	2	2	3	4	4	5	5	6	6	6	8	8	8	10	12	12	14	14	16	16	18	20

(b) Representación de la solución actual en la iteración 11

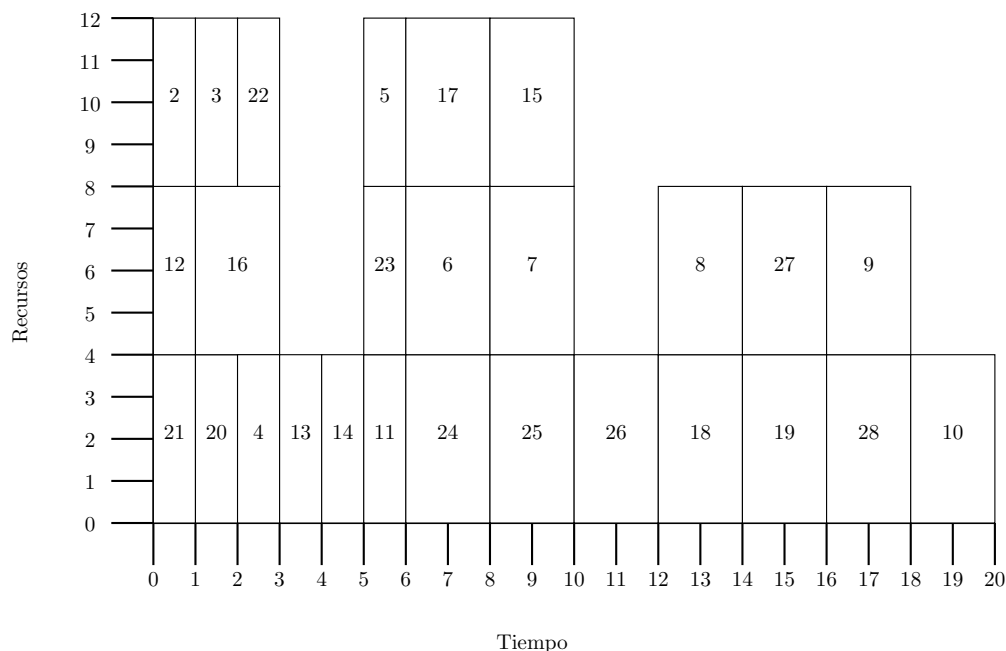
Figura 3.39: Planeación y representación de la solución actual en la iteración 11

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.





(a) Planeación actual en la iteración 12

L	1	21	12	2	20	16	3	4	22	13	14	11	23	5	24	6	17	25	7	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	1	1	2	2	3	4	5	5	5	6	6	6	8	8	8	10	12	12	14	14	16	16	18	20

(b) Representación de la solución actual en la iteración 12

Figura 3.40: Planeación y representación de la solución actual en la iteración 12

Iteración 13 **Temperatura actual:** 6.87

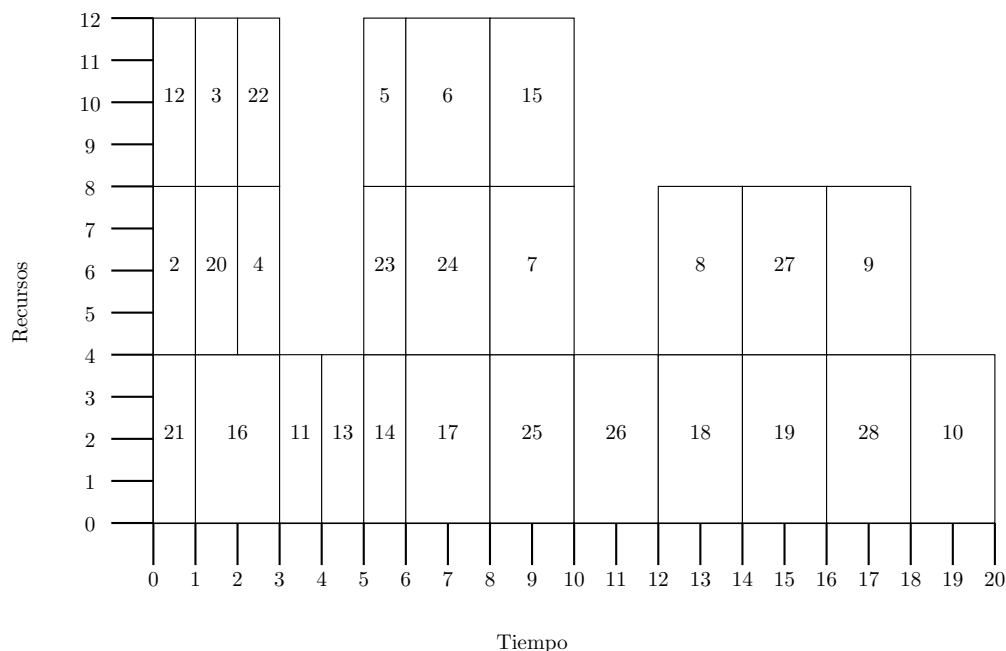
**Solución actual:**

La solución actual al iniciar la iteración 13 es la solución vecina 4 con costo 20 de la iteración anterior y que se muestra en la Figura 3.41.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 9 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta



(a) Planeación actual en la iteración 13

L	1	21	2	12	16	20	3	4	22	11	13	14	23	5	17	24	6	25	7	15	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	1	1	1	2	2	3	4	5	5	5	6	6	6	8	8	8	10	12	12	14	14	16	16	18	20

(b) Representación de la solución actual en la iteración 13

Figura 3.41: Planeación y representación de la solución actual en la iteración 13

solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

- Vecino 4. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 14 **Temperatura actual:** 5.49

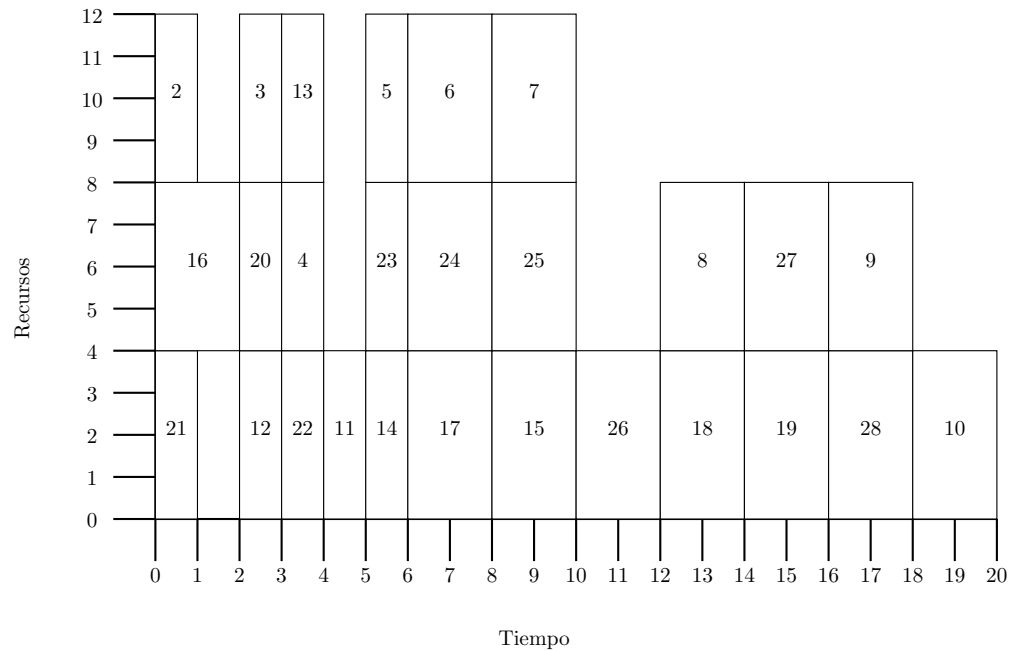
**Solución actual:**

La solución actual al iniciar la iteración 14 es la solución vecina 4 con costo 20 de la iteración anterior y que se muestra en la Figura 3.42.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.



(a) Planeación actual en la iteración 14

L	1	21	16	2	12	20	3	22	4	13	11	14	23	5	17	24	6	15	25	7	26	18	8	19	27	28	9	10	29
L'	0	0	0	0	2	2	2	3	3	3	4	5	5	5	6	6	6	8	8	8	10	12	12	14	14	16	16	18	20

(b) Representación de la solución actual en la iteración 14

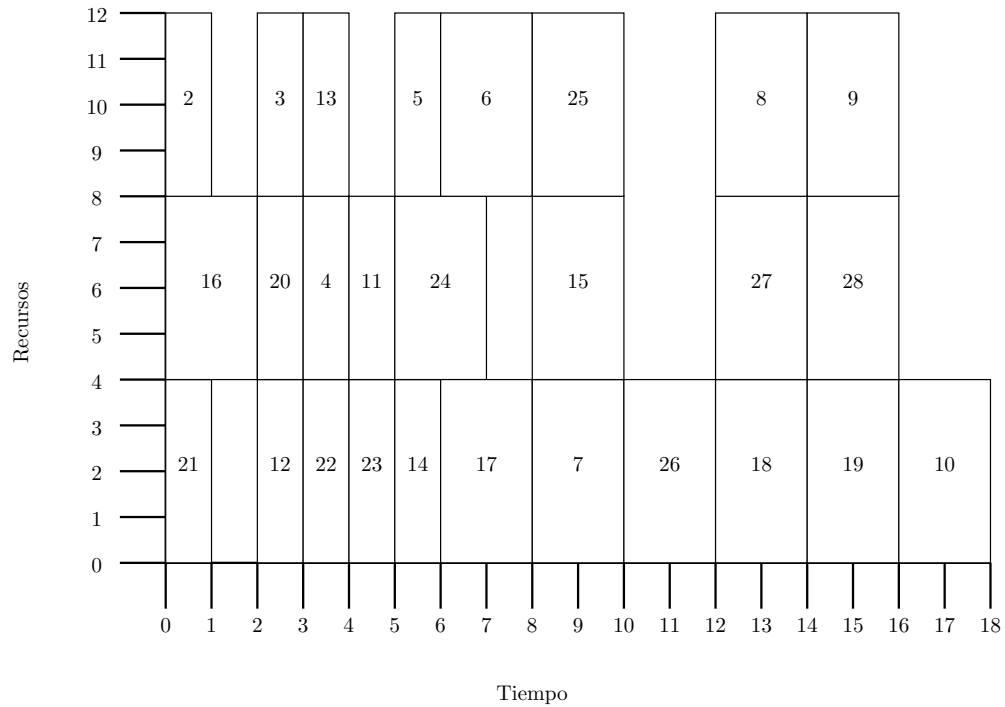
Figura 3.42: Planeación y representación de la solución actual en la iteración 14

- Vecino 2. Rotando las actividades alrededor de la posición 14 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 23 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Esta solución tiene un mejor costo que la solución actual, además de ser la mejor solución hasta el momento, por tanto, se toma como la nueva solución actual y como la mejor solución encontrada hasta el momento.

Iteración 15 **Temperatura actual:** 4.39

**Solución actual:**

La solución actual al iniciar la iteración 15 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.43.



(a) Planeación actual en la iteración 15

L	1	21	16	2	12	20	3	22	4	13	23	11	14	24	5	17	6	7	15	25	26	18	27	8	19	28	9	10	29
L'	0	0	0	0	2	2	2	3	3	3	4	4	5	5	5	6	6	8	8	8	10	12	12	12	14	14	14	16	18

(b) Representación de la solución actual en la iteración 15

Figura 3.43: Planeación y representación de la solución actual en la iteración 15

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

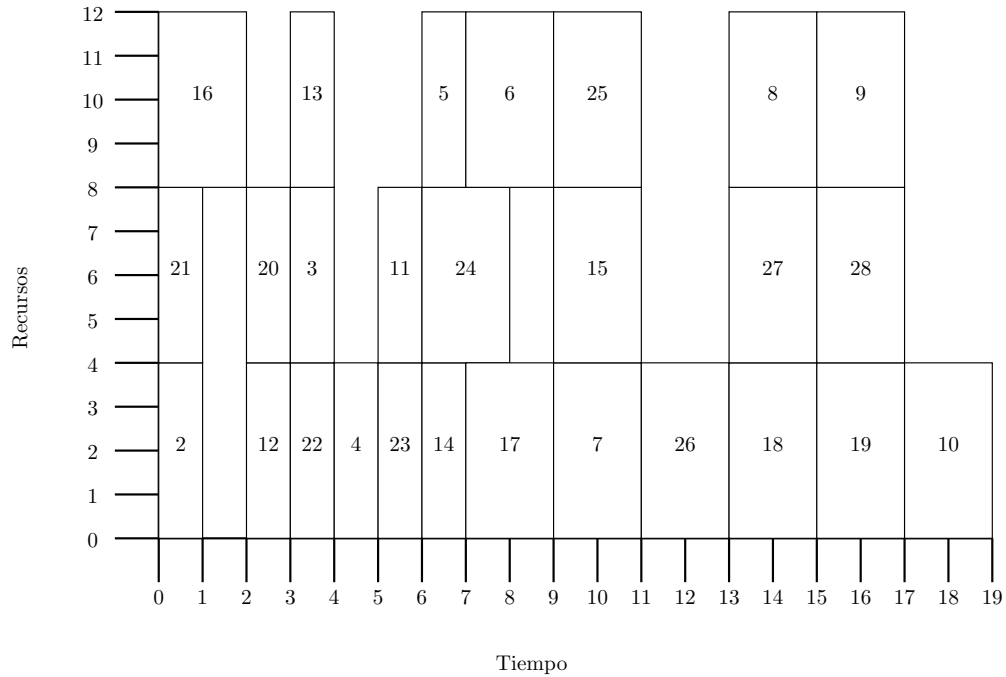
- Vecino 1. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 24 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.634, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.796, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.

Iteración 16 **Temperatura actual: 3.51**

**Solución actual:**

La solución actual al iniciar la iteración 16 es la solución vecina 4 con costo 19 de la iteración anterior y que se muestra en la Figura 3.44.



(a) Planeación actual en la iteración 16

L	1	2	21	16	12	20	22	3	13	4	23	11	14	24	5	17	6	7	15	25	26	18	27	8	19	28	9	10	29
L'	0	0	0	0	2	2	3	3	3	4	5	5	6	6	6	7	7	9	9	9	11	13	13	13	15	15	15	17	19

(b) Representación de la solución actual en la iteración 16

Figura 3.44: Planeación y representación de la solución actual en la iteración 16

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

- Vecino 2. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.426, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 3. Rotando las actividades alrededor de la posición 24 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.566, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 14 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.752, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.

Iteración 17 **Temperatura actual:** 2.81

**Solución actual:**

La solución actual al iniciar la iteración 17 es la solución vecina 3 con costo 21 de la iteración anterior y que se muestra en la Figura 3.45.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 14 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.7, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

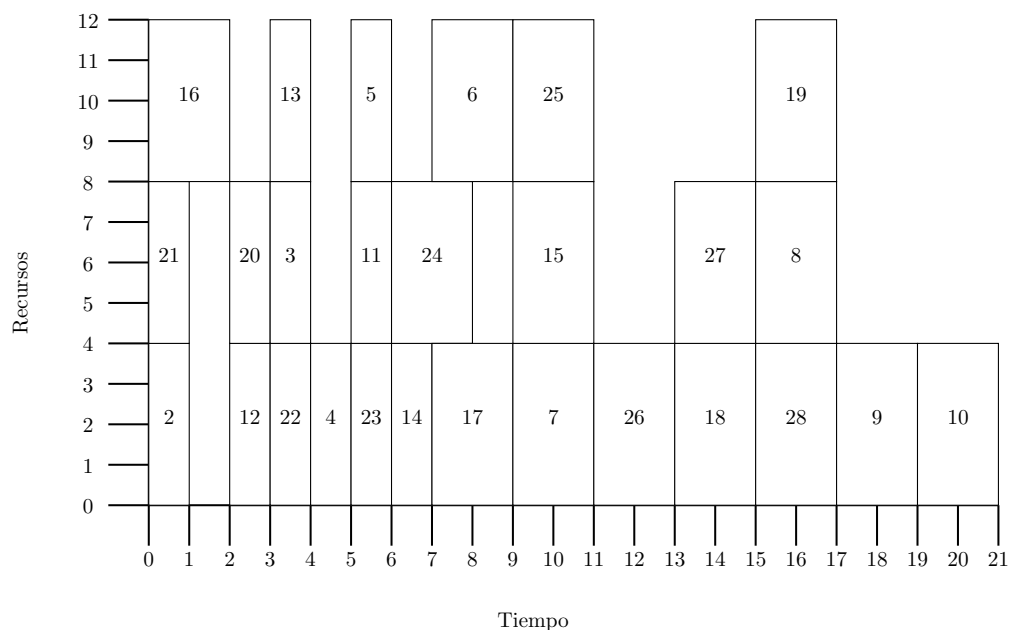
Iteración 18 **Temperatura actual:** 2.25

**Solución actual:**

La solución actual al iniciar la iteración 18 es la solución vecina 4 con costo 22 de la iteración anterior y que se muestra en la Figura 3.46.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.



(a) Planeación actual en la iteración 17

L	1	2	21	16	12	20	22	3	13	4	23	11	5	14	24	17	6	7	15	25	26	18	27	28	8	19	9	10	29
L'	0	0	0	0	2	2	3	3	3	4	5	5	5	6	6	7	7	9	9	9	11	13	13	15	15	15	17	19	21

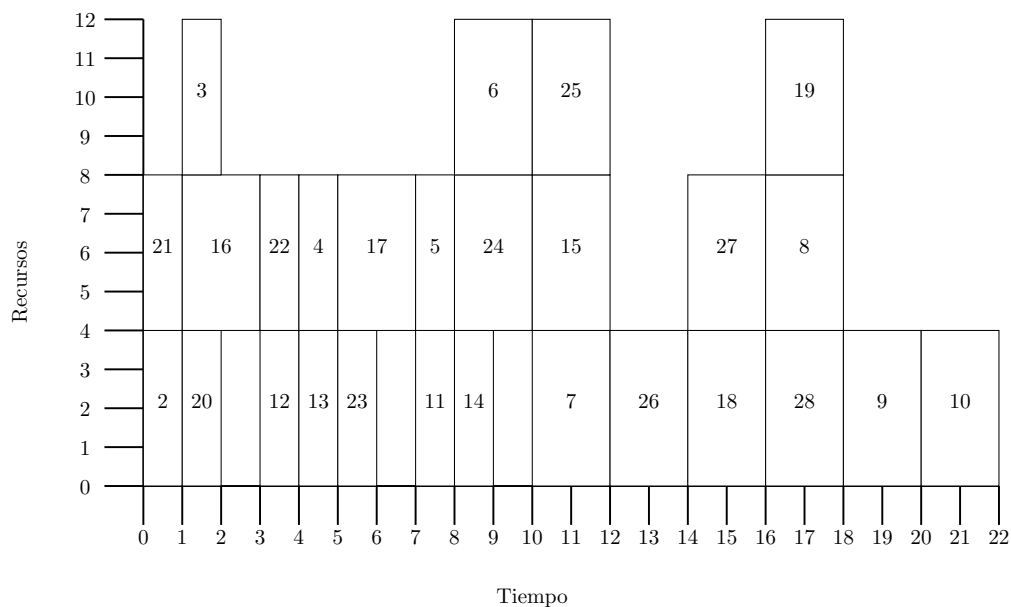
(b) Representación de la solución actual en la iteración 17

Figura 3.45: Planeación y representación de la solución actual en la iteración 17

- Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 24 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 19 **Temperatura actual:** 1.80

**Solución actual:**



(a) Planeación actual en la iteración 18

L	1	2	21	20	16	3	12	22	13	4	23	17	11	5	14	24	6	7	15	25	26	18	27	28	8	19	9	10	29	
L'	0	0	0	1	1	1	3	3	4	4	5	5	7	7	8	8	8	8	10	10	10	12	14	14	16	16	16	18	20	22

(b) Representación de la solución actual en la iteración 18

Figura 3.46: Planeación y representación de la solución actual en la iteración 18

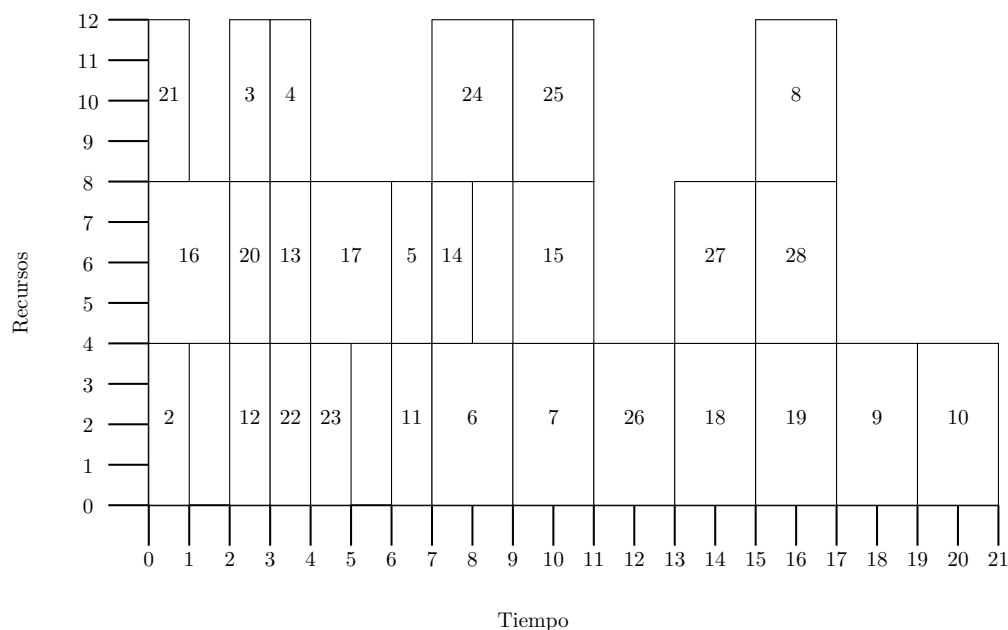
La solución actual al iniciar la iteración 19 es la solución vecina 4 con costo 21 de la iteración anterior y que se muestra en la Figura 3.47.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.574, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.





(a) Planeación actual en la iteración 19

L	1	2	16	21	12	20	3	22	13	4	23	17	11	5	6	14	24	7	15	25	26	18	27	19	28	8	9	10	29
L'	0	0	0	0	2	2	2	3	3	3	4	4	6	6	7	7	7	9	9	9	11	13	13	15	15	15	17	19	21

(b) Representación de la solución actual en la iteración 19

Figura 3.47: Planeación y representación de la solución actual en la iteración 19

Vecino 4. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 23. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.5, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.

Iteración 20 **Temperatura actual:** 1.44

**Solución actual:**

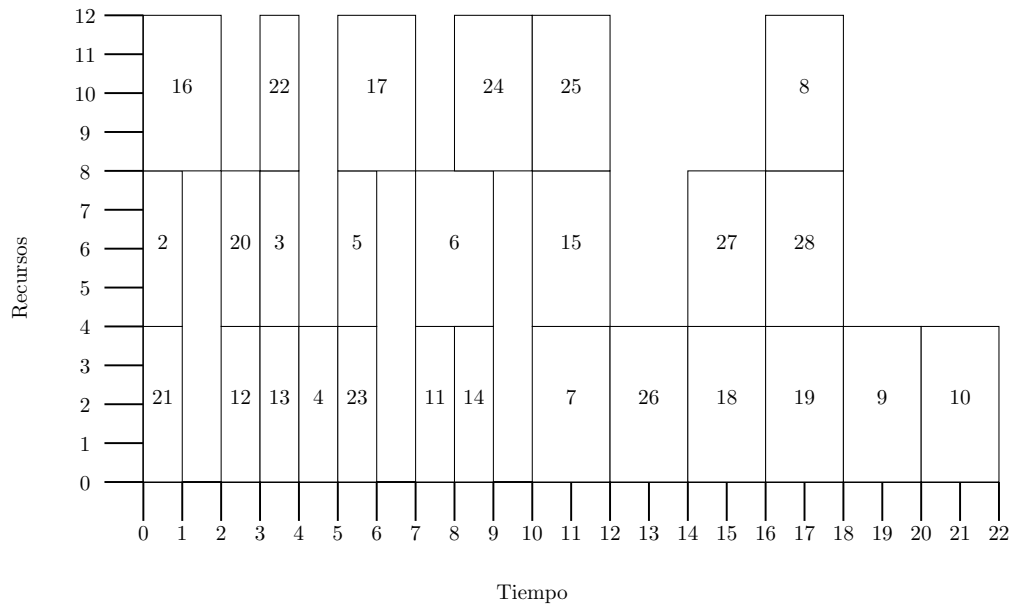
La solución actual al iniciar la iteración 20 es la solución vecina 3 con costo 22 de la iteración anterior y que se muestra en la Figura 3.48.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene



(a) Planeación actual en la iteración 20

L	1	21	2	16	12	20	13	3	22	4	23	5	17	11	6	14	24	7	15	25	26	18	27	19	28	8	9	10	29
L'	0	0	0	0	2	2	3	3	3	4	5	5	5	7	7	8	8	10	10	10	12	14	14	16	16	16	18	20	22

(b) Representación de la solución actual en la iteración 20

Figura 3.48: Planeación y representación de la solución actual en la iteración 20

un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.499, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.

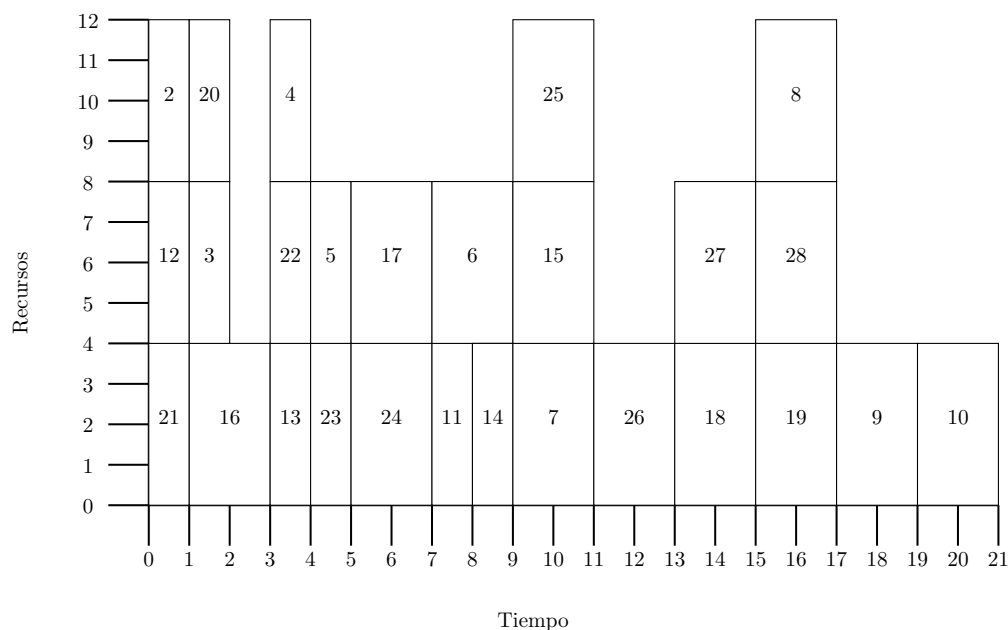
Iteración 21 **Temperatura actual:** 1.15

**Solución actual:**

La solución actual al iniciar la iteración 21 es la solución vecina 4 con costo 21 de la iteración anterior y que se muestra en la Figura 3.49.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.



(a) Planeación actual en la iteración 21

L	1	21	12	2	16	3	20	13	22	4	23	5	24	17	11	6	14	7	15	25	26	18	27	19	28	8	9	10	29
L'	0	0	0	0	1	1	1	3	3	3	4	4	5	5	7	7	8	9	9	9	11	13	13	15	15	15	17	19	21

(b) Representación de la solución actual en la iteración 21

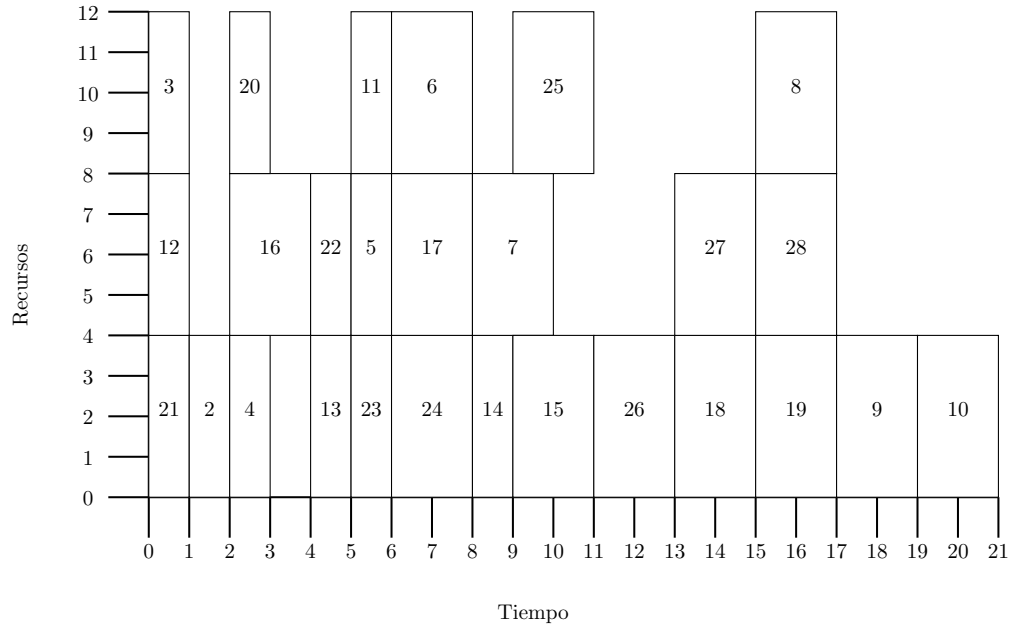
Figura 3.49: Planeación y representación de la solución actual en la iteración 21

- Vecino 1. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.42, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución es aceptada, por tanto, se toma como la nueva solución actual.

Iteración 22 **Temperatura actual:** 0.922

**Solución actual:**

La solución actual al iniciar la iteración 22 es la solución vecina 4 con costo 21 de la iteración anterior y que se muestra en la Figura 3.50.



(a) Planeación actual en la iteración 22

L	1	21	12	3	2	4	16	20	13	22	23	5	11	24	17	6	14	7	15	25	26	18	27	19	28	8	9	10	29
L'	0	0	0	0	1	2	2	2	4	4	5	5	5	6	6	6	8	8	9	9	11	13	13	15	15	15	17	19	21

(b) Representación de la solución actual en la iteración 22

Figura 3.50: Planeación y representación de la solución actual en la iteración 22

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

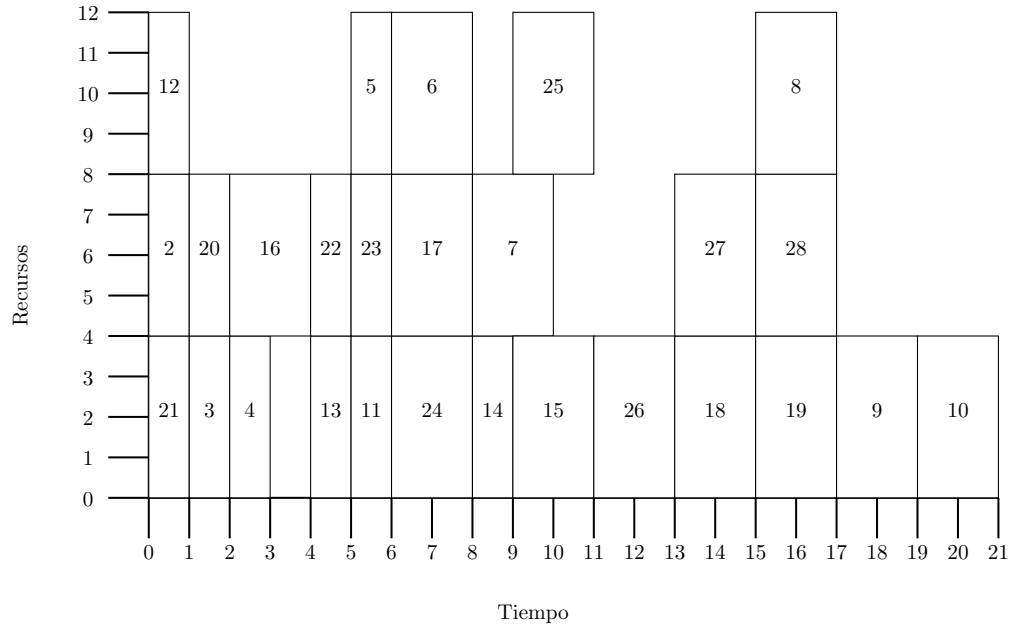
- Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 24. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.038, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 3. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 23 **Temperatura actual:** 0.737

**Solución actual:**

La solución actual al iniciar la iteración 23 es la solución vecina 4 con costo 21 de la iteración anterior y que se muestra en la Figura 3.51.



(a) Planeación actual en la iteración 23

L	1	21	2	12	3	20	4	16	13	22	11	23	5	24	17	6	14	7	15	25	26	18	27	19	28	8	9	10	29
L'	0	0	0	0	1	1	2	2	4	4	5	5	5	6	6	6	8	8	9	9	11	13	13	15	15	15	17	19	21

(b) Representación de la solución actual en la iteración 23

Figura 3.51: Planeación y representación de la solución actual en la iteración 23

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 24 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

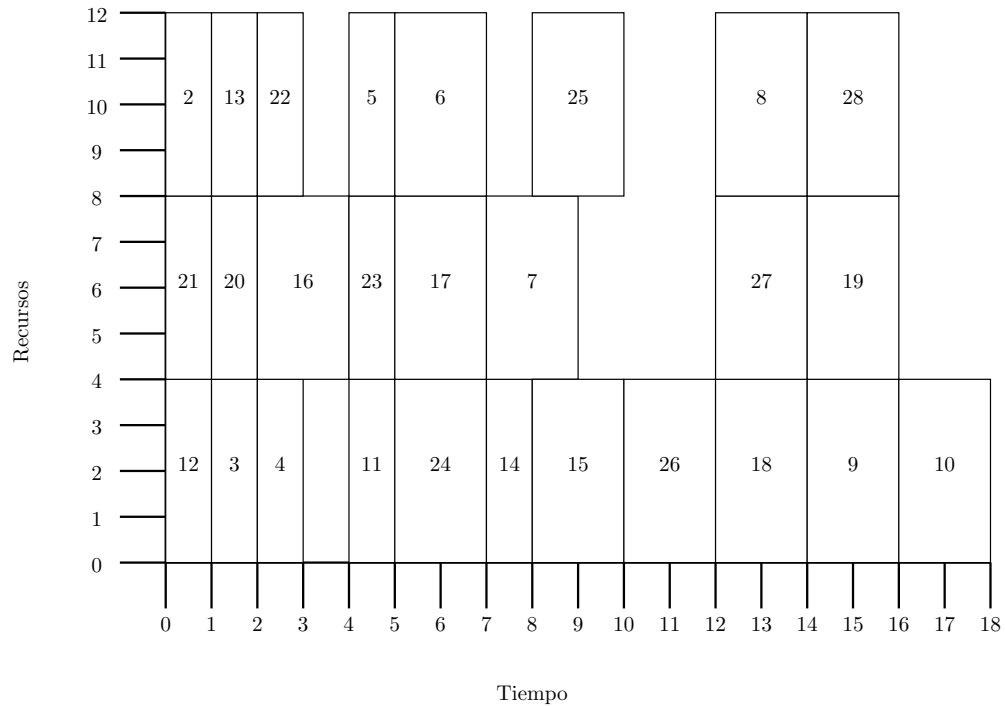
Vecino 3. Rotando las actividades alrededor de la posición 25 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Esta solución tiene un mejor costo que la solución actual, por tanto, se toma como la nueva solución actual.

Iteración 24 **Temperatura actual:** 0.59

**Solución actual:**

La solución actual al iniciar la iteración 24 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.52.



(a) Planeación actual en la iteración 24

L	1	12	21	2	3	20	13	4	16	22	11	23	5	24	17	6	14	7	15	25	26	18	27	8	9	19	28	10	29
L'	0	0	0	0	1	1	1	2	2	2	4	4	4	5	5	5	7	7	8	8	10	12	12	12	14	14	14	16	18

(b) Representación de la solución actual en la iteración 24

Figura 3.52: Planeación y representación de la solución actual en la iteración 24

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.183, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 3. Rotando las actividades alrededor de la posición 26 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 25 **Temperatura actual:** 0.472

**Solución actual:**

La solución actual al iniciar la iteración 25 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.53.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.014, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 3. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

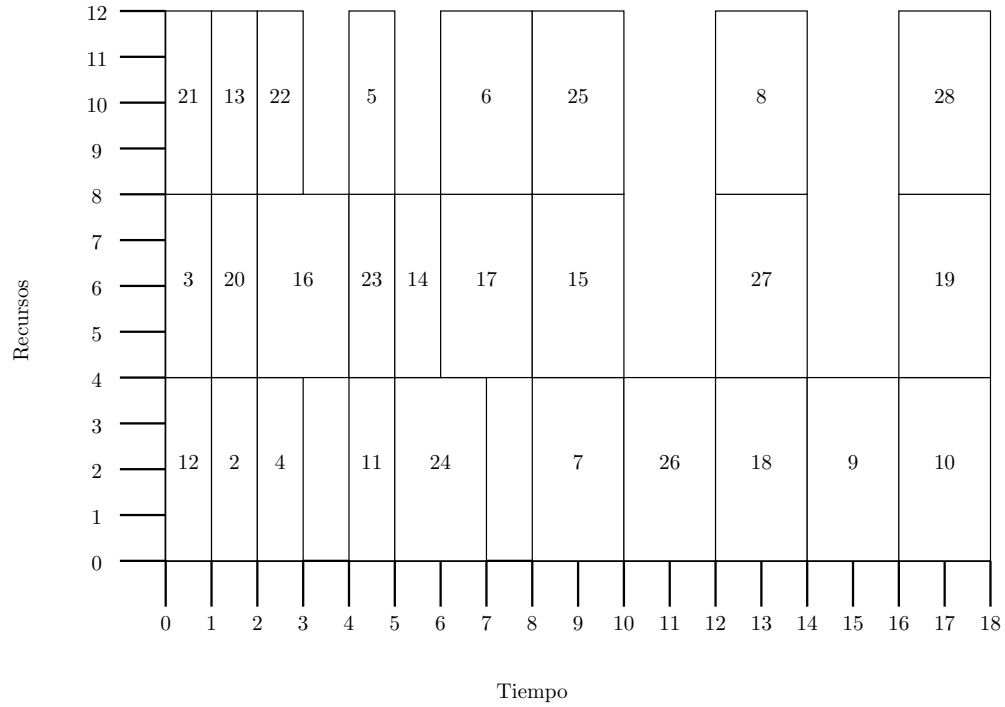
Iteración 26 **Temperatura actual:** 0.377

**Solución actual:**

La solución actual al iniciar la iteración 26 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.54.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.



(a) Planeación actual en la iteración 25

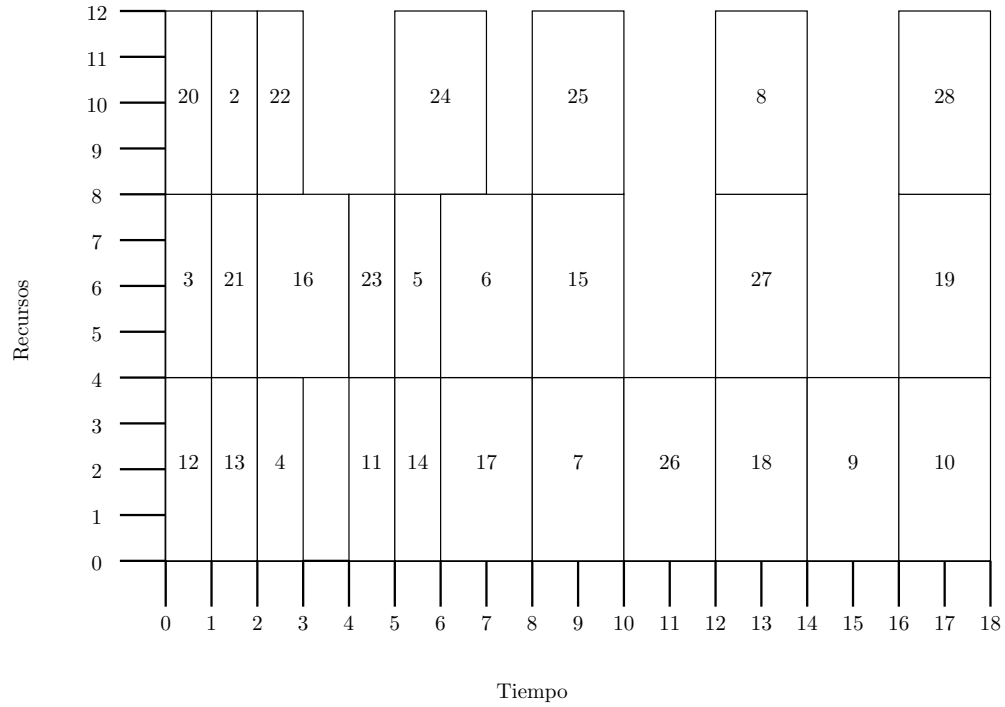
L	1	12	3	21	2	20	13	4	16	22	11	23	5	24	14	17	6	7	15	25	26	18	27	8	9	10	19	28	29	
L'	0	0	0	0	1	1	1	2	2	2	4	4	4	5	5	6	6	8	8	8	8	10	12	12	12	14	16	16	16	18

(b) Representación de la solución actual en la iteración 25

Figura 3.53: Planeación y representación de la solución actual en la iteración 25

- Vecino 1. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.0003, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.





(a) Planeación actual en la iteración 26

L	1	12	3	20	13	21	2	4	16	22	11	23	14	5	24	17	6	7	15	25	26	18	27	8	9	10	19	28	29
L'	0	0	0	0	1	1	1	2	2	2	4	4	5	5	5	6	6	8	8	8	10	12	12	12	14	16	16	16	18

(b) Representación de la solución actual en la iteración 26

Figura 3.54: Planeación y representación de la solución actual en la iteración 26

Iteración 27 **Temperatura actual:** 0.302

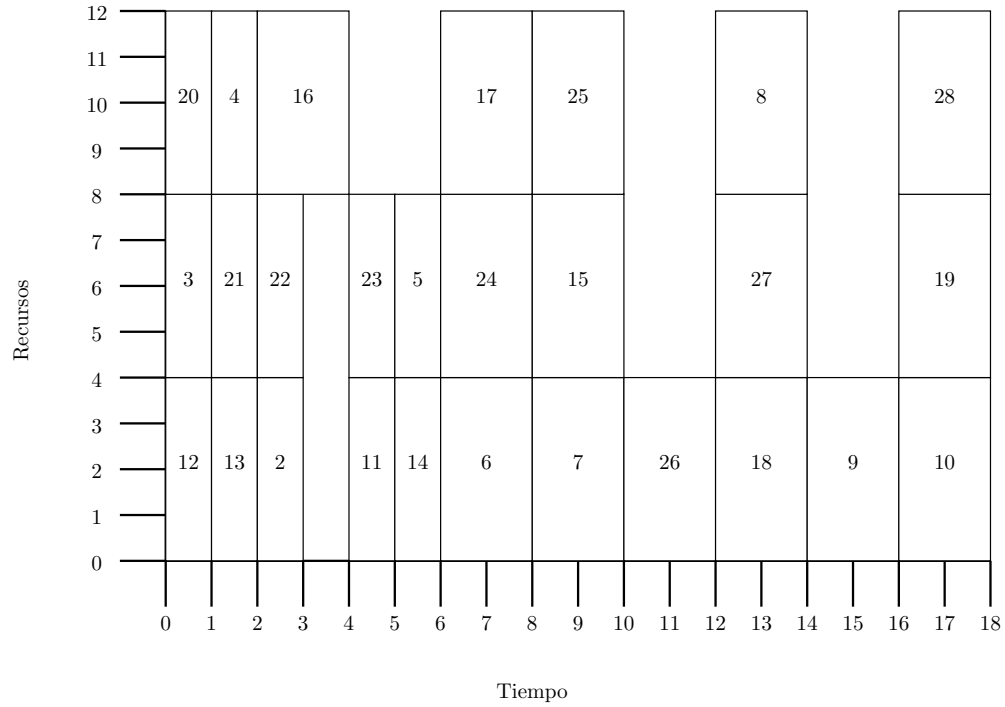
**Solución actual:**

La solución actual al iniciar la iteración 27 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.55.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 26 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución



(a) Planeación actual en la iteración 27

L	1	12	3	20	13	21	4	2	22	16	11	23	14	5	6	24	17	7	15	25	26	18	27	8	9	10	19	28	29	
L'	0	0	0	0	1	1	1	2	2	2	4	4	5	5	6	6	6	8	8	8	8	10	12	12	12	14	16	16	16	18

(b) Representación de la solución actual en la iteración 27

Figura 3.55: Planeación y representación de la solución actual en la iteración 27

tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 28 **Temperatura actual:** 0.241

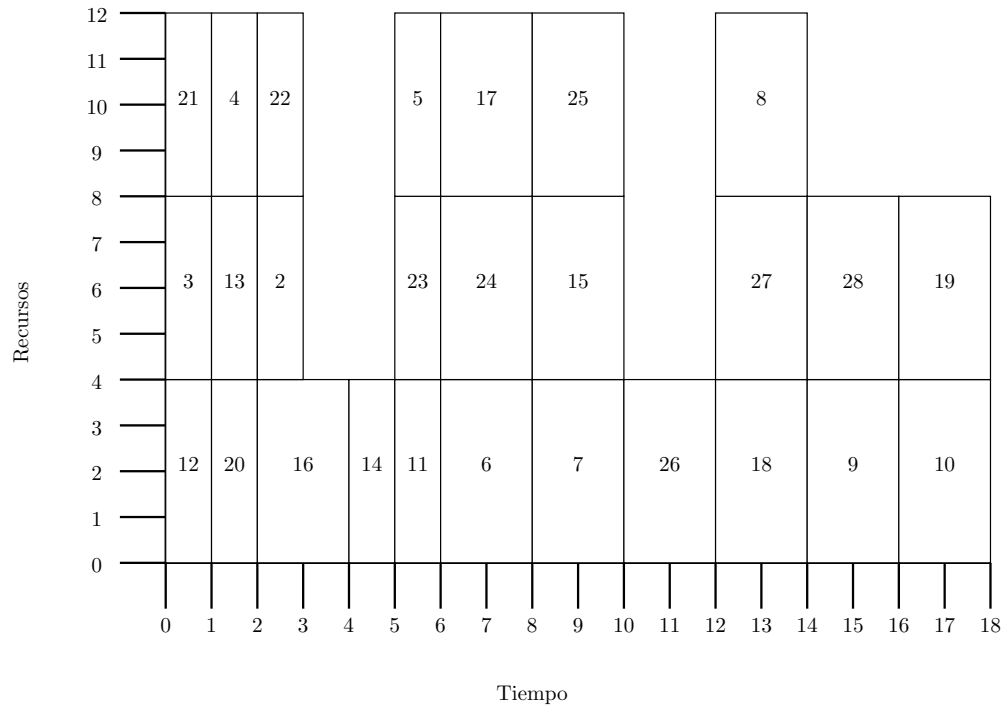
**Solución actual:**

La solución actual al iniciar la iteración 28 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.56.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución



(a) Planeación actual en la iteración 28

L	1	12	3	21	20	13	4	16	2	22	14	11	23	5	6	24	17	7	15	25	26	18	27	8	9	28	10	19	29
L'	0	0	0	0	1	1	1	2	2	2	4	5	5	5	6	6	6	8	8	8	10	12	12	12	14	14	16	16	18

(b) Representación de la solución actual en la iteración 28

Figura 3.56: Planeación y representación de la solución actual en la iteración 28

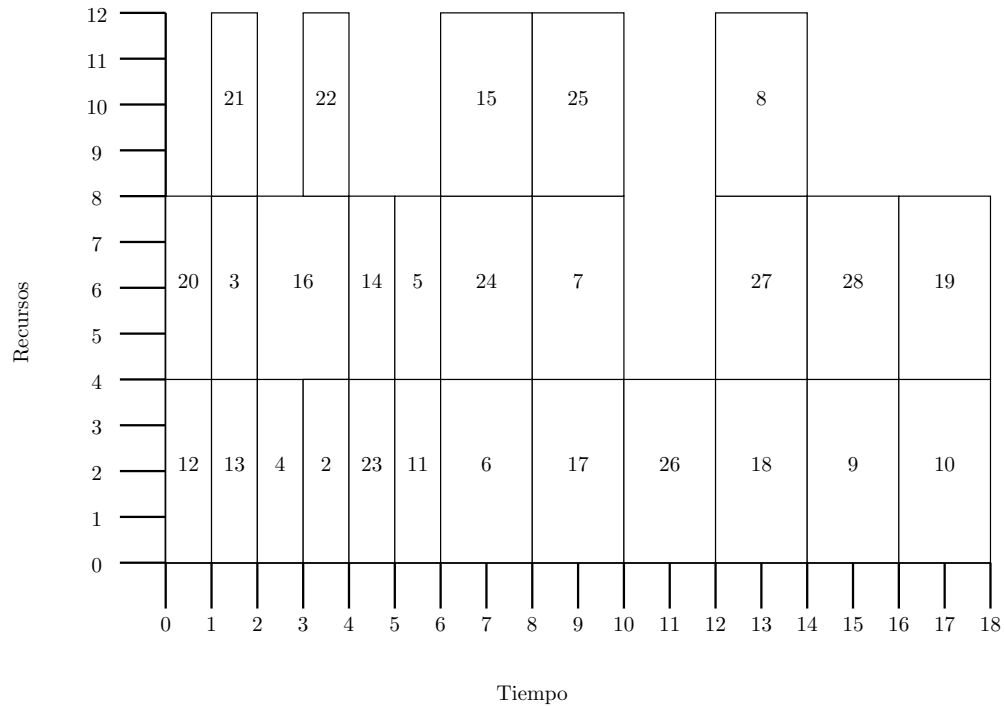
tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

- Vecino 2. Rotando las actividades alrededor de la posición 4 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 4. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 29 **Temperatura actual:** 0.193

**Solución actual:**

La solución actual al iniciar la iteración 29 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.57.



(a) Planeación actual en la iteración 29

L	1	12	20	13	3	21	4	16	2	22	23	14	11	5	6	24	15	17	7	25	26	18	27	8	9	28	10	19	29	
L'	0	0	0	1	1	1	2	2	3	3	4	4	5	5	6	6	6	8	8	8	8	10	12	12	12	14	14	16	16	18

(b) Representación de la solución actual en la iteración 29

Figura 3.57: Planeación y representación de la solución actual en la iteración 29

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

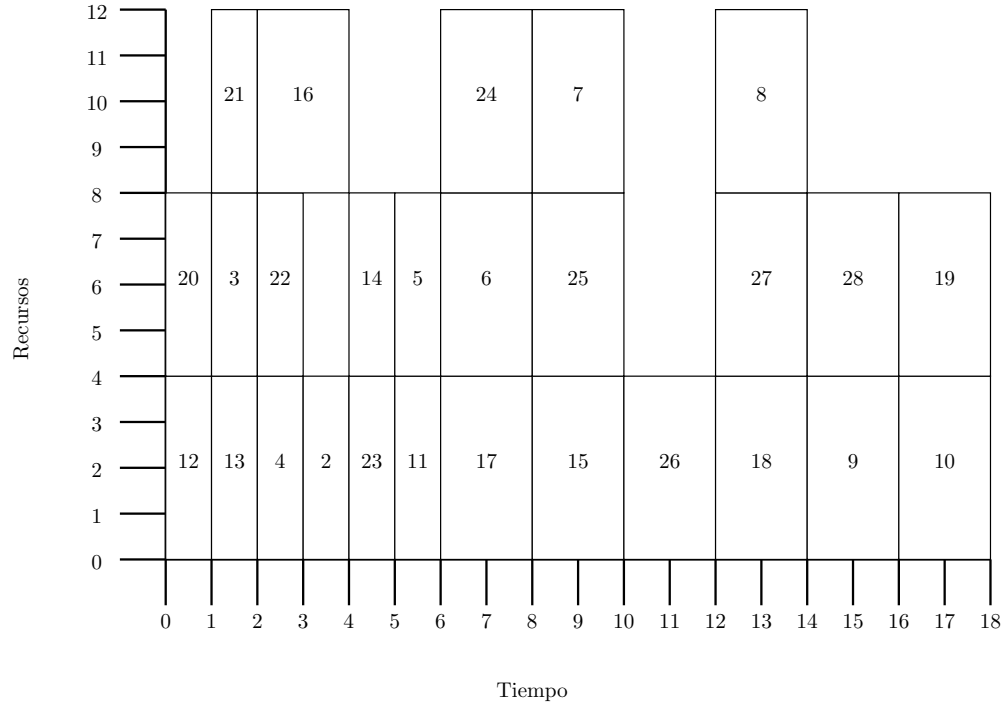
- Vecino 1. Rotando las actividades alrededor de la posición 8 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Vecino 4. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 30 **Temperatura actual:** 0.154

**Solución actual:**

La solución actual al iniciar la iteración 30 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.58.



(a) Planeación actual en la iteración 30

L	1	12	20	13	3	21	4	22	16	2	23	14	11	5	17	6	24	15	25	7	26	18	27	8	9	28	10	19	29	
L'	0	0	0	1	1	1	2	2	2	3	4	4	5	5	6	6	6	8	8	8	8	10	12	12	12	14	14	16	16	18

(b) Representación de la solución actual en la iteración 30

Figura 3.58: Planeación y representación de la solución actual en la iteración 30

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.000002, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.

- Vecino 2. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 16 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.000002, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 4. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

Iteración 31 **Temperatura actual:** 0.12

**Solución actual:**

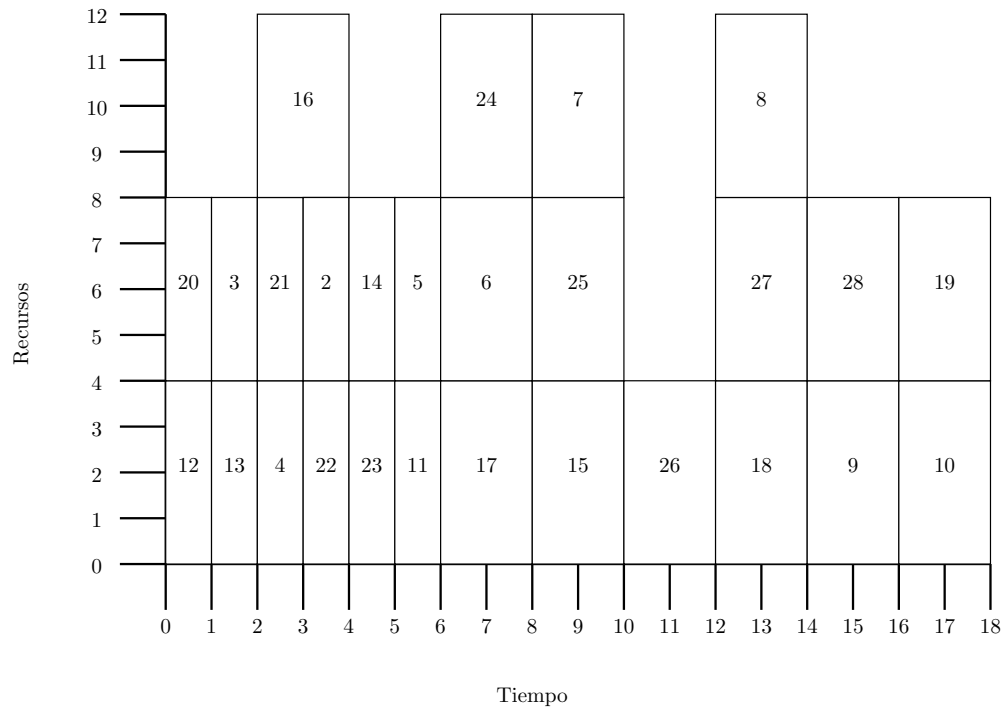
La solución actual al iniciar la iteración 31 es la solución vecina 4 con costo 18 de la iteración anterior y que se muestra en la Figura 3.59.

**Soluciones vecinas a revisar y su costo:**

Se muestran las cuatro soluciones vecinas a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.00000009, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 2. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 13 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución no tiene un mejor costo que la solución actual, pero la probabilidad de ser aceptada es de 0.00000009, así que se decide de manera aleatoria si se puede aceptar bajo esta probabilidad, teniendo como resultado que la solución no es aceptada.
- Vecino 4. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Como esta solución tiene el mismo costo que la solución actual, entonces se toma como la nueva solución actual.

**Resultados** Después de 31 iteraciones, la mejor solución se encontró en la iteración 14 con un costo de 18 unidades, la cual se muestra en la Figura 3.43, aunque en las últimas iteraciones se encontraron soluciones con el mismo costo, esta solución se toma como la mejor porque fue la primera con este costo en ser encontrada. En la Figura 3.60 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística.



(a) Planeación actual en la iteración 31

L	1	12	20	13	3	4	21	16	22	2	23	14	11	5	17	6	24	15	25	7	26	18	27	8	9	28	10	19	29
L'	0	0	0	1	1	2	2	2	3	3	4	4	5	5	6	6	6	8	8	8	10	12	12	12	14	14	16	16	18

(b) Representación de la solución actual en la iteración 31

Figura 3.59: Planeación y representación de la solución actual en la iteración 31

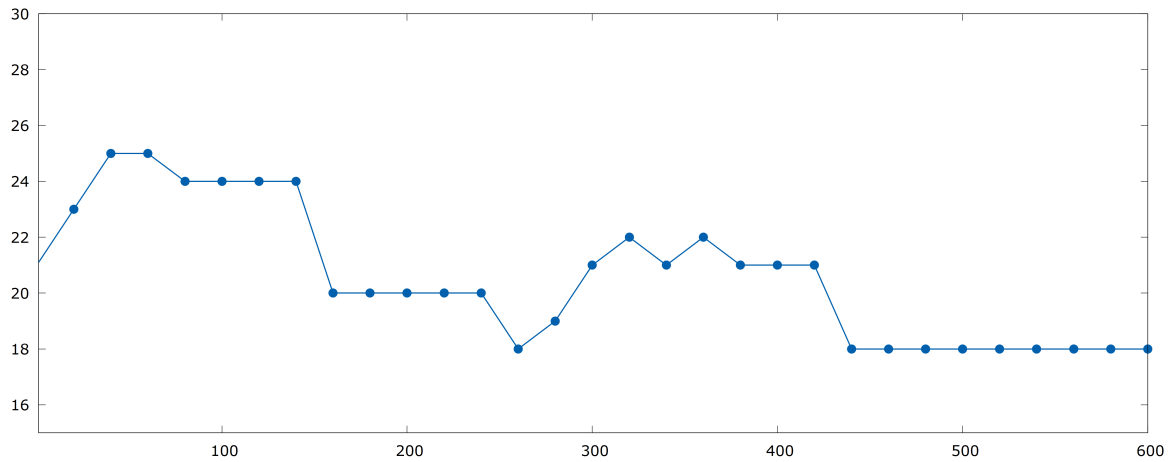


Figura 3.60: Soluciones visitadas durante la ejecución del Recocido Simulado para el proyecto de 15 actividades

## Capítulo 4

# Búsqueda Tabú

La metaheurística de Búsqueda Tabú creada por Fred Glover en 1986 se basa en la búsqueda inteligente para la resolución de problemas, de manera que busca recordar movimientos y con esto, obtener información que pueda utilizar para salir de óptimos locales o modificar la búsqueda. A diferencia de varias metaheurísticas, ésta no utiliza la probabilidad para la elección de soluciones, por lo que, suele decirse que esta metaheurística es determinista, por la razón de tomar decisiones basándose en la memoria.

La Búsqueda Tabú al igual que el Recocido Simulado, usa como heurística subordinada a la búsqueda local, pero a diferencia del Recocido Simulado, esta metaheurística usa memoria para guardar información de los movimientos más recientes, con el objetivo de evitar realizar estos movimientos durante un periodo determinado y de esta forma evitar repetir movimientos que causen regresar a soluciones ya exploradas, aunque como se especifica adelante, a veces es posible realizar estos movimientos prohibidos si se cumplen algunas condiciones.

### 4.1. Descripción

Sea  $S$  el conjunto de todas las soluciones posibles para un problema de optimización  $\mathcal{P}$ .

**Definición 4.1.1** Un **movimiento** es la modificación que se realiza sobre una solución para obtener una solución vecina.

**Definición 4.1.2** Un movimiento se considera **tabú** si éste no se puede realizar durante un determinado tiempo.

**Definición 4.1.3** El **criterio de aspiración** son las condiciones que debe cumplir un movimiento tabú para eliminar la restricción de tabú y, por tanto, ser elegible.

**Definición 4.1.4** Se define el **conjunto de vecinos admisibles**  $N_A(i)$  de una solución  $i$  como aquellas soluciones vecinas que no son generadas por un movimiento tabú o que cumplen con el criterio de aspiración.

Existen dos tipos de memoria que se pueden utilizar:

- La **memoria basada en lo reciente**, es donde se almacenan los últimos movimientos realizados y que se puede utilizar para “*recordar*” los movimientos que hacen caer en soluciones ya exploradas.



- La **memoria basada en frecuencias**, que complementa a la memoria basada en lo reciente, guardando la frecuencia en que se utiliza un movimiento, esto ayuda a identificar cuando un movimiento se realiza tantas veces, lo cual puede afectar la diversidad de soluciones exploradas.

La metaheurística empieza con una solución inicial  $i$ , la mejor solución hasta el momento  $i^*$  y una lista  $T$  donde se almacenan los movimientos tabú. En cada iteración, se revisa un cierto número de vecinos del conjunto de vecinos admisibles  $N_A(i)$  de la solución actual  $i$ , de entre estos, si alguno tiene mejor función de costo que la mejor solución encontrada hasta el momento  $i^*$ , entonces se toma como la nueva mejor solución y se actualiza la solución actual, además, el movimiento que generó a esta solución se vuelve tabú y se agrega a la lista  $T$ , donde permanece con este estado durante un tiempo  $t > 0$  y también se reduce el tiempo tabú para todos los movimientos en  $T$ , cuando un movimiento llega a un tiempo cero, entonces se descarta de la lista  $T$  y puede ser de nuevo elegible.

Todo este proceso se realiza mientras no se cumpla un criterio de término, como pueden ser;

- Hasta alcanzar un número de iteraciones.
- Después de una cantidad de iteraciones sin mejorar la función de costo.
- Cuando la función objetivo alcanzó un valor específico.

En el Algoritmo 14 se presenta un pseudo-código para la metaheurística descrita.

---

#### Algoritmo 14 BÚSQUEDA TABÚ

---

**Entrada:** Solución inicial  $i$ .

Tiempo tabú  $t$ ,  $t > 0$ .

**Salida:** Mejor solución encontrada.

```

1:  $T \leftarrow \emptyset$ 
2: // La mejor solución
3:  $i^* \leftarrow i$ 
4: while No se cumpla el criterio de término do
5:   Sea  $j$  la mejor solución en  $N_A(i)$ 
6:   if  $f(j) < f(i^*)$  then
7:      $i^* \leftarrow j$ 
8:   end if
9:    $i \leftarrow j$ 
10:  Actualiza los tiempos de cada movimiento tabú en  $T$ 
11:  Guarda al movimiento usado para  $j$  en  $T$  con el tiempo  $t$ 
12: end while
13: return  $i^*$ 

```

---

## 4.2. Intensificación y diversificación

A continuación se describen las estrategias de intensificación y diversificación.

### Intensificación

Durante la búsqueda en el espacio de soluciones realizada por la metaheurística y como se sigue la estrategia de la búsqueda local, se debe explorar de manera inteligente una zona de este espacio donde las soluciones parecen más atractivas y esto se realiza una vez que se escoge una zona a explorar, de

manera que se utilizan mecanismos para intensificar la exploración en esta zona, algunos mecanismos utilizan la memoria basado en lo reciente, con la cual se evita repetir movimientos que provoquen tener soluciones cíclicas, otro mecanismo usado en la fase de intensificación es modificar las reglas para considerar un movimiento tabú, de esta manera, se pueden explorar soluciones distintas a las encontradas recientemente. Al intensificar la búsqueda, se tiene el riesgo de explorar una zona con soluciones pobres y por lo tanto, caer en un óptimo local.

## Diversificación

Como se menciona antes, el explorar zonas del espacio de soluciones que contengan soluciones pobres, provoca caer en óptimos locales, inherentemente por la búsqueda local, ante esto, para poder explorar otras zonas más interesantes del espacio donde se pueden encontrar soluciones más prometedoras, es necesario utilizar la diversificación, la cual es una estrategia donde se usan mecanismos para forzar la búsqueda en zonas no exploradas previamente del espacio de soluciones.

Algunos mecanismos usan la memoria basada en las frecuencias, con ésta, al saber cuantas veces se utilizó un movimiento, es posible identificar que movimientos pueden haber conducido a la región actual y por lo tanto, crear una estrategia para alterar la manera en que se utilizan estos movimientos y de esto, cambiar la búsqueda hacia otra región del espacio.

Otro mecanismo para diversificar la búsqueda puede ser reiniciar la búsqueda sobre una de las mejores soluciones encontradas.

También está un mecanismo que utiliza penalizaciones, en el cual, si después de un determinado tiempo, no se logra encontrar mejores soluciones, entonces se aceptan soluciones sumando una penalización al costo de éstas:

$$f(i) = f(i) + \text{penalización}$$

## Oscilación estratégica

Esta metaheurística al realizar una búsqueda local, además de utilizar los mecanismos de memoria para evitar óptimos locales tan pronto, es necesario realizar la diversificación e intensificación de la búsqueda de manera apropiada, para encontrar soluciones más atractivas, para esto, la oscilación estratégica consiste en alternar entre diversificación e intensificación, alrededor de una *barrera de oscilación*, la cual es un valor de la función de costo, la manera en que se puede buscar soluciones con costo menor a la barrera es intensificar la búsqueda hasta que se pasa la barrera de oscilación o bien se cae en un óptimo local, después, se diversifica la búsqueda aceptando soluciones con un costo más cercano a la barrera, hasta pasar ésta, para después, repetir esta alternancia, por lo que se obtiene un comportamiento oscilatorio. En la Figura 4.1 se ilustra la oscilación estratégica.

### 4.3. Aplicado a PRCPSP

Dado un proyecto de actividades, es necesario crear un nuevo proyecto a nivel de subactividades, donde cada actividad se interrumpe a los más  $m$  veces, si  $m$  es mayor o igual a la duración de la actividad, entonces la actividad se divide en subactividades de duración uno. Las subactividades resultantes utilizan la misma cantidad de recursos que la actividad original.

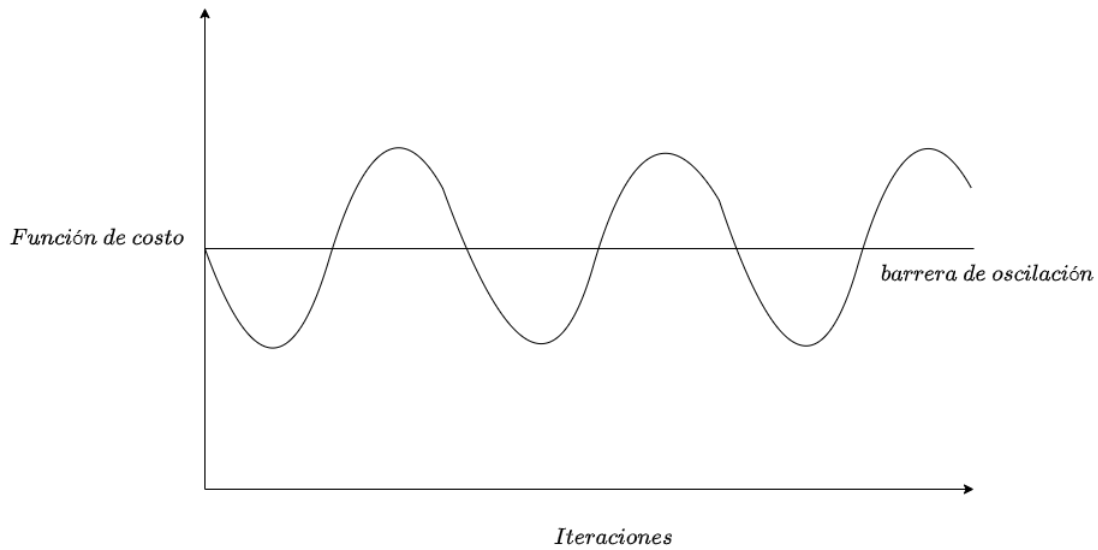


Figura 4.1: Oscilación estratégica

Para utilizar la metaheurística en el PRCPSP, la representación de la solución, como se genera un vecino, la función de costo y la solución inicial, se realizan de la misma forma que en el capítulo anterior con el Recocido Simulado.

Por lo que sólo es necesario definir tres cosas más:

- Los movimientos tabú.
- Criterio de aspiración.
- Diversificación de las soluciones.

### Movimientos tabú

Dado que para generar una nueva solución, se realiza un desplazamiento circular de solo tres actividades en el arreglo  $L$  de la representación, entonces sean  $u, v, w$  las tres subactividades que se intercambian al generar una nueva solución vecina, se agrega la 4-tupla  $(u, v, w, t)$  a la lista tabú  $T$ , donde  $t > 0$  es el tiempo en que este movimiento permanece con el estado de tabú, de manera que no se puede realizar ningún movimiento donde sólo estas tres actividades se intercambian, aunque sí pueden ocurrir movimientos donde solo una o dos de estas actividades se intercambien.

Por lo que en cada iteración de la metaheurística se reduce en una unidad el tiempo  $t$  de todos los movimientos tabú en  $T$  hasta llegar a cero, en tal caso ya se pueden descartar de la lista.

### Criterio de aspiración

La única manera en la que un movimiento tabú puede ser usado es sólo si el movimiento genera una solución con el mejor costo hasta el momento, por lo tanto, es una solución que no se había alcanzado.

### Diversificación de las soluciones

Con el objetivo de tener una oscilación estratégica, en cada iteración se actualiza la solución actual al mejor vecino y que además tenga un costo distinto, pero en caso de que todas las soluciones vecinas

revisadas tengan el mismo costo, se genera una nueva solución vecina y se toma como la solución actual.

Para ejemplificar la diversificación usando la oscilación estratégica, es necesario retomar la planeación de costo 6 mostrada en la Figura 3.1 del capítulo anterior, si se toma como la planeación actual, se puede elegir de entre dos planeaciones vecinas generadas de la siguiente forma:

- Al rotar las actividades alrededor de la actividad 3, se obtiene la planeación y su representación como solución en las Figuras 4.2 (a) y (b) respectivamente, se puede notar que la planeación tiene un costo de 7, por lo cual, es una planeación elegible a ser la actual, pues de acuerdo a la oscilación estratégica, tiene un costo distinto a la planeación actual.
- Al rotar las actividades alrededor de la actividad 6, se obtiene la planeación y su representación como solución en las Figuras 4.3 (a) y (b) respectivamente, se puede notar que la planeación tiene un costo de 6, pero esta planeación no puede ser elegible, pues tiene el mismo costo que la planeación actual y por tanto, no sigue el objetivo de la oscilación estratégica para diversificar la búsqueda.

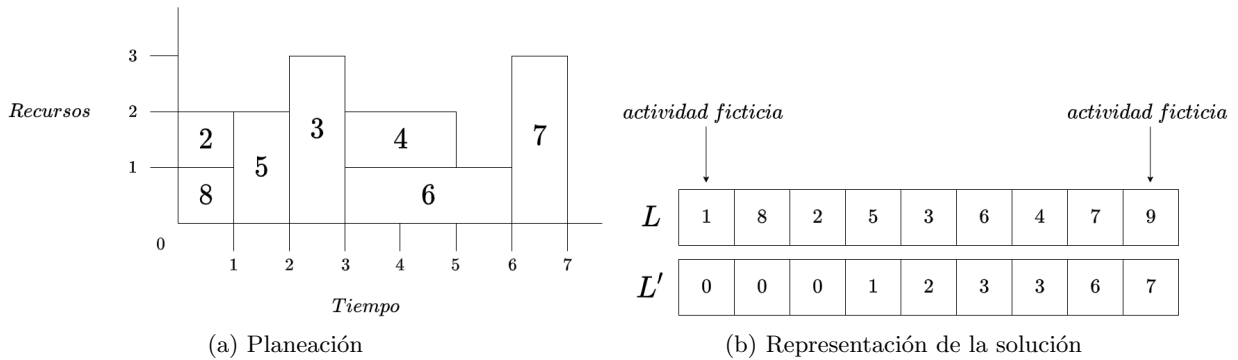


Figura 4.2: Planeación elegible a ser la actual

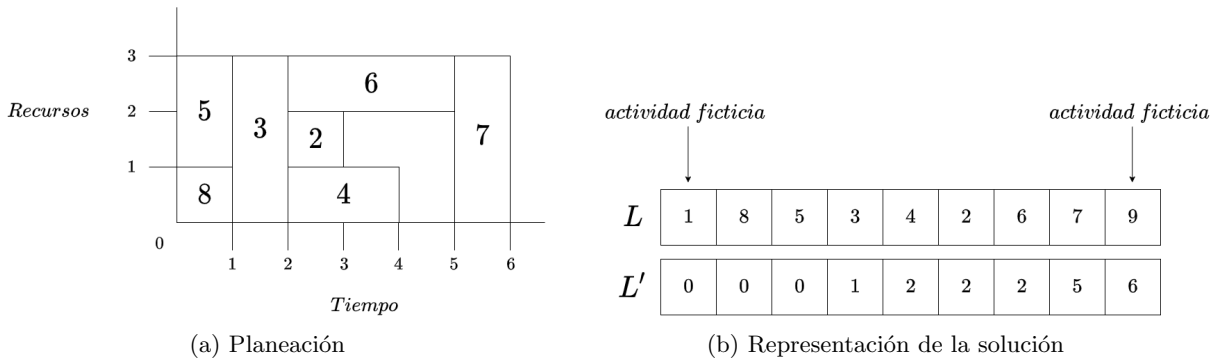


Figura 4.3: Planeación no elegible a ser la actual

## Ejemplo del funcionamiento

Para mostrar el funcionamiento detallado de la metaheurística, se utiliza el mismo proyecto utilizado con el Recocido Simulado del capítulo anterior, el cual se muestra en la Figura 3.26 de 15 actividades (sin contar las dos ficticias 1 y 17), con un costo óptimo de 14 unidades, con 4 recursos de 3 unidades cada uno. Además de la planeación óptima conocida, mostrada anteriormente en la Figura 3.27.

Para el proyecto mostrado anteriormente se utilizan los siguientes parámetros en la metaheurística:

- Número de interrupciones: 2
- Tiempo tabú: 4
- Vecinos admisibles a revisar: 3
- Número de iteraciones: 20

A continuación la ejecución de la metaheurística, mostrando cada iteración en donde se modifica la solución actual después de revisar un conjunto de vecinos dado. En total se muestran 20 iteraciones.

**Iteración 1 Solución actual:**

Al ser la primer iteración, primero se obtiene el proyecto a nivel de subactividades del proyecto original, el cual se muestra en la Figura 4.4. Después se utiliza el Algoritmo 13 sobre este proyecto obteniendo así una planeación inicial con costo 20, la cual se muestra en la Figura 4.5 (a), además de la representación de la solución en la Figura 4.5 (b).

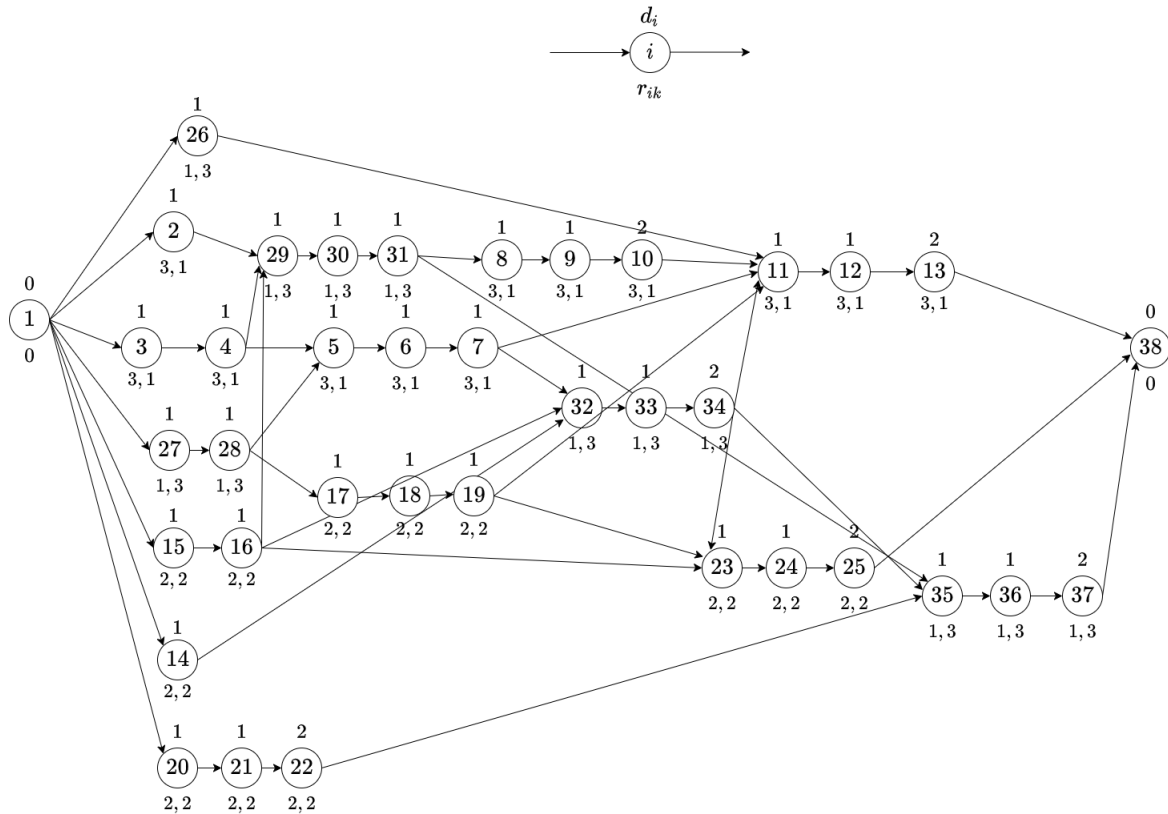
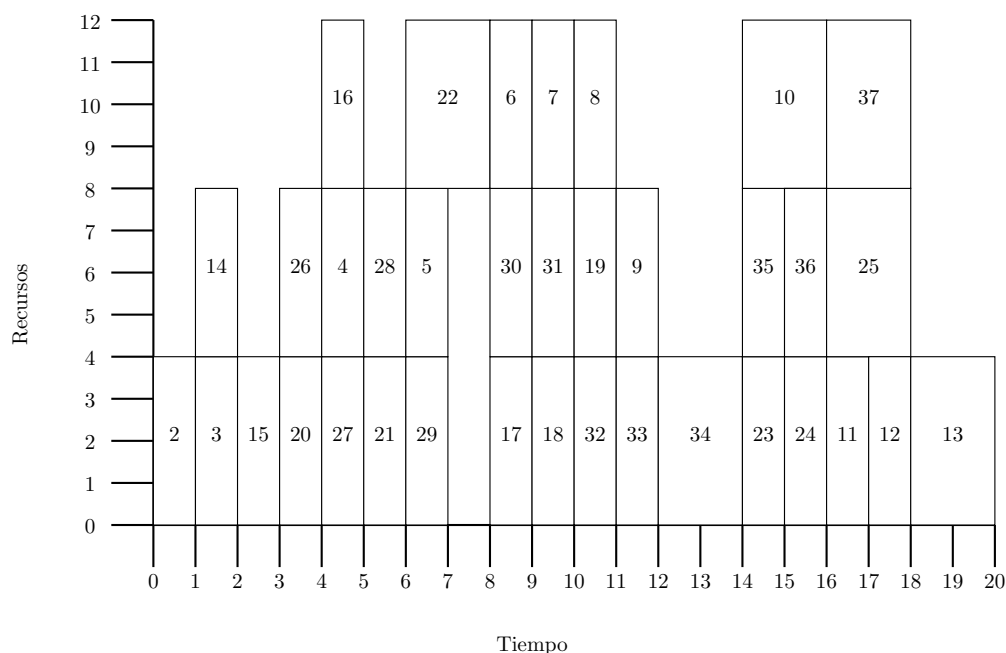


Figura 4.4: Proyecto a nivel de subactividades

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución



(a) Planeación actual en la iteración 1

L	1	2	3	14	15	20	26	27	4	16	21	28	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	23	35	10	24	36	11	25	37	12	13	38
L'	0	0	1	1	2	3	3	4	4	4	5	5	6	6	6	8	8	8	9	9	9	10	10	10	11	11	12	14	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 1

Figura 4.5: Planeación y representación de la solución actual en la iteración 1

tiene un mejor costo que la solución actual, además de ser la mejor solución hasta el momento, por tanto, se toma como la nueva solución actual y como la mejor solución encontrada hasta el momento.

- Vecino 2. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

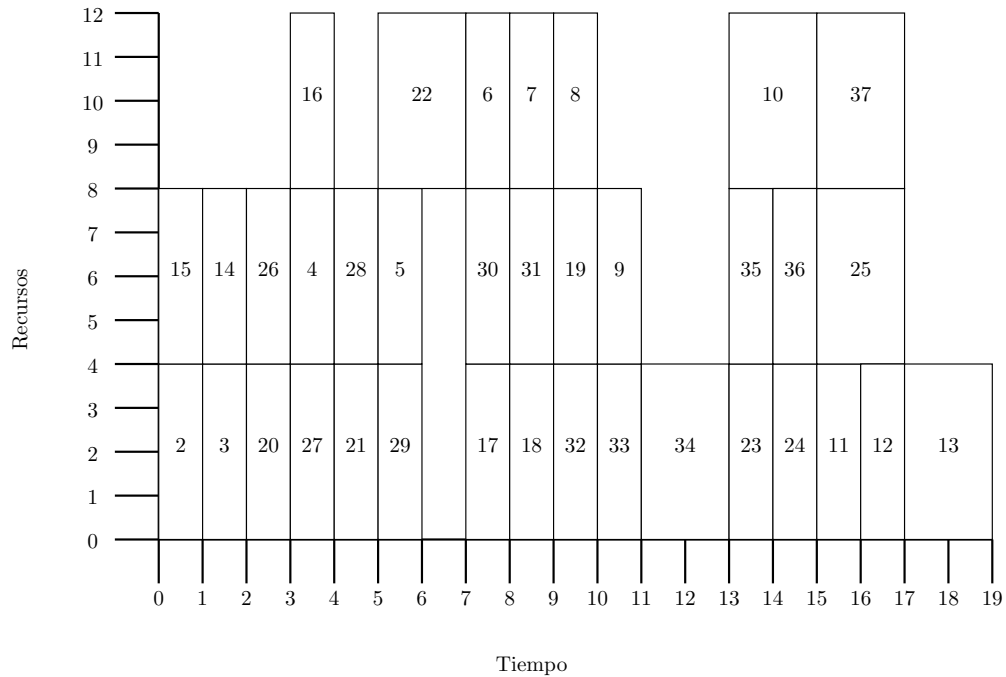
**Lista Tabú:**

Se agregan a la lista tabú las actividades 3, 14, 15, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(3, 14, 15, 4)\}$$

Iteración 2 **Solución actual:**

La solución actual al iniciar la iteración 2 es la solución vecina 1 con costo 19 de la iteración anterior y que se muestra en la Figura 4.6.



(a) Planeación actual en la iteración 2

L	1	2	15	3	14	20	26	27	4	16	21	28	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	23	35	10	24	36	11	25	37	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	7	7	8	8	8	9	9	9	10	10	11	13	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 2

Figura 4.6: Planeación y representación de la solución actual en la iteración 2

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 28 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Esta solución tiene un mejor costo que la solución actual, además de ser la mejor solución hasta el momento, por tanto, se toma como la nueva solución actual y como la mejor solución encontrada hasta el momento.

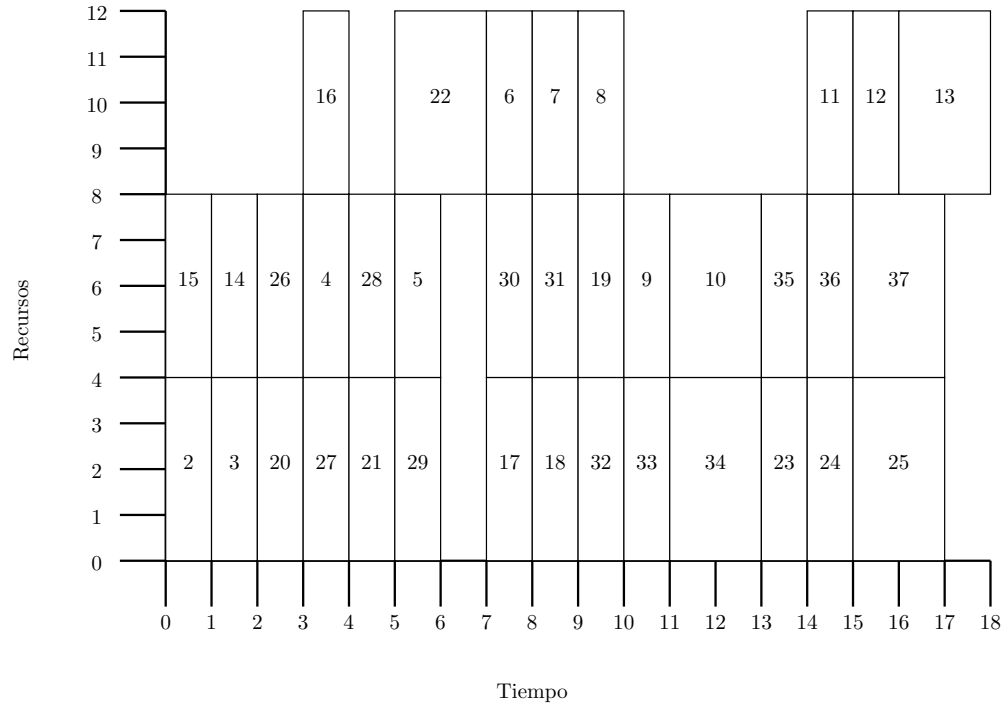
**Lista Tabú:**

Se agregan a la lista tabú las actividades 23, 35, 10, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(3, 14, 15, 3), (23, 35, 10, 4)\}$$

**Iteración 3 Solución actual:**

La solución actual al iniciar la iteración 3 es la solución vecina 3 con costo 18 de la iteración anterior y que se muestra en la Figura 4.7.



(a) Planeación actual en la iteración 3

L	1	2	15	3	14	20	26	27	4	16	21	28	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	11	25	37	12	13	38	
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	14	14	15	15	15	16	18

(b) Representación de la solución actual en la iteración 3

Figura 4.7: Planeación y representación de la solución actual en la iteración 3

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 10 en el arreglo *L* de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.



Vecino 2. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

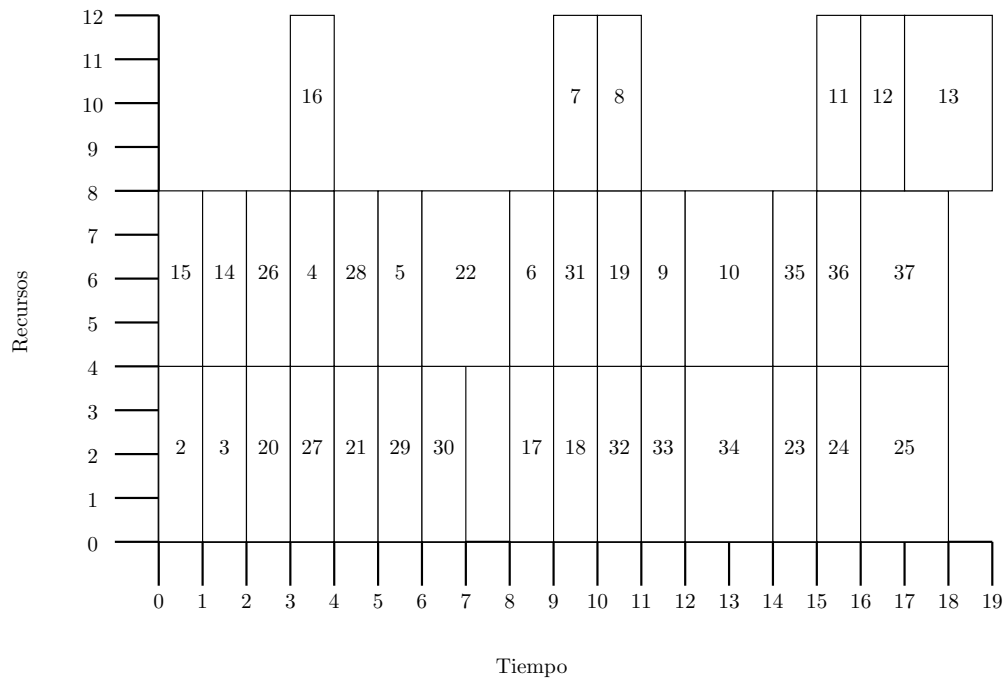
**Lista Tabú:**

Se agregan a la lista tabú las actividades 22, 17, 30, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(3, 14, 15, 2), (23, 35, 10, 3), (22, 17, 30, 4)\}$$

**Iteración 4 Solución actual:**

La solución actual al iniciar la iteración 4 es la solución vecina 3 con costo 19 de la iteración anterior y que se muestra en la Figura 4.8.



(a) Planeación actual en la iteración 4

L	1	2	15	3	14	20	26	27	4	16	21	28	29	5	30	22	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	11	25	37	12	13	38		
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	6	6	8	8	9	9	9	9	10	10	10	11	11	12	12	14	14	15	15	15	15	16	16	16	17	19

(b) Representación de la solución actual en la iteración 4

Figura 4.8: Planeación y representación de la solución actual en la iteración 4

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 32 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 33 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 14 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 5, 30, 22, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(3, 14, 15, 1), (23, 35, 10, 2), (22, 17, 30, 3), (5, 30, 22, 4)\}$$

**Iteración 5 Solución actual:**

La solución actual al iniciar la iteración 5 es la solución vecina 3 con costo 18 de la iteración anterior y que se muestra en la Figura 4.9.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 32 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 2, 15, 3, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

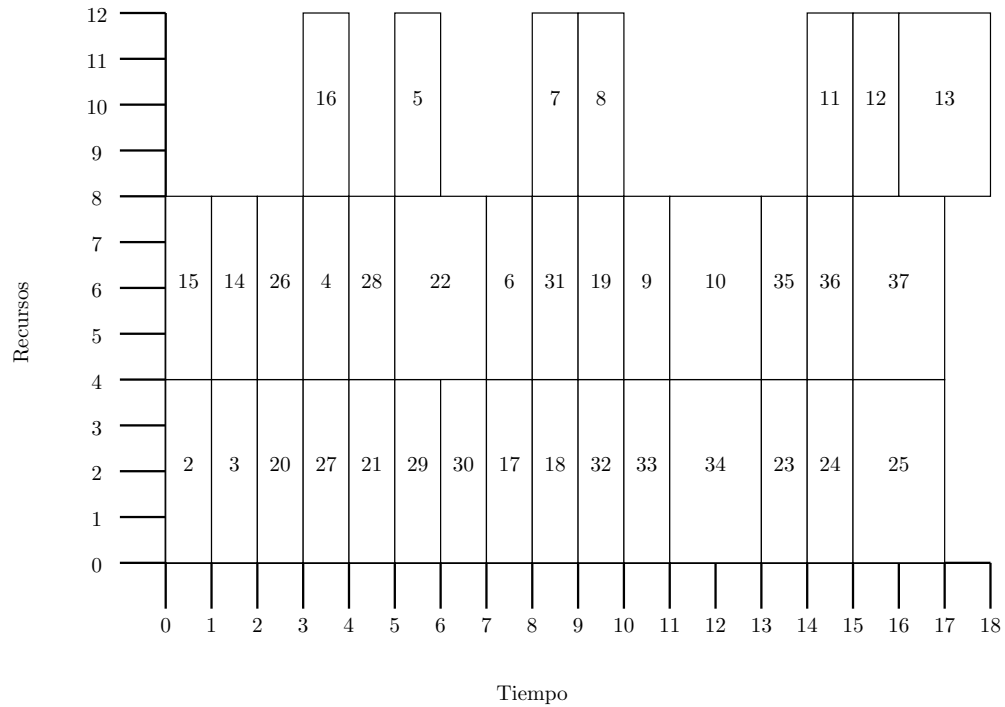
$$T = \{(23, 35, 10, 1), (22, 17, 30, 2), (5, 30, 22, 3), (2, 15, 3, 4)\}$$

**Iteración 6 Solución actual:**

La solución actual al iniciar la iteración 6 es la solución vecina 3 con costo 19 de la iteración anterior y que se muestra en la Figura 4.10.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.



(a) Planeación actual en la iteración 5

L	1	2	15	3	14	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	11	25	37	12	13	38	
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	6	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	14	14	15	15	15	16	18

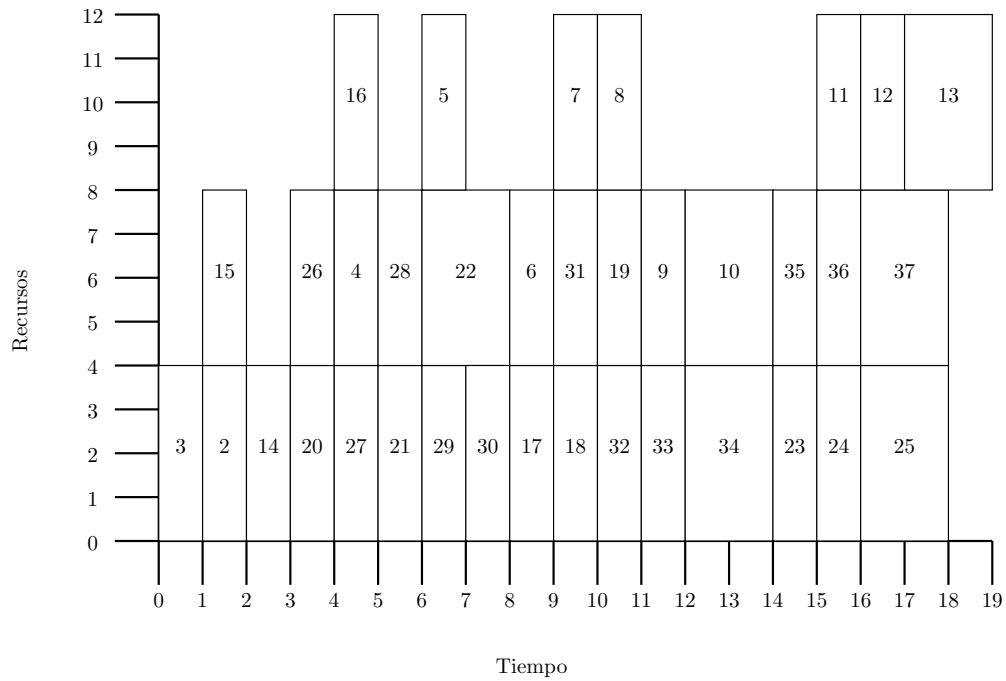
(b) Representación de la solución actual en la iteración 5

Figura 4.9: Planeación y representación de la solución actual en la iteración 5

- Vecino 1. Rotando las actividades alrededor de la posición 32 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 18. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 33 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 2, 15, 14, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.



(a) Planeación actual en la iteración 6

L	1	3	2	15	14	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	11	25	37	12	13	38	
L'	0	0	1	1	2	3	3	4	4	4	5	5	6	6	6	7	8	8	9	9	9	10	10	10	11	11	12	12	14	14	15	15	15	15	16	16	16	17	19

(b) Representación de la solución actual en la iteración 6

Figura 4.10: Planeación y representación de la solución actual en la iteración 6

$$T = \{(22, 17, 30, 1), (5, 30, 22, 2), (2, 15, 3, 3), (2, 15, 14, 4)\}$$

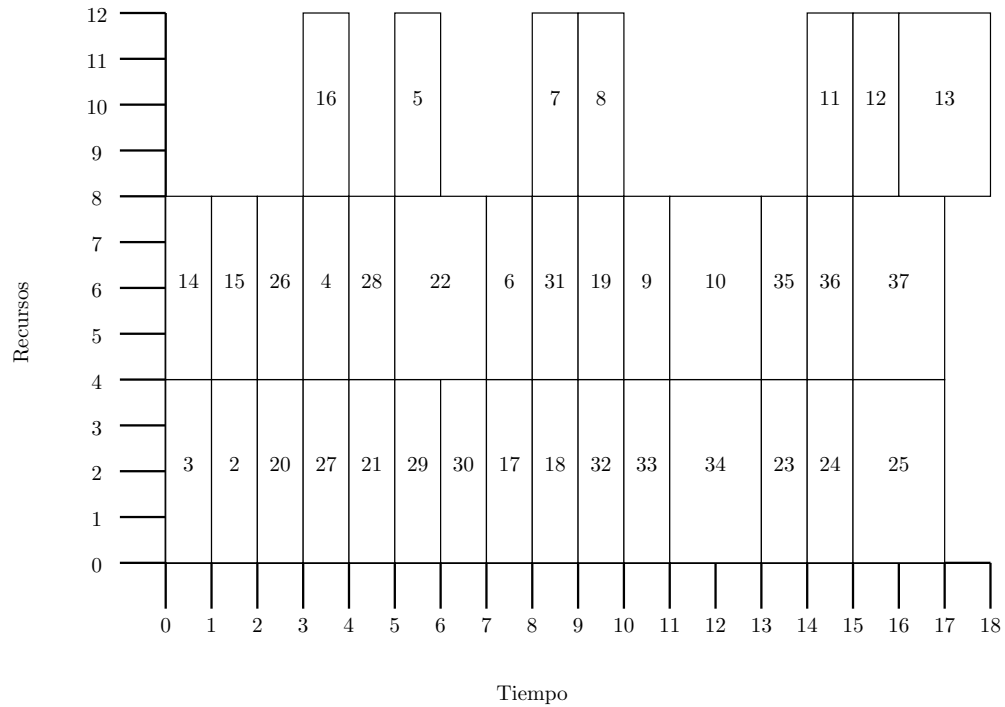
**Iteración 7 Solución actual:**

La solución actual al iniciar la iteración 7 es la solución vecina 2 con costo 18 de la iteración anterior y que se muestra en la Figura 4.11.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 32 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 33 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución



(a) Planeación actual en la iteración 7

L	1	3	14	2	15	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	11	25	37	12	13	38		
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	6	7	7	8	8	8	9	9	9	9	10	10	11	11	13	13	14	14	14	14	15	15	15	16	18

(b) Representación de la solución actual en la iteración 7

Figura 4.11: Planeación y representación de la solución actual en la iteración 7

tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 11, 25, 37, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(5, 30, 22, 1), (2, 15, 3, 2), (2, 15, 14, 3), (11, 25, 37, 4)\}$$

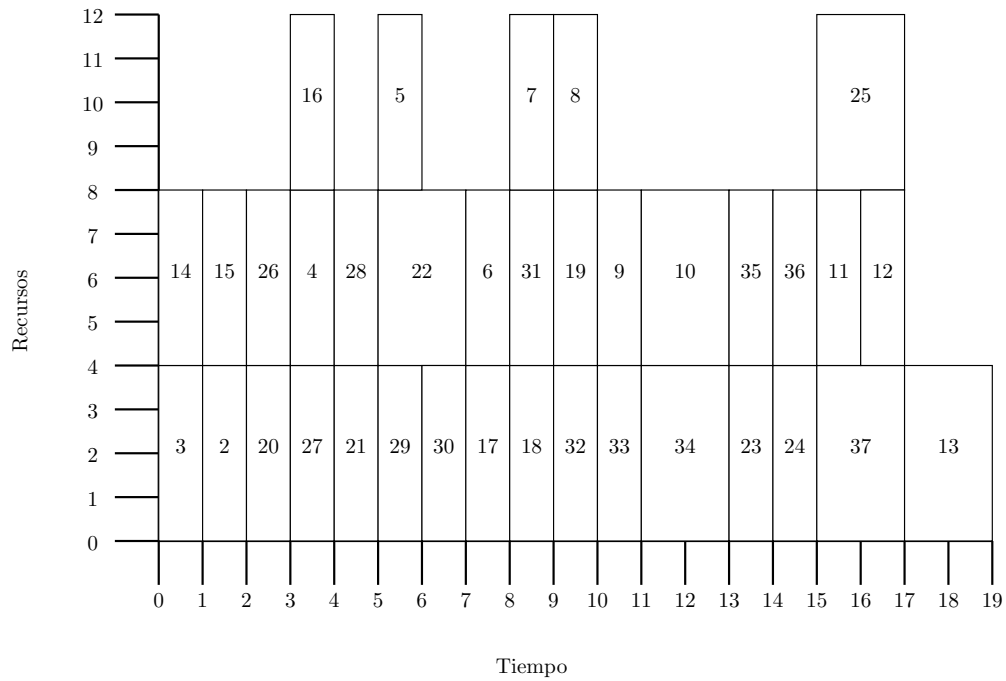
**Iteración 8 Solución actual:**

La solución actual al iniciar la iteración 8 es la solución vecina 3 con costo 19 de la iteración anterior y que se muestra en la Figura 4.12.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo *L* de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no



(a) Planeación actual en la iteración 8

L	1	3	14	2	15	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	6	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 8

Figura 4.12: Planeación y representación de la solución actual en la iteración 8

es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.

- Vecino 2. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

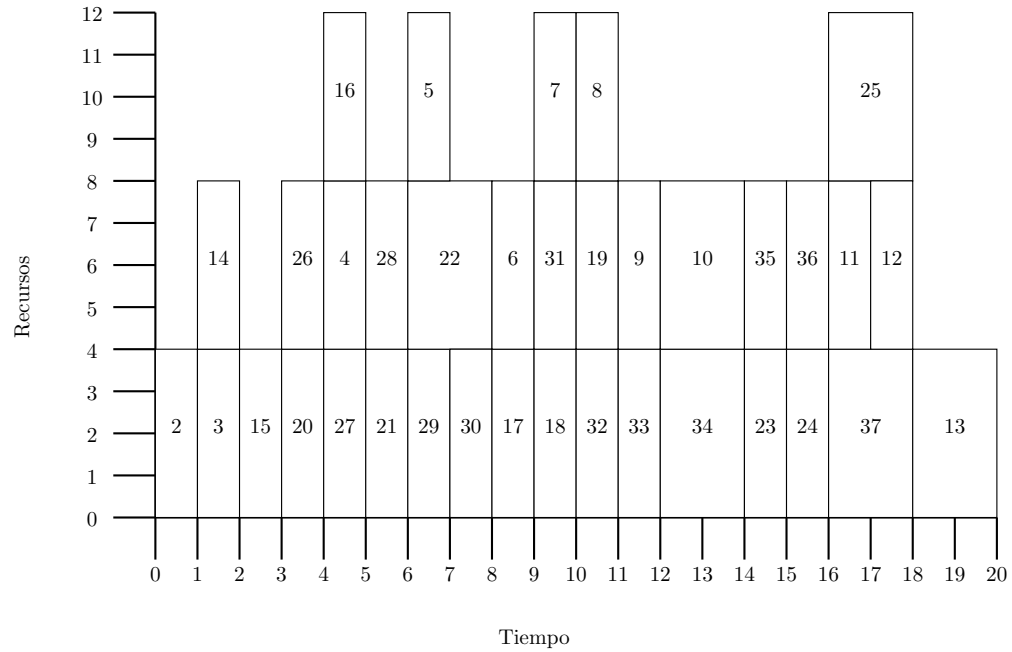
**Lista Tabú:**

Se agregan a la lista tabú las actividades 3, 14, 2, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(2, 15, 3, 1), (2, 15, 14, 2), (11, 25, 37, 3), (3, 14, 2, 4)\}$$

**Iteración 9 Solución actual:**

La solución actual al iniciar la iteración 9 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.13.



(a) Planeación actual en la iteración 9

L	1	2	3	14	15	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38	
L'	0	0	1	1	2	3	3	4	4	4	4	5	5	6	6	6	7	8	8	9	9	9	10	10	10	11	11	12	12	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 9

Figura 4.13: Planeación y representación de la solución actual en la iteración 9

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

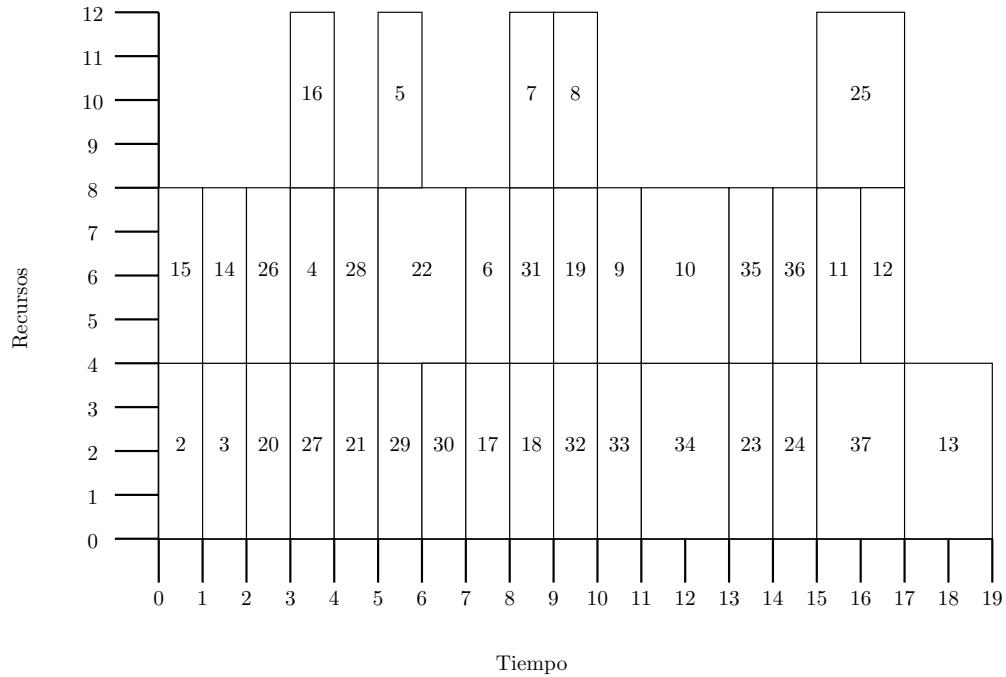
**Lista Tabú:**

Se agregan a la lista tabú las actividades 3, 14, 15, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(2, 15, 14, 1), (11, 25, 37, 2), (3, 14, 2, 3), (3, 14, 15, 4)\}$$

**Iteración 10 Solución actual:**

La solución actual al iniciar la iteración 10 es la solución vecina 1 con costo 19 de la iteración anterior y que se muestra en la Figura 4.14.



(a) Planeación actual en la iteración 10

L	1	2	15	3	14	20	26	27	4	16	21	28	29	22	5	30	17	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	6	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 10

Figura 4.14: Planeación y representación de la solución actual en la iteración 10

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.



Vecino 3. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

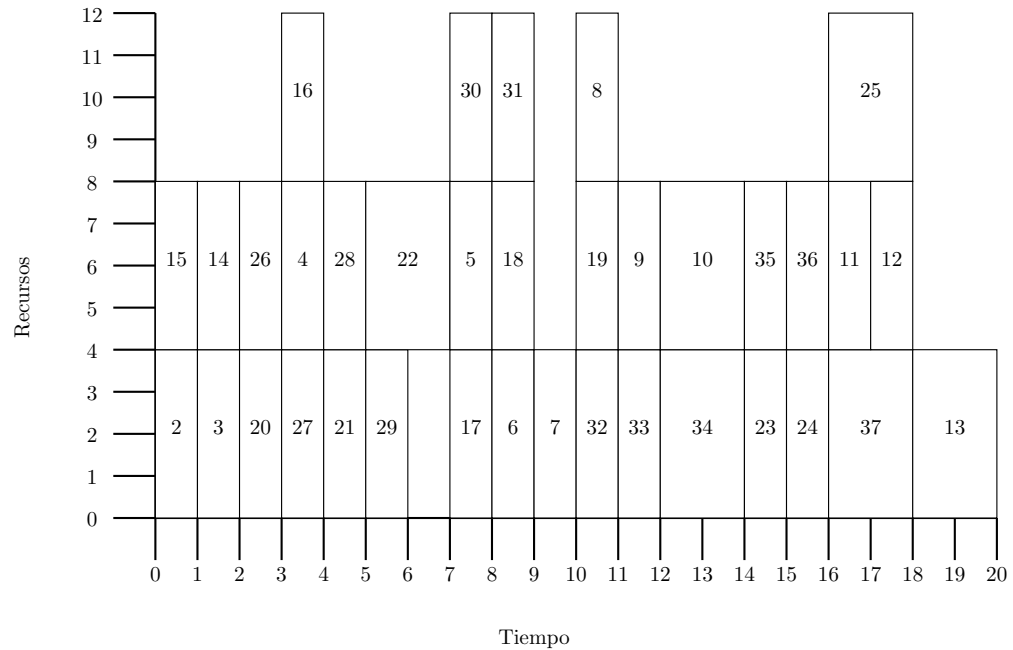
**Lista Tabú:**

Se agregan a la lista tabú las actividades 5, 30, 17, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(11, 25, 37, 1), (3, 14, 2, 2), (3, 14, 15, 3), (5, 30, 17, 4)\}$$

**Iteración 11 Solución actual:**

La solución actual al iniciar la iteración 11 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.15.



(a) Planeación actual en la iteración 11

L	1	2	15	3	14	20	26	27	4	16	21	28	29	22	17	5	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	7	7	7	8	8	8	9	10	10	10	11	11	12	12	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 11

Figura 4.15: Planeación y representación de la solución actual en la iteración 11

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no

es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 14 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.

#### **Lista Tabú:**

Se agregan a la lista tabú las actividades 22, 17, 5, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(3, 14, 2, 1), (3, 14, 15, 2), (5, 30, 17, 3), (22, 17, 5, 4)\}$$

#### **Iteración 12 Solución actual:**

La solución actual al iniciar la iteración 12 es la solución vecina 3 con costo 19 de la iteración anterior y que se muestra en la Figura 4.16.

#### **Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

Vecino 1. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.

Vecino 2. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 10 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

#### **Lista Tabú:**

Se agregan a la lista tabú las actividades 16, 21, 28, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

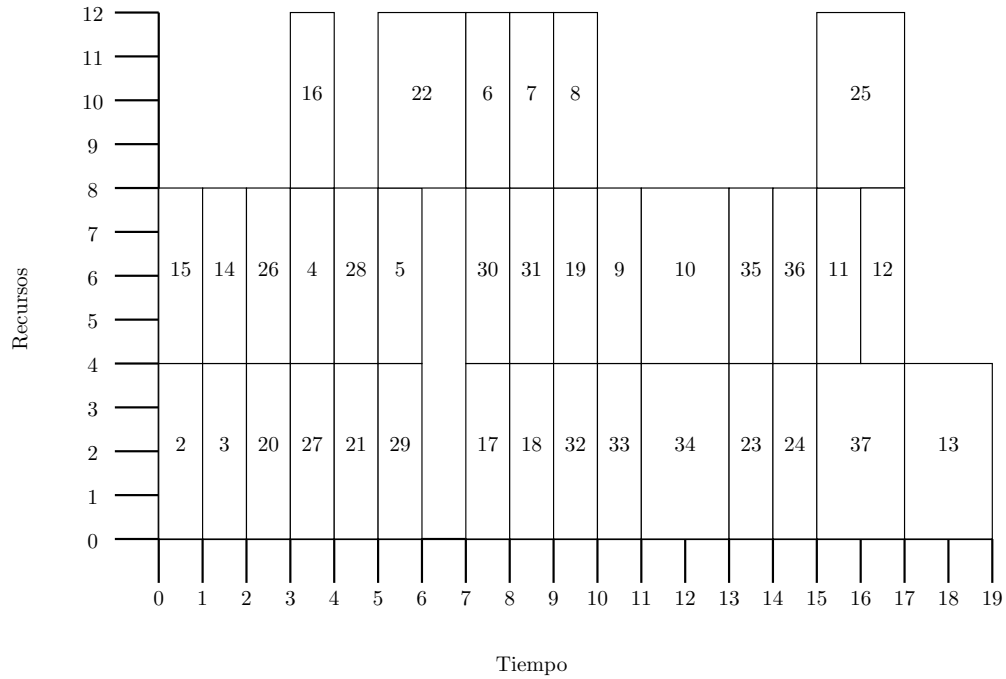
$$T = \{(3, 14, 15, 1), (5, 30, 17, 2), (22, 17, 5, 3), (16, 21, 28, 4)\}$$

#### **Iteración 13 Solución actual:**

La solución actual al iniciar la iteración 13 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.17.

#### **Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.



(a) Planeación actual en la iteración 12

L	1	2	15	3	14	20	26	27	4	16	21	28	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 12

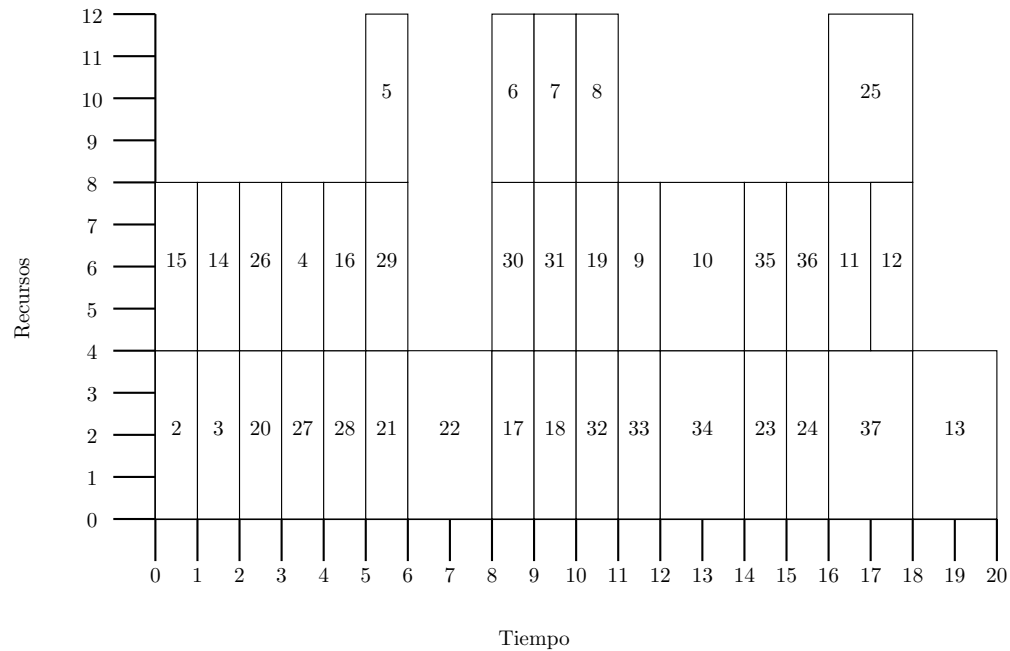
Figura 4.16: Planeación y representación de la solución actual en la iteración 12

- Vecino 1. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 20, 26, 27, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(5, 30, 17, 1), (22, 17, 5, 2), (16, 21, 28, 3), (20, 26, 27, 4)\}$$



(a) Planeación actual en la iteración 13

L	1	2	15	3	14	20	26	27	4	28	16	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38	
L'	0	0	0	1	1	2	2	3	3	4	4	5	5	5	6	8	8	8	8	9	9	9	10	10	10	11	11	12	12	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 13

Figura 4.17: Planeación y representación de la solución actual en la iteración 13

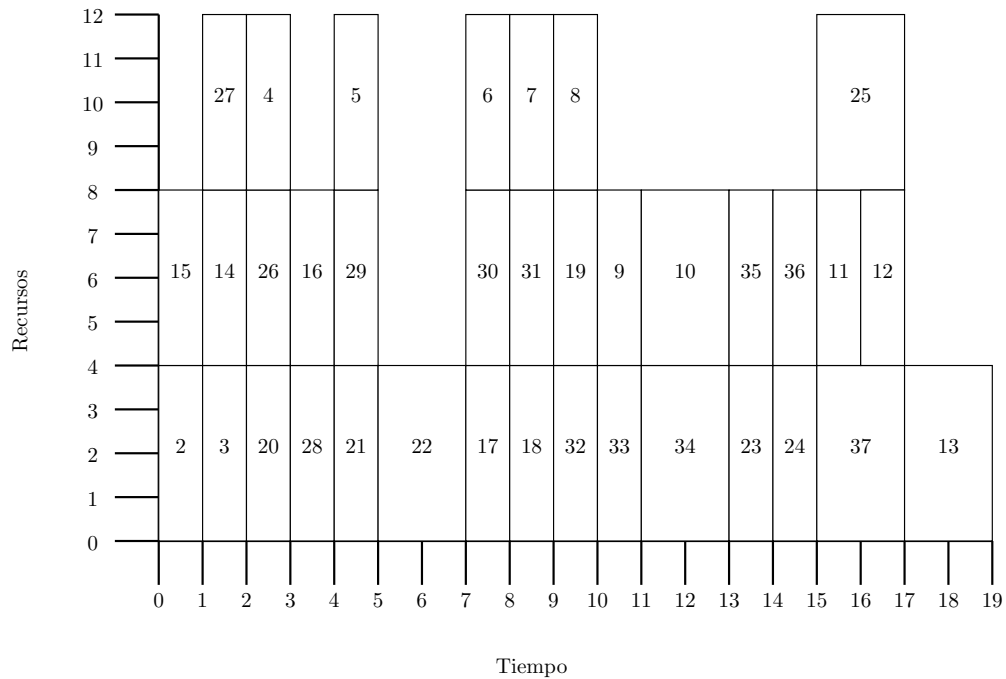
**Iteración 14 Solución actual:**

La solución actual al iniciar la iteración 14 es la solución vecina 2 con costo 19 de la iteración anterior y que se muestra en la Figura 4.18.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 5 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.



(a) Planeación actual en la iteración 14

L	1	2	15	3	14	27	20	26	4	28	16	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	1	2	2	2	3	3	4	4	4	5	7	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 14

Figura 4.18: Planeación y representación de la solución actual en la iteración 14

**Lista Tabú:**

Se agregan a la lista tabú las actividades 14, 27, 20, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(22, 17, 5, 1), (16, 21, 28, 2), (20, 26, 27, 3), (14, 27, 20, 4)\}$$

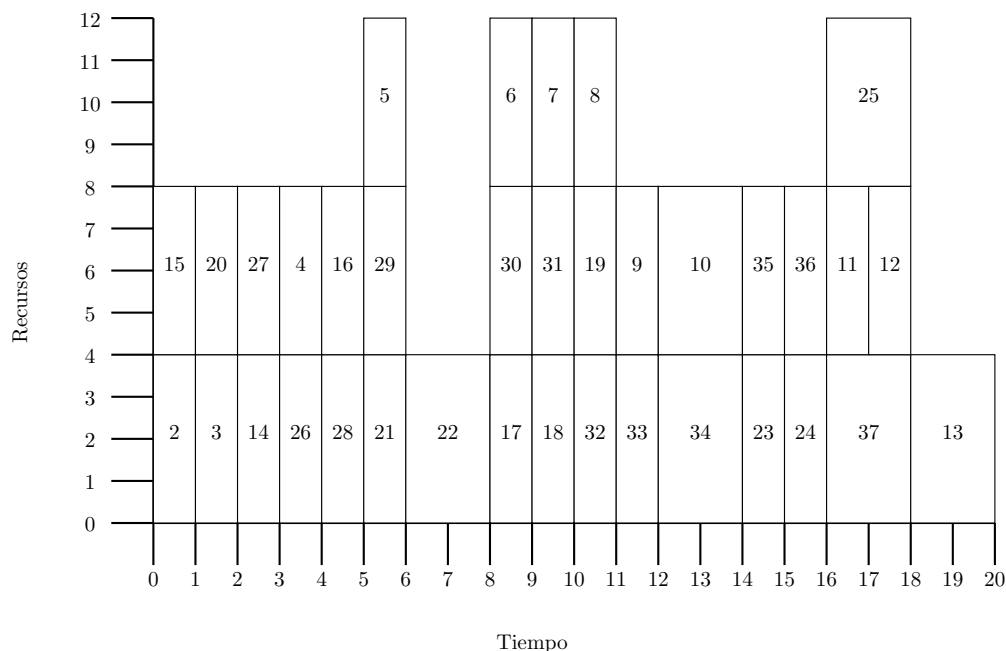
**Iteración 15 Solución actual:**

La solución actual al iniciar la iteración 15 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.19.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 9 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene



(a) Planeación actual en la iteración 15

L	1	2	15	3	20	14	27	26	4	28	16	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	4	4	5	5	5	6	8	8	8	9	9	9	10	10	10	11	11	12	12	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 15

Figura 4.19: Planeación y representación de la solución actual en la iteración 15

un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 18 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 4, 28, 16, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

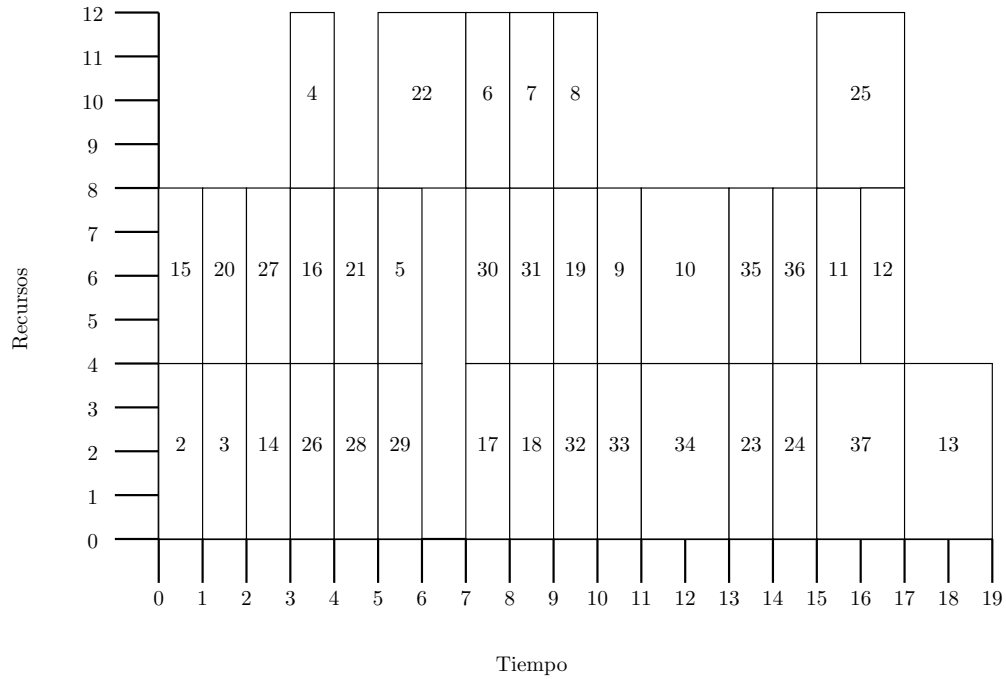
$$T = \{(16, 21, 28, 1), (20, 26, 27, 2), (14, 27, 20, 3), (4, 28, 16, 4)\}$$

**Iteración 16 Solución actual:**

La solución actual al iniciar la iteración 16 es la solución vecina 1 con costo 19 de la iteración anterior y que se muestra en la Figura 4.20.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.



(a) Planeación actual en la iteración 16

L	1	2	15	3	20	14	27	26	16	4	28	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	7	7	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 16

Figura 4.20: Planeación y representación de la solución actual en la iteración 16

- Vecino 1. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

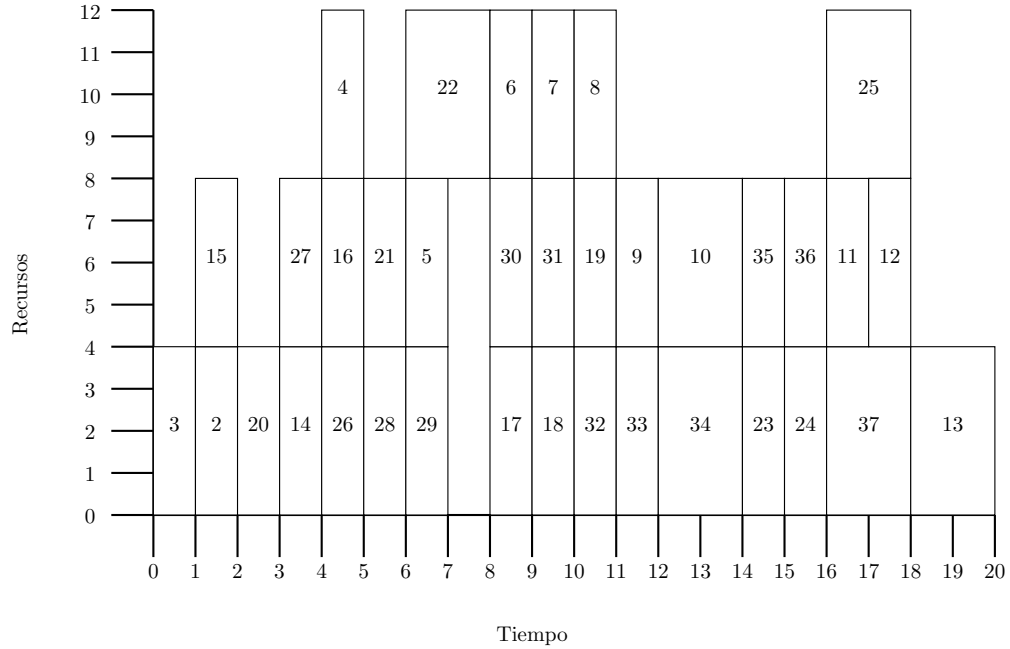
**Lista Tabú:**

Se agregan a la lista tabú las actividades 2, 15, 3, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(20, 26, 27, 1), (14, 27, 20, 2), (4, 28, 16, 3), (2, 15, 3, 4)\}$$

**Iteración 17 Solución actual:**

La solución actual al iniciar la iteración 17 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.21.



(a) Planeación actual en la iteración 17

L	1	3	2	15	20	14	27	26	16	4	28	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38			
L'	0	0	1	1	2	3	3	4	4	4	4	5	5	6	6	6	8	8	8	8	9	9	9	9	10	10	10	11	11	12	12	14	14	15	15	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 17

Figura 4.21: Planeación y representación de la solución actual en la iteración 17

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 3 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 19. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene un costo mayor a la actual, por tanto, no se puede tomar como la nueva solución actual.



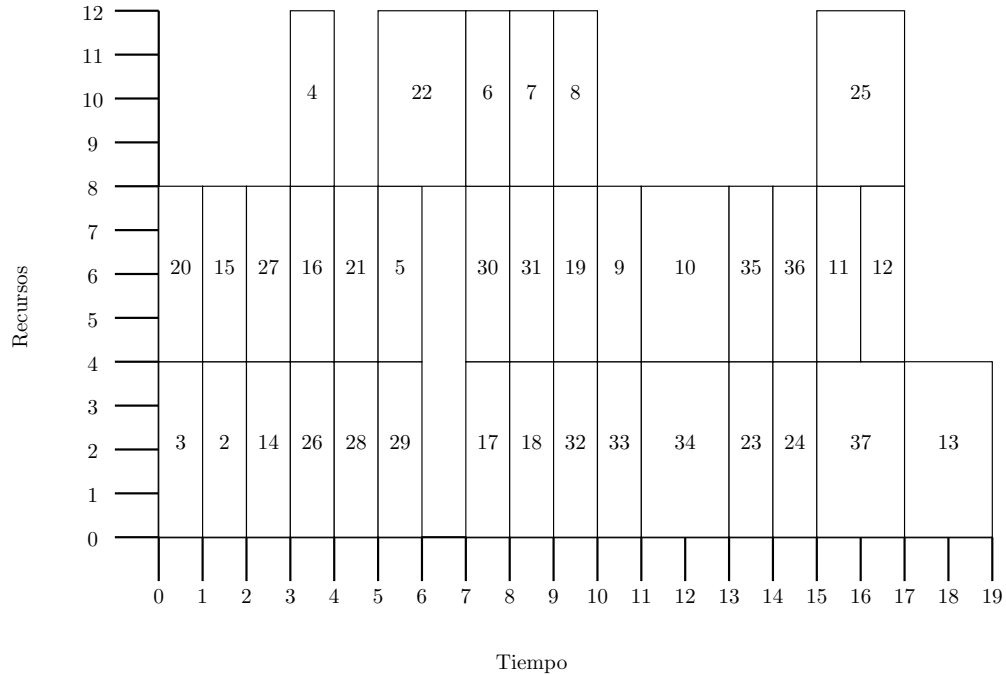
**Lista Tabú:**

Se agregan a la lista tabú las actividades 2, 15, 20, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(14, 27, 20, 1), (4, 28, 16, 2), (2, 15, 3, 3), (2, 15, 20, 4)\}$$

**Iteración 18 Solución actual:**

La solución actual al iniciar la iteración 18 es la solución vecina 1 con costo 19 de la iteración anterior y que se muestra en la Figura 4.22.



(a) Planeación actual en la iteración 18

L	1	3	20	2	15	14	27	26	16	4	28	21	29	5	22	17	30	6	18	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38	
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	7	7	8	8	8	8	9	9	9	10	10	11	11	13	13	14	14	15	15	15	16	17	19

(b) Representación de la solución actual en la iteración 18

Figura 4.22: Planeación y representación de la solución actual en la iteración 18

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 15 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución

tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

Vecino 3. Rotando las actividades alrededor de la posición 17 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

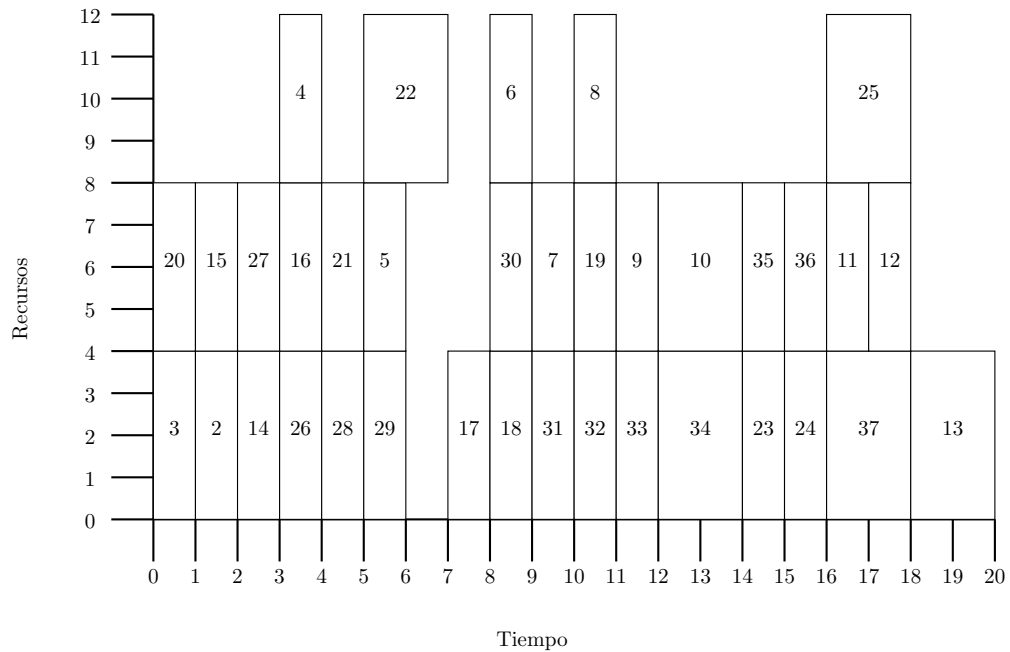
**Lista Tabú:**

Se agregan a la lista tabú las actividades 30, 6, 18, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(4, 28, 16, 1), (2, 15, 3, 2), (2, 15, 20, 3), (30, 6, 18, 4)\}$$

**Iteración 19 Solución actual:**

La solución actual al iniciar la iteración 19 es la solución vecina 3 con costo 20 de la iteración anterior y que se muestra en la Figura 4.23.



(a) Planeación actual en la iteración 19

L	1	3	20	2	15	14	27	26	16	4	28	21	29	5	22	17	18	30	6	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38			
L'	0	0	0	1	1	2	2	3	3	3	4	4	5	5	5	7	8	8	8	8	9	9	10	10	10	10	11	11	12	12	14	14	15	15	16	16	16	16	17	18	20

(b) Representación de la solución actual en la iteración 19

Figura 4.23: Planeación y representación de la solución actual en la iteración 19

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

- Vecino 1. Rotando las actividades alrededor de la posición 12 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 7 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 21. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 27, 26, 16, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(2, 15, 3, 1), (2, 15, 20, 2), (30, 6, 18, 3), (27, 26, 16, 4)\}$$

**Iteración 20 Solución actual:**

La solución actual al iniciar la iteración 20 es la solución vecina 3 con costo 21 de la iteración anterior y que se muestra en la Figura 4.24.

**Soluciones vecinas a revisar y su costo:**

Se muestran las tres soluciones vecinas admisibles a revisar.

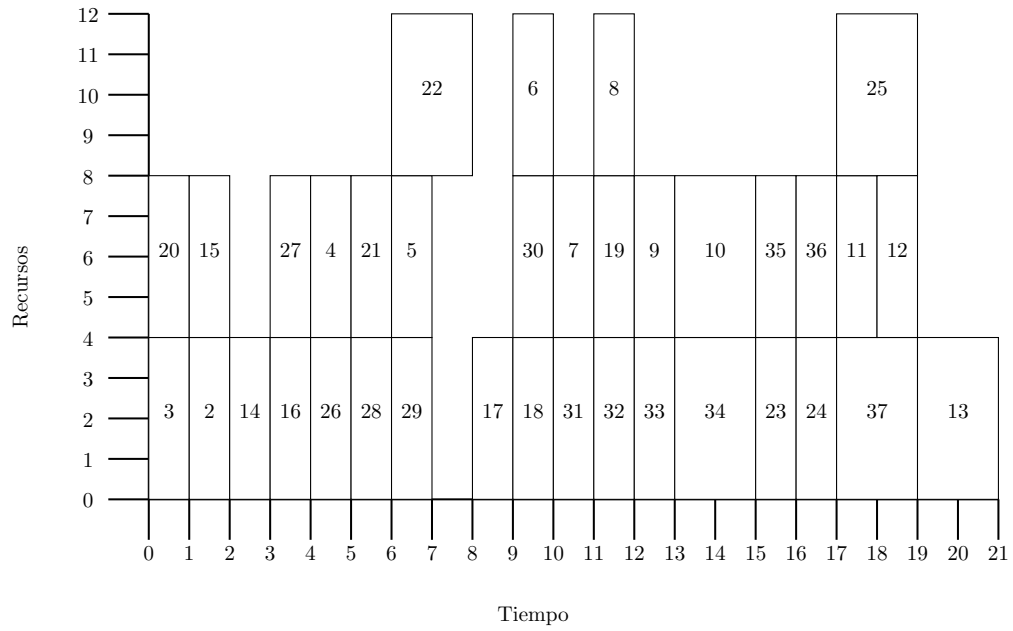
- Vecino 1. Rotando las actividades alrededor de la posición 11 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución no es mejor que la actual al inicio de la iteración, pero tiene un costo distinto, por lo que de acuerdo a la oscilación estratégica, se toma como la nueva solución actual.
- Vecino 2. Rotando las actividades alrededor de la posición 2 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 22. Esta solución tiene el mismo costo que la solución actual y su costo es distinto al de la solución al principio de la iteración, por tanto, se toma como la nueva solución actual.
- Vecino 3. Rotando las actividades alrededor de la posición 6 en el arreglo  $L$  de la representación de la solución actual se obtiene una planeación con costo 20. Esta solución tiene un costo distinto a la actual y es la de mejor costo de entre las soluciones admisibles con costo diferente a la actual, por tanto, se toma como la nueva solución actual.

**Lista Tabú:**

Se agregan a la lista tabú las actividades 14, 16, 27, que fueron intercambiadas en el movimiento realizado y se decrementa en una unidad el tiempo tabú de los demás registros en la lista tabú.

$$T = \{(2, 15, 20, 1), (30, 6, 18, 2), (27, 26, 16, 3), (14, 16, 27, 4)\}$$

**Resultados** Después de 20 iteraciones, la mejor solución se encontró en la iteración 2 con un costo de 18 unidades, la cual se muestra en la Figura 4.7. En la Figura 4.25 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística.



(a) Planeación actual en la iteración 20

L	1	3	20	2	15	14	16	27	26	4	28	21	29	5	22	17	18	30	6	31	7	32	19	8	33	9	34	10	23	35	24	36	37	11	25	12	13	38	
L'	0	0	0	1	1	2	3	3	4	4	5	5	6	6	6	8	9	9	9	10	10	11	11	11	11	12	12	13	13	15	15	16	16	17	17	17	18	19	21

(b) Representación de la solución actual en la iteración 20

Figura 4.24: Planeación y representación de la solución actual en la iteración 20

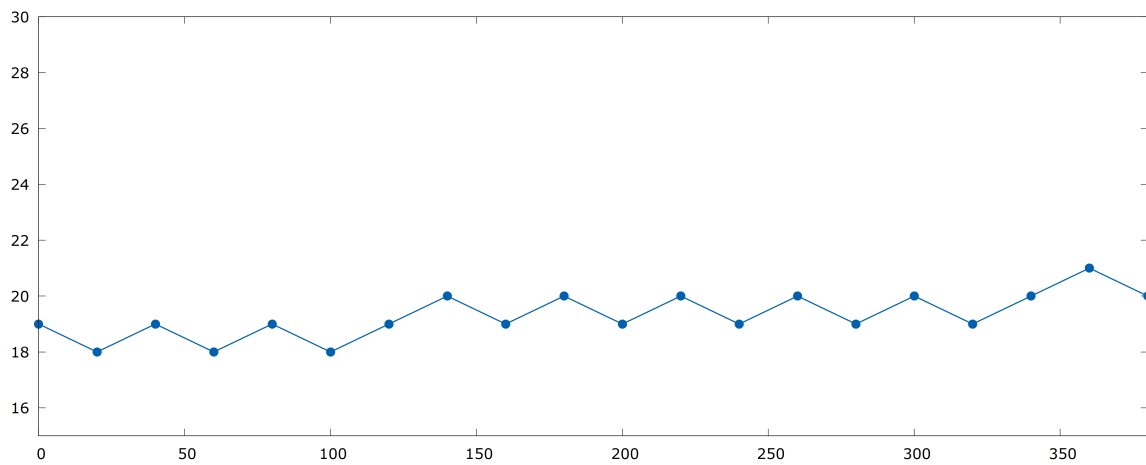


Figura 4.25: Soluciones visitadas durante la ejecución de la Búsqueda Tabú para el proyecto de 15 actividades

## Capítulo 5

# Comparación de las metaheurísticas para el PRCPSP

Una de las grandes dificultades en el uso de las metaheurísticas es la correcta asignación de parámetros para el problema, no se conoce cómo calcular los parámetros ideales en una metaheurística para obtener las mejores soluciones al problema, por lo que la única manera de saber que parámetros son los más adecuados para el problema a optimizar es mediante la experimentación en la práctica.

En este capítulo se trata la generación de ejemplares y los resultados de las metaheurísticas. En la sección 5.1 se define la manera en que se construyen ejemplares aleatorios de manera controlada junto a su planeación óptima. En la sección 5.2 se comparan los resultados obtenidos en la práctica de las dos metaheurísticas de interés aplicadas a ejemplares de tamaño 10, 20, 40 y 80. Aunque los ejemplares no tienen un tamaño significativo, al usar interrupciones, su tamaño incrementa, por lo que es interesante analizar ejemplares donde el número de actividades aumenta al doble.

### 5.1. Generación de ejemplares aleatorios

El objetivo de generar ejemplares aleatorios de manera controlada es porque de éstos se puede saber cual es la planeación óptima y esto ayuda a comparar el desempeño de las metaheurísticas más adelante.

Para generar los ejemplares son necesarios los siguientes parámetros:

- La **semilla** para generar números pseudo-aleatorios y tener ejemplares reproducibles.
- El **costo óptimo** de la planeación a generar.
- El **número de actividades** a generar, sin contar las dos actividades ficticias.
- El **número de recursos** en el proyecto.
- La **suma total de las capacidades de los recursos**.

Dado un costo óptimo  $OPT$ , el ejemplar con una planeación de este costo se construye de la siguiente manera:

La duración que puede tener una actividad va desde 1, 2, 3, 4, .. hasta un entero  $z$ , tal que la suma acumulada de la duración de actividades sea menor igual a  $OPT$ , esto es:

$$1 + 2 + 3 + 4 + \dots + z \leq OPT$$

Si la suma de duraciones no es igual a  $OPT$ , se va a tener una duración  $d_r > 0$  como la diferencia entre  $OPT$  y la suma acumulada de la duración de las actividades:

$$d_r = OPT - \sum_{i=1}^z i$$

Por lo que las actividades generadas van a tener alguna de estas duraciones que van desde 1 hasta  $z$ , además de  $d_r$  (si existe) , esto se ilustra en la Figura 5.1.

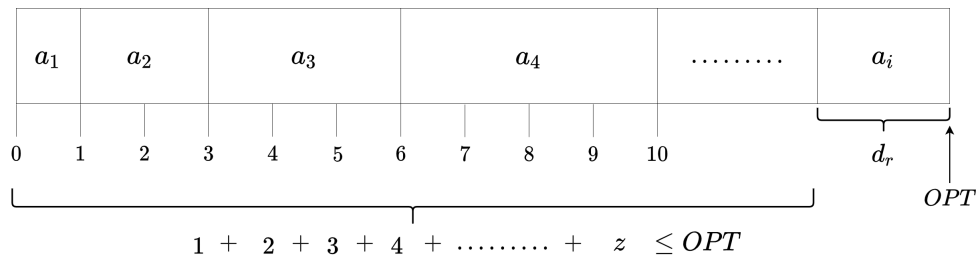


Figura 5.1: Duración de actividades

En cuanto a los recursos, tienen la misma capacidad, excepto el último recurso que además contiene lo restante al dividir la suma total de capacidades de los recursos entre la cantidad de recursos. Todas las actividades usan la misma cantidad de recursos (no necesariamente los mismos recursos), excepto las últimas actividades que usan las unidades restantes de más.

Como se busca una planeación óptima, de manera que todos los recursos se usen, se asigna la primera actividad con duración 1 en el tiempo 0, la segunda con duración 2 en el tiempo 1, la tercera con duración 3 en el tiempo 3 y así hasta  $z$  y la duración  $d_r$ , después, si aún quedan actividades por planear, se repite la asignación de tiempos de duración, aunque existe un caso especial, que es cuando solo queda la última actividad por planearse, como se busca tener una planeación óptima, es necesario que no queden espacios, por lo que la última actividad en planearse tiene como duración la diferencia entre el costo óptimo con el tiempo en que esta actividad se planea, sea  $d_l$  esta duración. En la Figura 5.2 se ilustra la forma de una planeación óptima.

La planeación construida de esta forma es óptima porque todos los recursos se utilizan en todos los tiempos, por lo que no puede haber una planeación con menor tiempo, pues si ésta existiera, tendría que haber una mayor disponibilidad de recursos, pero en la planeación construida ningún recurso está disponible, o visto de otra forma; tendrían que haber espacios, pero la asignación de recursos a las actividades y la duración de éstas, hacen que no quede algún espacio en la planeación.

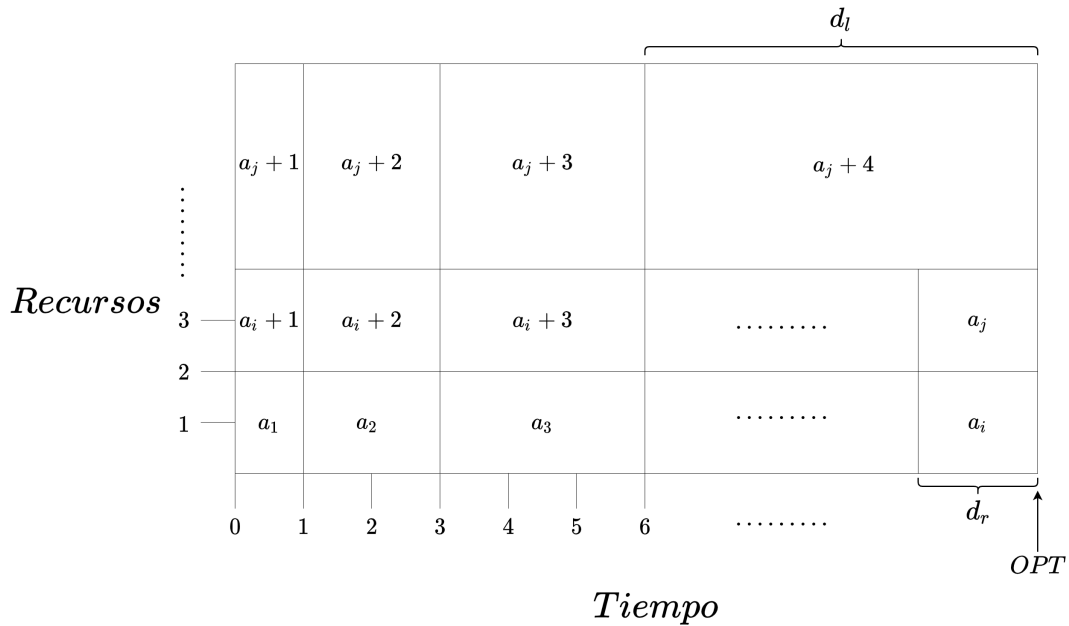


Figura 5.2: Forma de una planeación óptima

La parte aleatoria consiste en asignar la precedencia de actividades, se tiene que todas las actividades pueden tener a lo más  $\lfloor \sqrt{n} \rfloor$  sucesores; de donde, para cada actividad se escoge un número aleatorio entre 1 y  $\lfloor \sqrt{n} \rfloor$ , este número es la cantidad de actividades sucesoras que se tienen que asignar de manera aleatoria, como se deben escoger actividades sucesoras, éstas deben iniciar en un tiempo mayor o igual al tiempo en que termina la actividad a la que se le asignan sucesores.

Se tienen tres casos especiales:

- En el primer caso están las actividades que no tienen actividades predecesoras, debido a que durante la asignación aleatoria, puede que algunas actividades no hayan sido seleccionadas como sucesoras, por lo que estas actividades van a ser sucesoras de la actividad ficticia inicial.
- También se tienen las actividades que terminan en el tiempo del costo óptimo, estas actividades van a tener como única actividad sucesora a la actividad ficticia final.
- Por último, las actividades que de acuerdo a los tiempos de término, no pueden tener tantas actividades sucesoras como lo indica el número aleatorio obtenido previamente, para éstas se calculan las posibles actividades sucesoras que pueden tener y se asigna aleatoriamente un subconjunto de estas como las actividades sucesoras.

Por ejemplo, en la Figura 5.3 se muestra una planeación de 20 actividades (sin contar las ficticias), de costo óptimo 23, con 5 recursos y usando 11 unidades en total de recursos y en la Figura 5.4 se muestra la gráfica del proyecto a la que pertenece la planeación mostrada anteriormente. Se puede notar que gracias a la asignación de recursos y duración de las actividades, no existe algún espacio en la planeación y por tanto, la planeación construida es la óptima.

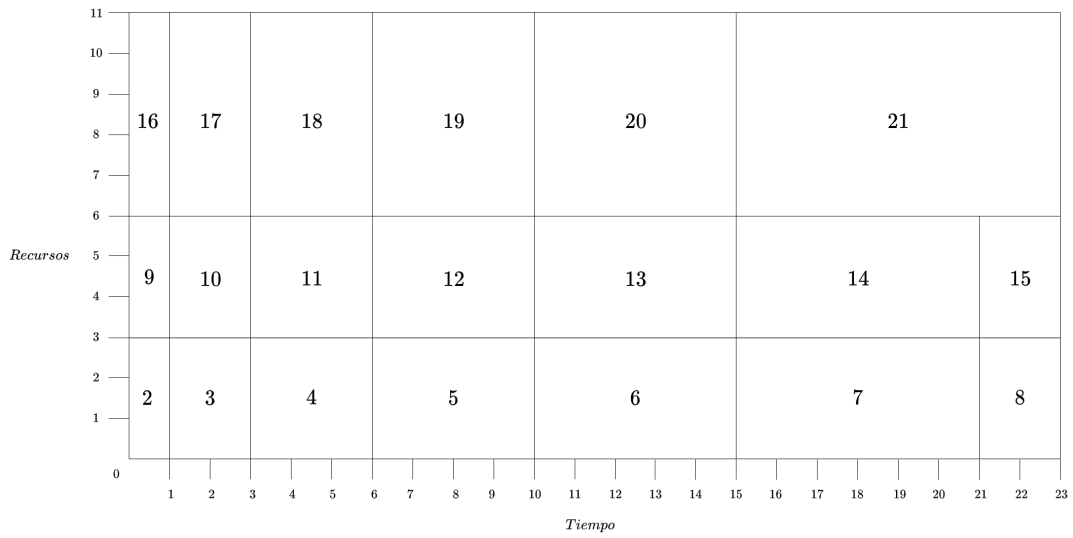


Figura 5.3: Planeación óptima con costo de 23

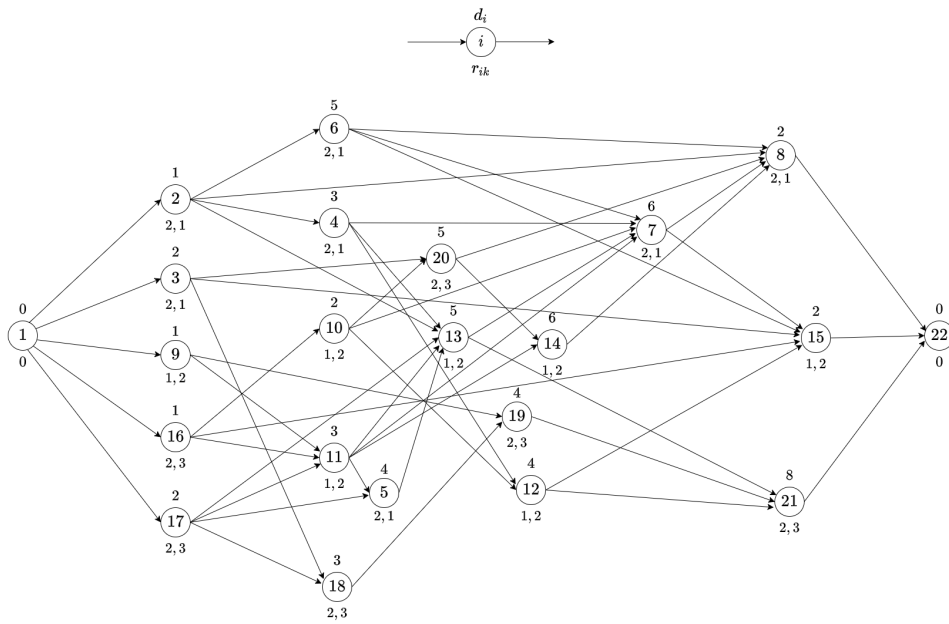


Figura 5.4: Gráfica del proyecto



## 5.2. Resultados

En el Cuadro 5.1 se presentan los resultados y los parámetros de usar Búsqueda Tabú para distintos ejemplares, de la misma manera en el Cuadro 5.2 usando Recocido Simulado. En ambas metaheurísticas se utilizan los mismos ejemplares.

		Búsqueda Tabú									
n	OP	SE	m	s(n)	TT	VE	IT	TM	IN	RE	
10	30	64	9	43	100	45	570	00:03:35	70	<b>31</b>	
20	40	60	2	54	39	18	300	00:01:42	70	<b>44</b>	
40	50	9	1	78	10	42	360	00:05:48	70	<b>67</b>	
50	60	6	3	302	40	25	550	00:57:20	70	<b>94</b>	
60	70	60	1	117	14	15	550	00:05:44	100	<b>118</b>	
70	80	56	1	136	40	35	450	00:17:47	100	<b>145</b>	
80	90	55	1	155	29	19	350	00:10:38	100	<b>176</b>	

Cuadro 5.1: Resultados usando Búsqueda Tabú

		Recocido Simulado									
n	OP	SE	m	s(n)	ITE	TE	DE	EP	TM	IN	RE
10	30	56	6	40	5	2500	0.95	0.01	00:00:15	70	<b>38</b>
20	40	45	4	76	34	2400	0.95	0.1	00:02:31	70	<b>41</b>
40	50	23	2	110	40	2000	0.95	0.001	00:08:30	70	<b>58</b>
50	60	69	2	137	22	2500	0.95	0.001	00:08:41	70	<b>111</b>
60	70	41	1	117	19	2400	0.94	0.01	00:03:50	100	<b>105</b>
70	80	21	1	136	32	2400	0.95	0.01	00:11:50	100	<b>107</b>
80	90	63	1	155	29	2500	0.95	0.01	00:19:39	100	<b>126</b>

Cuadro 5.2: Resultados usando Recocido Simulado

Abreviaturas:

**n**: Número de actividades (Sin contar actividades ficticias).

**OP**: Costo de la planeación óptima del ejemplar generado de manera aleatoria.

**SE**: Semilla.

**m**: Número de interrupciones.

**s(n)**: Número de subactividades.

**TT**: Tiempo tabú.

**VE**: Vecinos a explorar.

**IT**: Iteraciones.

**TM**: Tiempo que tomó la ejecución de la metaheurística (hh:mm:ss).

**ITE**: Iteraciones por nivel de temperatura.

**TE**: Temperatura.

**DE**: Decremento.

**EP**: Epsilon.

**IN**: Número de veces que se realizó la metaheurística.

**RE**: Costo de la planeación obtenida.

A continuación se muestran gráficamente algunos resultados.

### 5.2.1. Ejemplar de 10 actividades

Para el ejemplar de 10 actividades se muestra en la Figura 5.5 la planeación óptima con un costo de 30 unidades de tiempo.

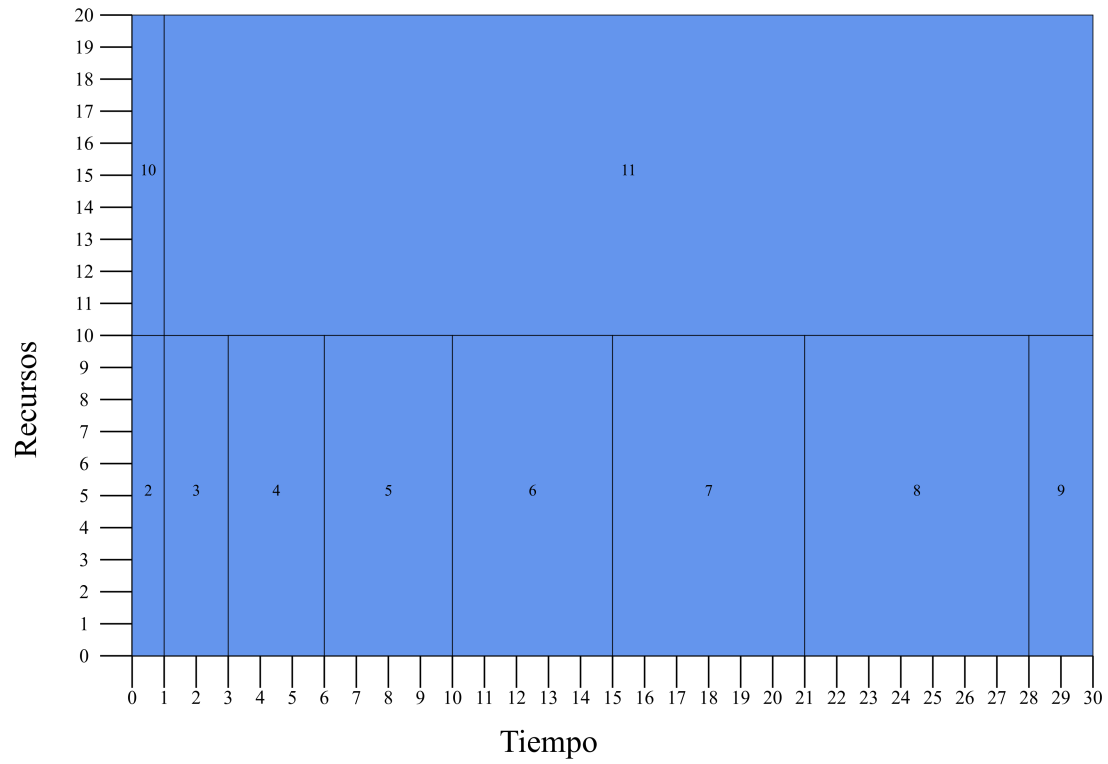


Figura 5.5: Planeación óptima para 10 actividades

## Mejor planeación encontrada con Recocido Simulado

Para la planeación mostrada en la Figura 5.5 se tiene en la Figura 5.6 la mejor planeación encontrada con Recocido Simulado con un costo de 38 unidades de tiempo, interrumpiendo a lo más 6 veces cada actividad, con lo cual, se planean 40 subactividades.

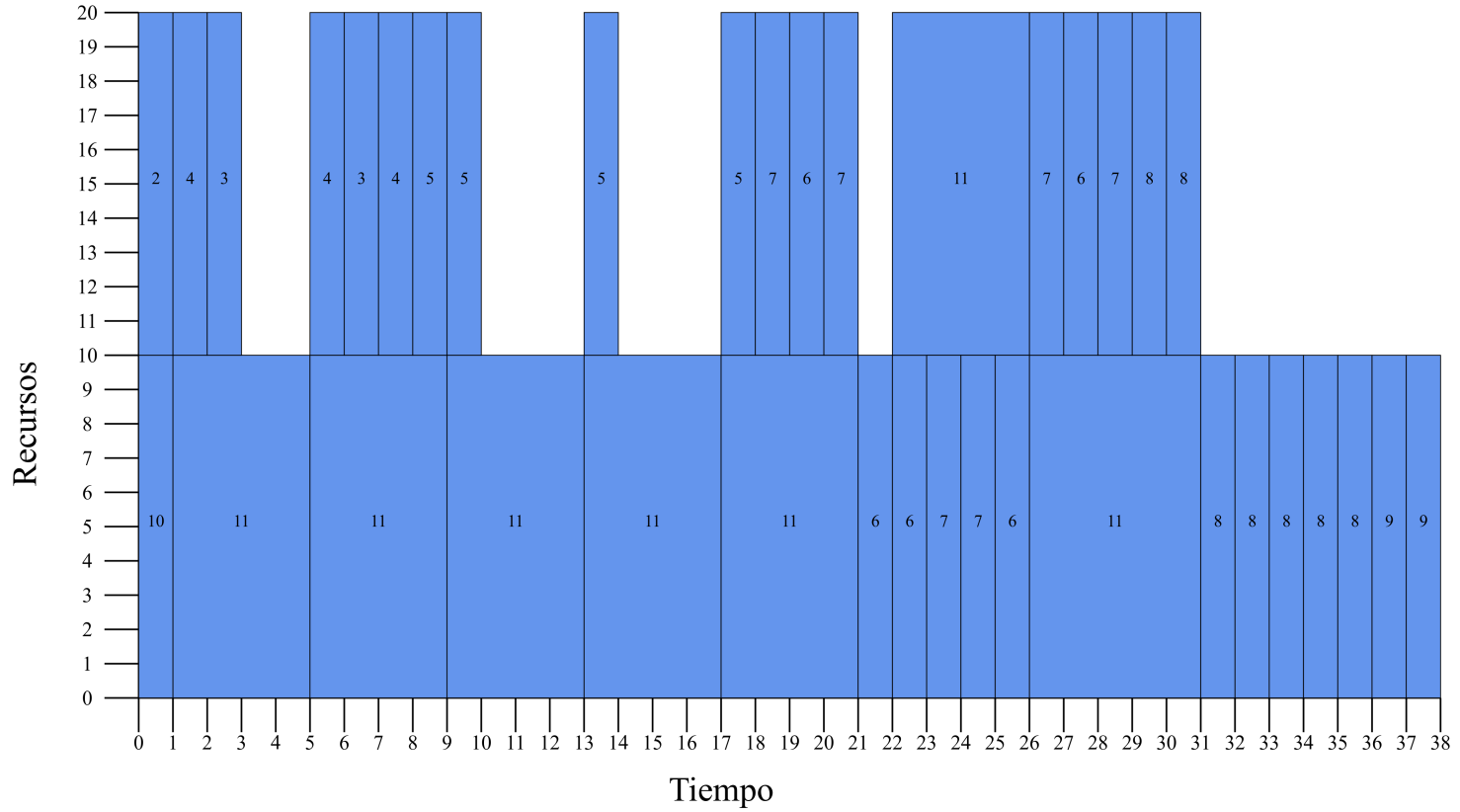


Figura 5.6: Mejor planeación encontrada con Recocido Simulado para 10 actividades

En la Figura 5.7 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.6.

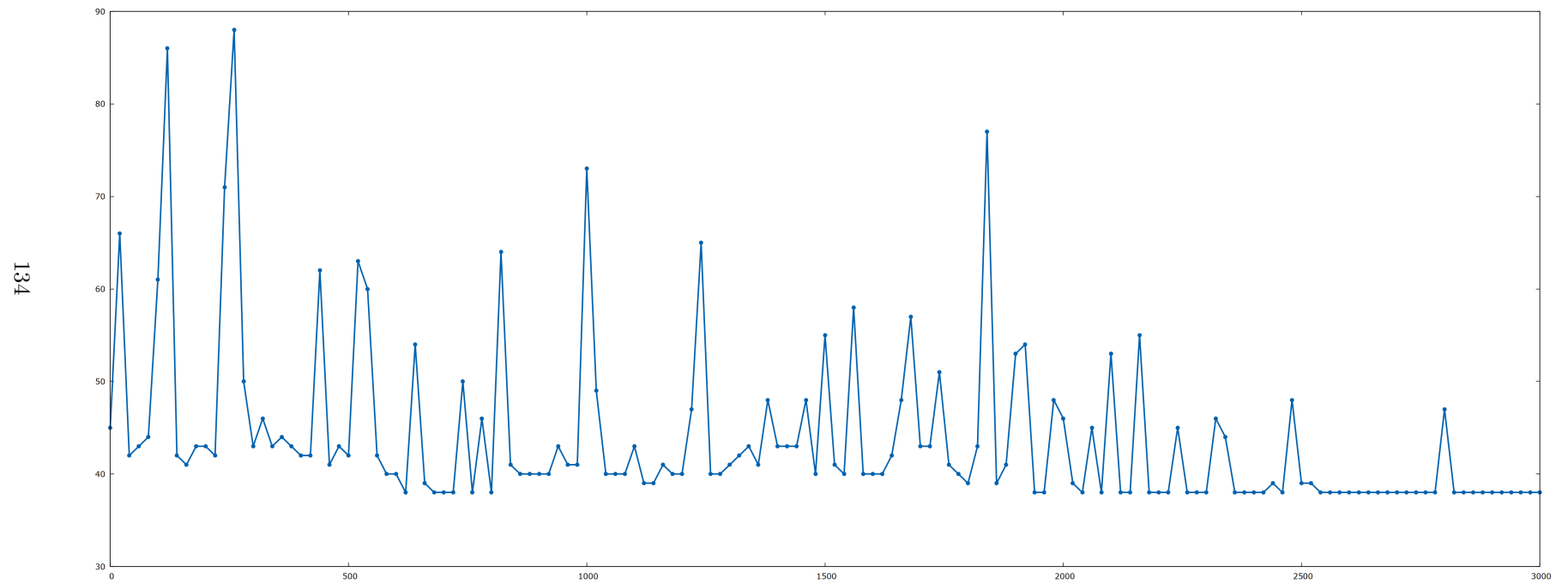


Figura 5.7: Soluciones visitadas con Recocido Simulado para 10 actividades

## Mejor planeación encontrada con Búsqueda Tabú

Para la planeación mostrada en la Figura 5.5 se tiene en la Figura 5.8 la mejor planeación encontrada con Búsqueda Tabú con un costo de 31 unidades de tiempo, interrumpiendo a lo más 9 veces cada actividad, con lo cual, se planean 43 subactividades.

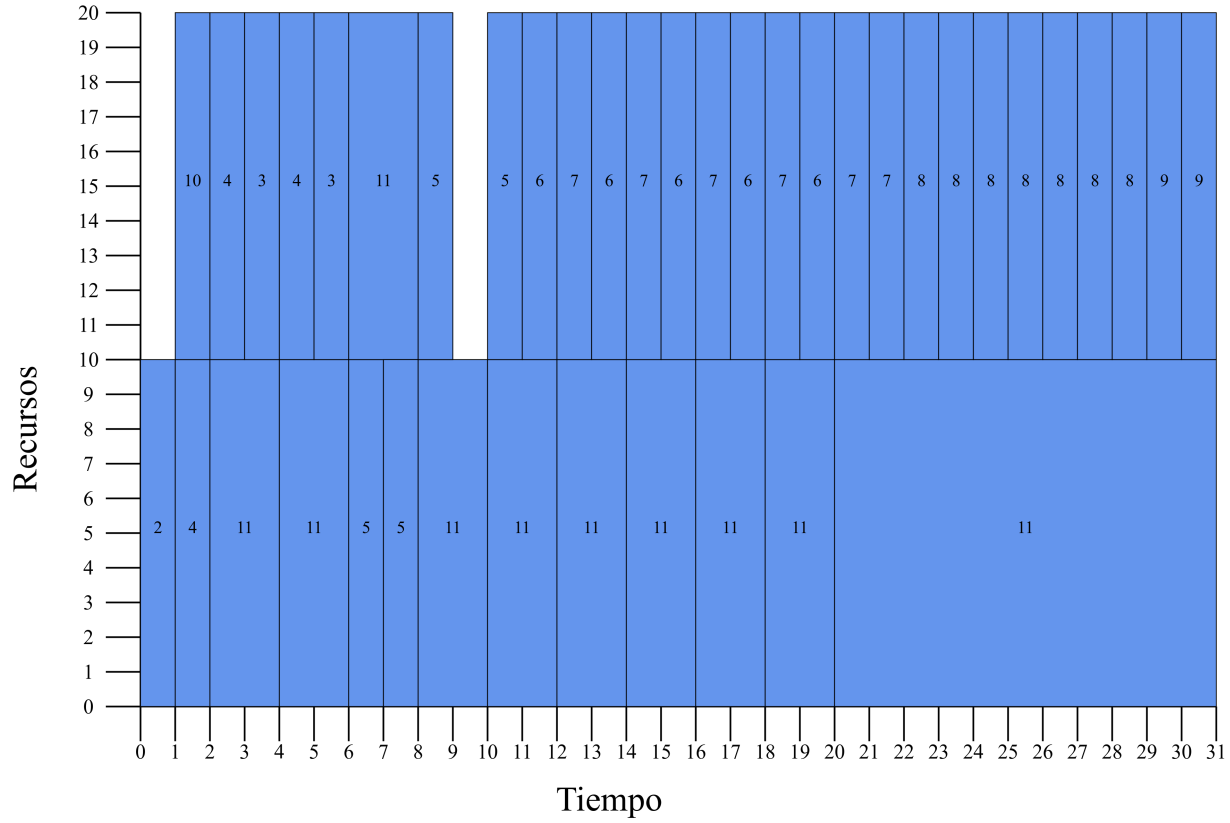


Figura 5.8: Mejor planeación encontrada con Búsqueda Tabú para 10 actividades

En la Figura 5.9 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.8.

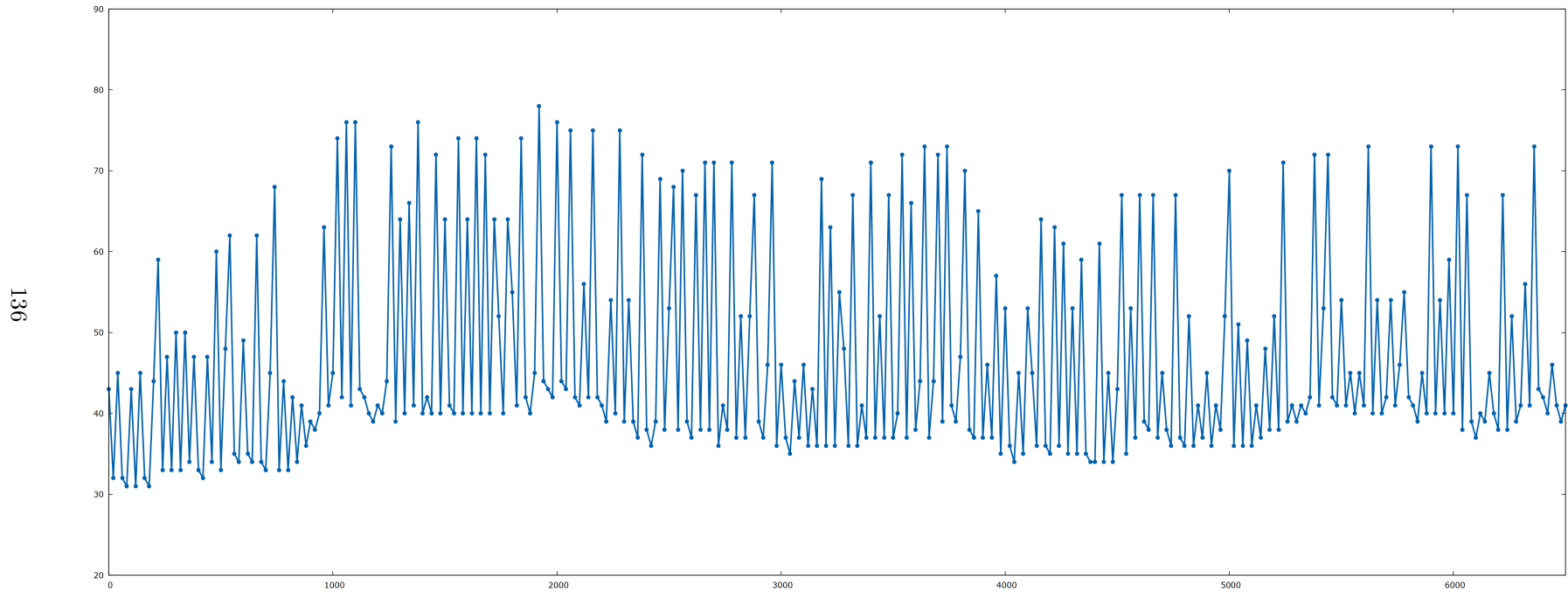


Figura 5.9: Soluciones visitadas con Búsqueda Tabú para 10 actividades

### 5.2.2. Ejemplar de 20 actividades

Para el ejemplar de 20 actividades se muestra en la Figura 5.10 la planeación óptima con un costo de 40 unidades de tiempo.

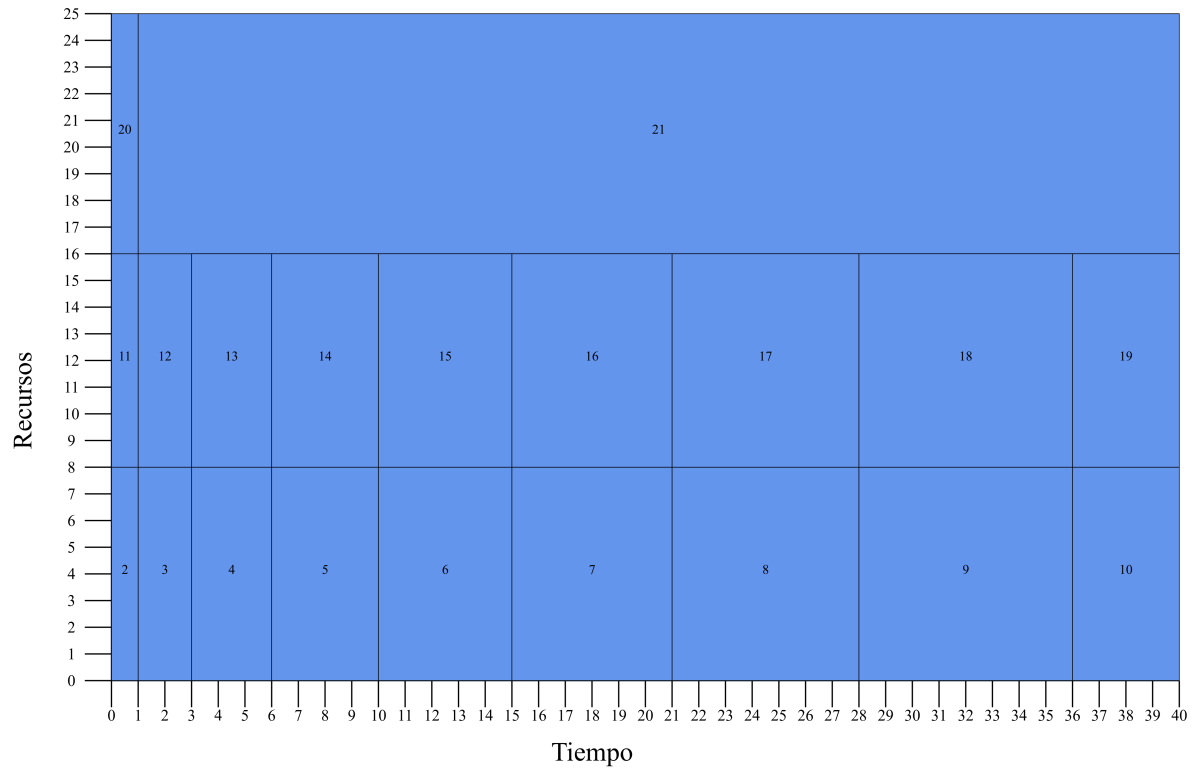


Figura 5.10: Planeación óptima para 20 actividades

### Mejor planeación encontrada con Recocido Simulado

Para la planeación mostrada en la Figura 5.10 se tiene en la Figura 5.11 la mejor planeación encontrada con Recocido Simulado con un costo de 41 unidades de tiempo, interrumpiendo a lo más 4 veces cada actividad, con lo cual, se planean 76 subactividades.

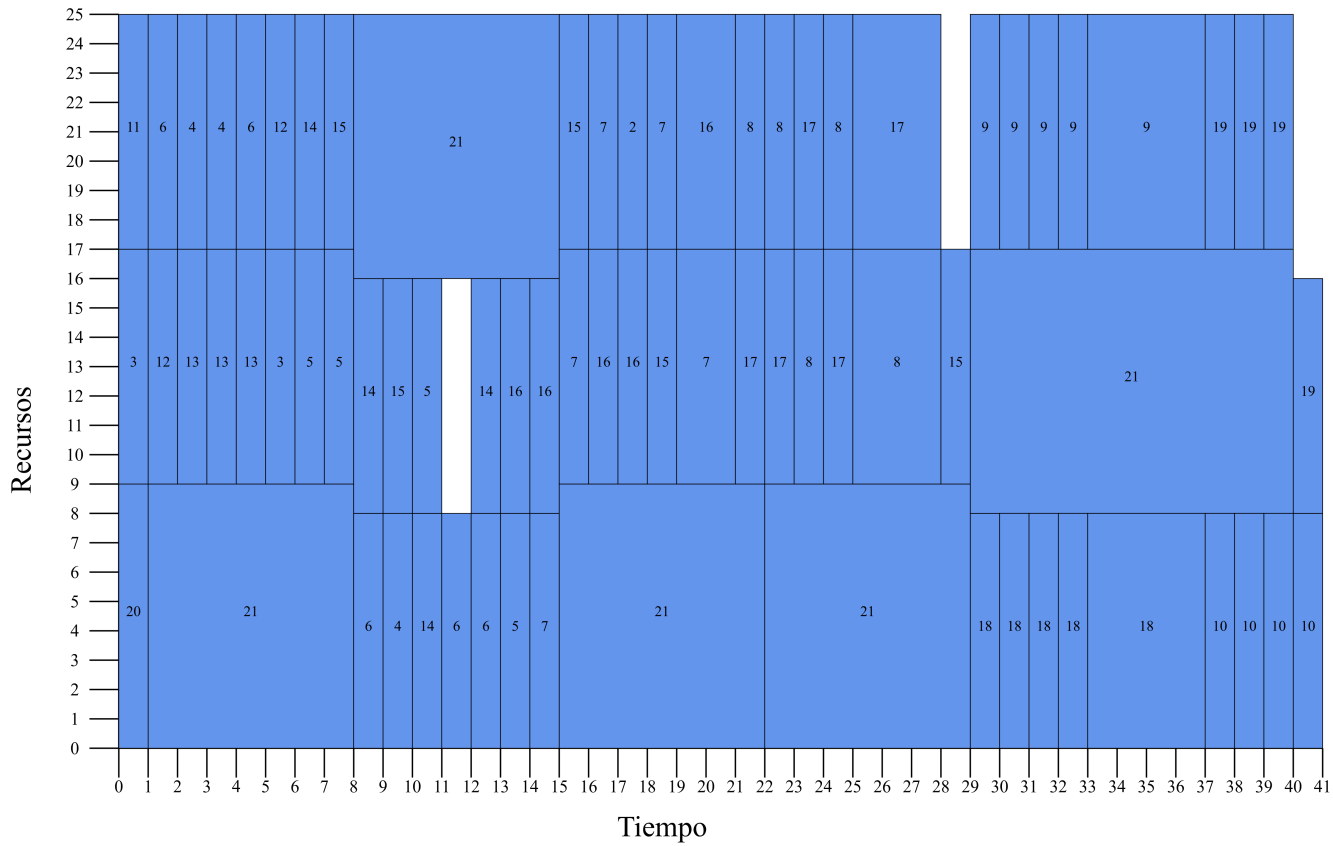


Figura 5.11: Mejor planeación encontrada con Recocido Simulado para 20 actividades



En la Figura 5.12 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.11.

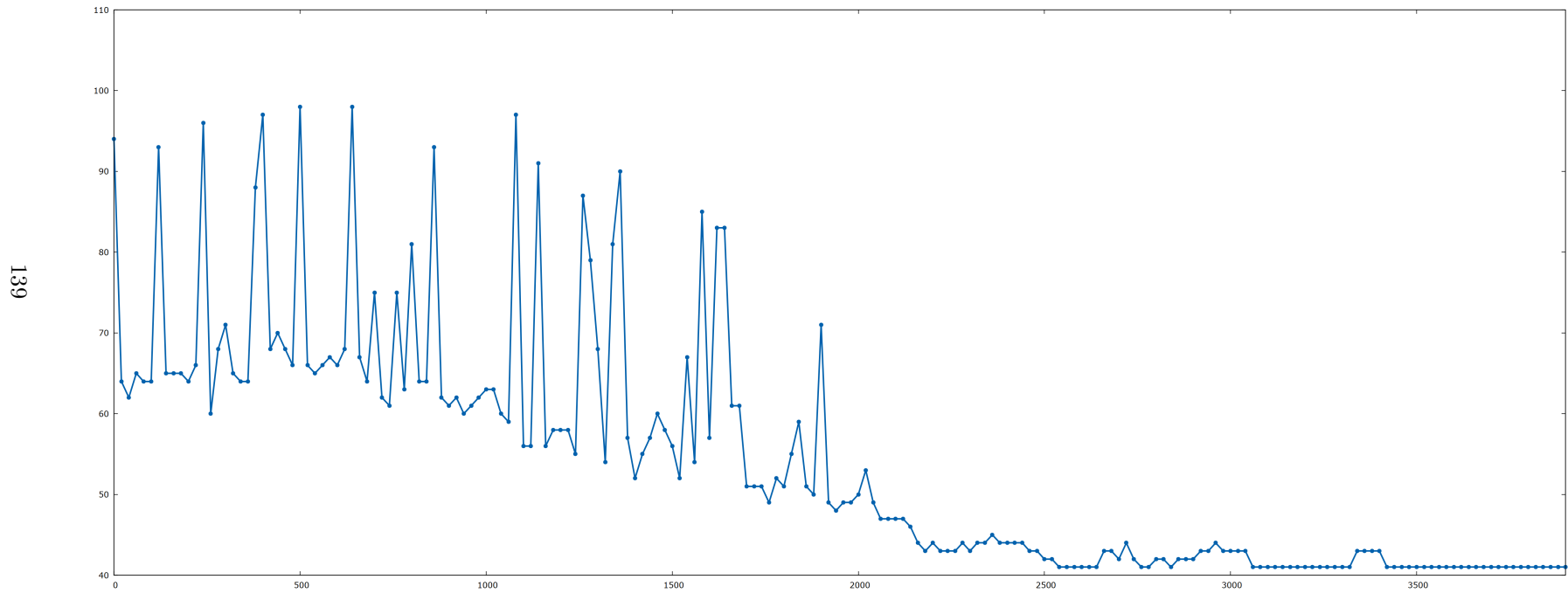


Figura 5.12: Soluciones visitadas con Recocido Simulado para 20 actividades

### Mejor planeación encontrada con Búsqueda Tabú

Para la planeación mostrada en la Figura 5.10 se tiene en la Figura 5.13 la mejor planeación encontrada con Búsqueda Tabú con un costo de 44 unidades de tiempo, interrumpiendo a lo más 2 veces cada actividad, con lo cual, se planean 54 subactividades.

140

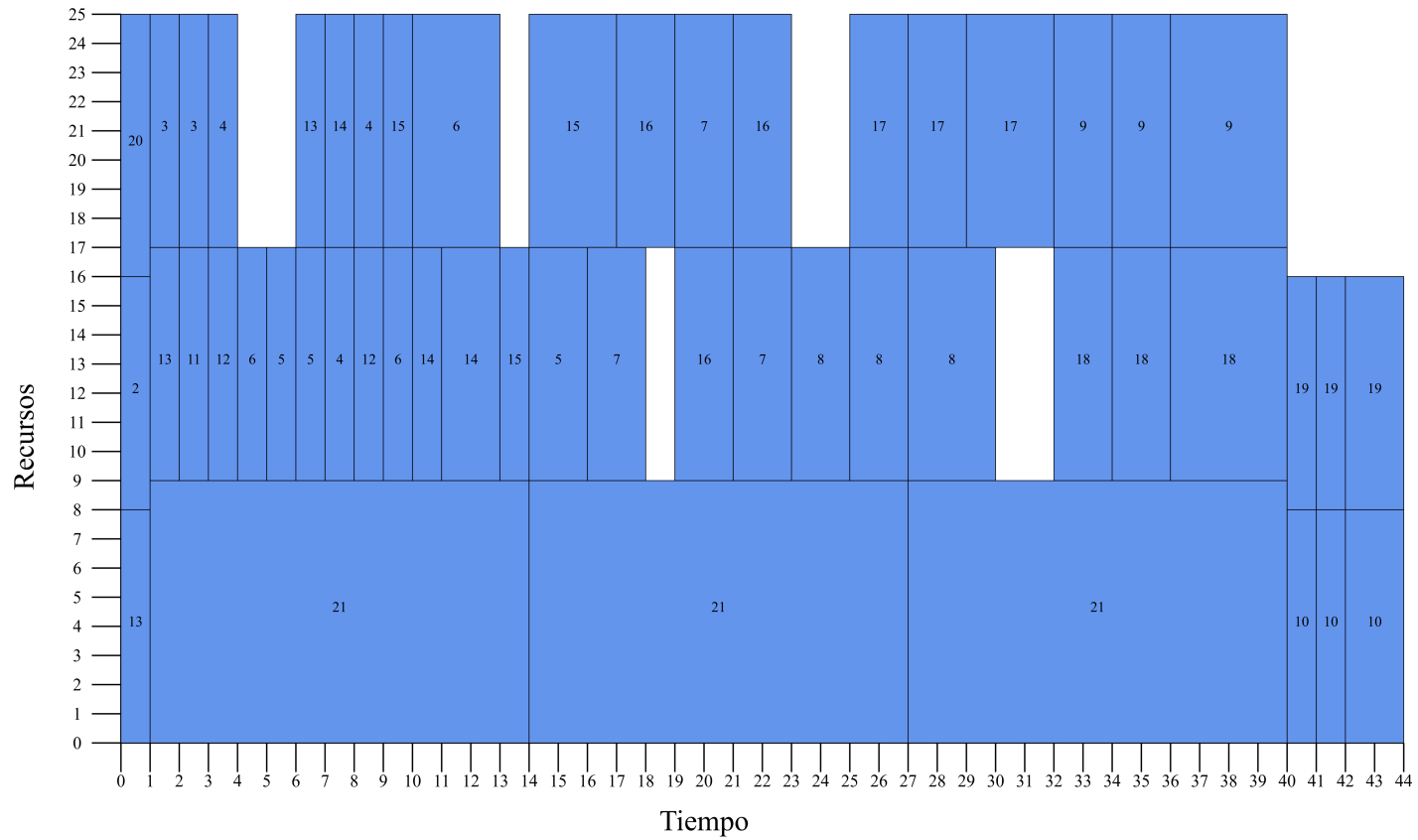


Figura 5.13: Mejor planeación encontrada con Búsqueda Tabú para 20 actividades

En la Figura 5.14 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.13.

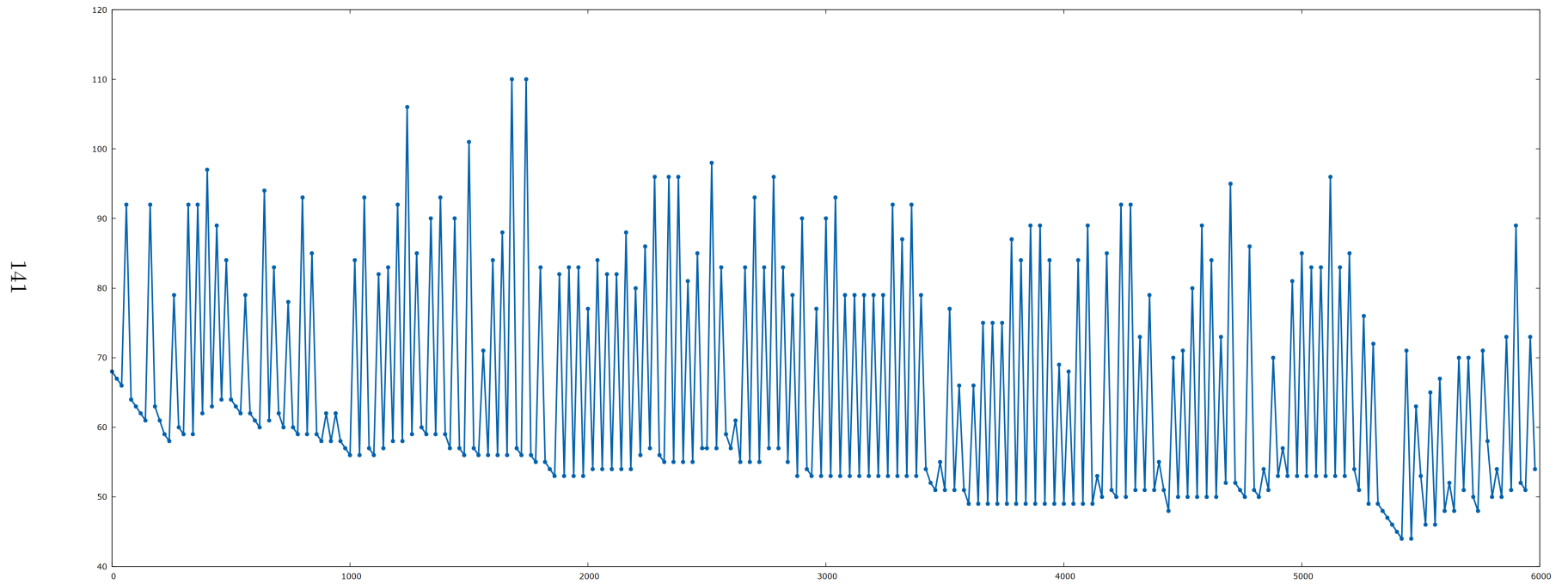


Figura 5.14: Soluciones visitadas con Búsqueda Tabú para 20 actividades

### 5.2.3. Ejemplar de 40 actividades

Para el ejemplar de 40 actividades se muestra en la Figura 5.15 la planeación óptima con un costo de 50 unidades de tiempo.

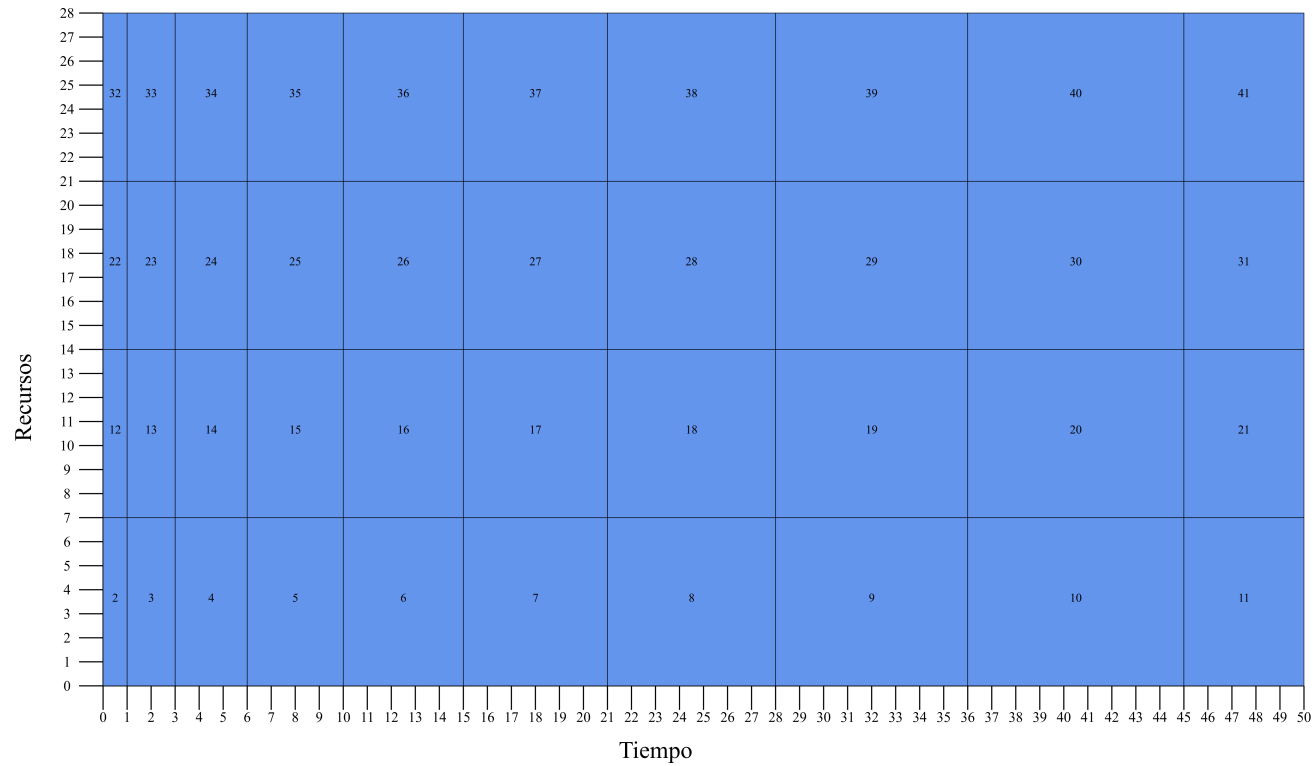


Figura 5.15: Planeación óptima para 40 actividades

### Mejor planeación encontrada con Recocido Simulado

Para la planeación mostrada en la Figura 5.15 se tiene en la Figura 5.16 la mejor planeación encontrada con Recocido Simulado con un costo de 58 unidades de tiempo, interrumpiendo a lo más 2 veces cada actividad, con lo cual, se planean 110 subactividades.

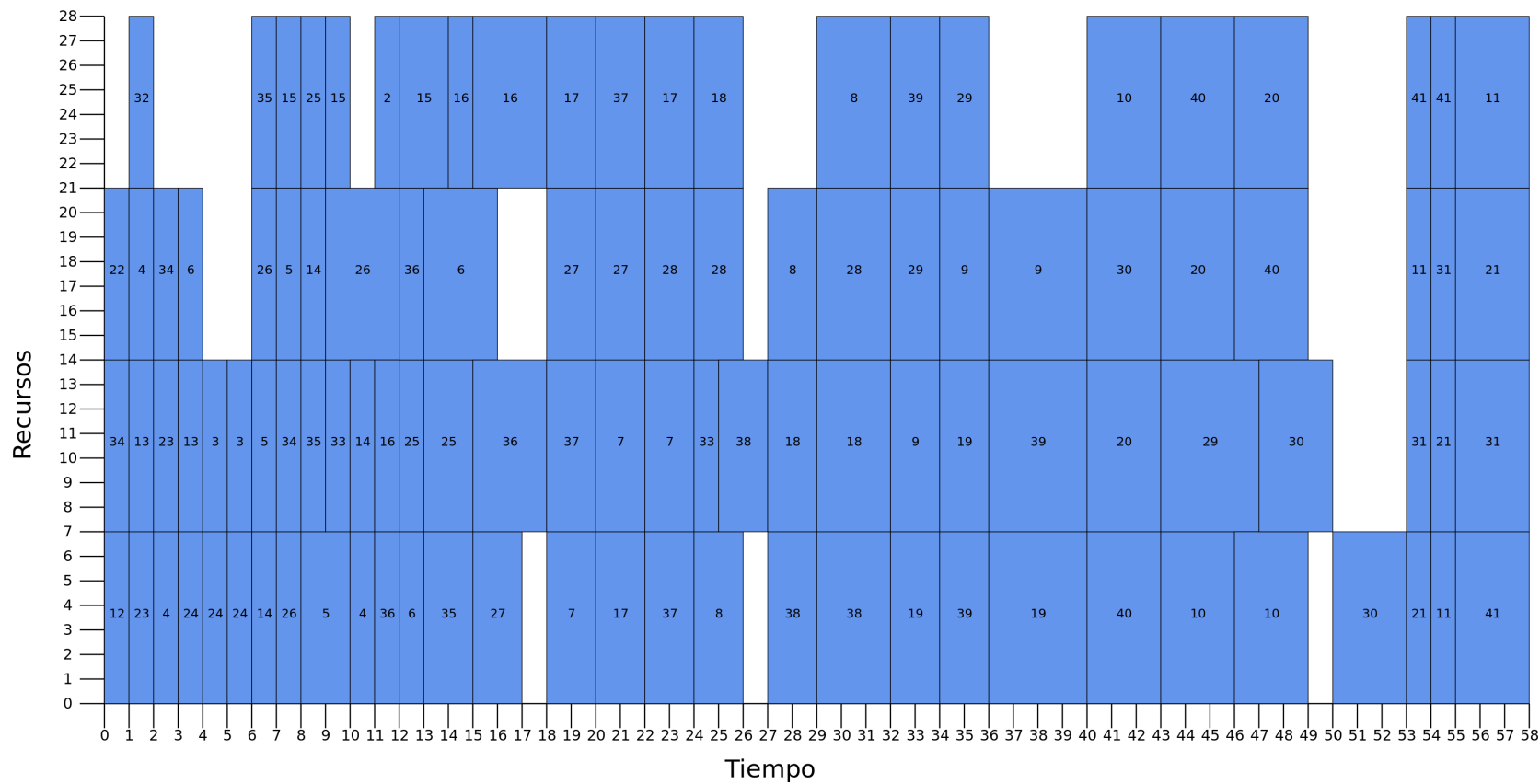


Figura 5.16: Mejor planeación encontrada con Recocido Simulado para 40 actividades

En la Figura 5.17 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.16.

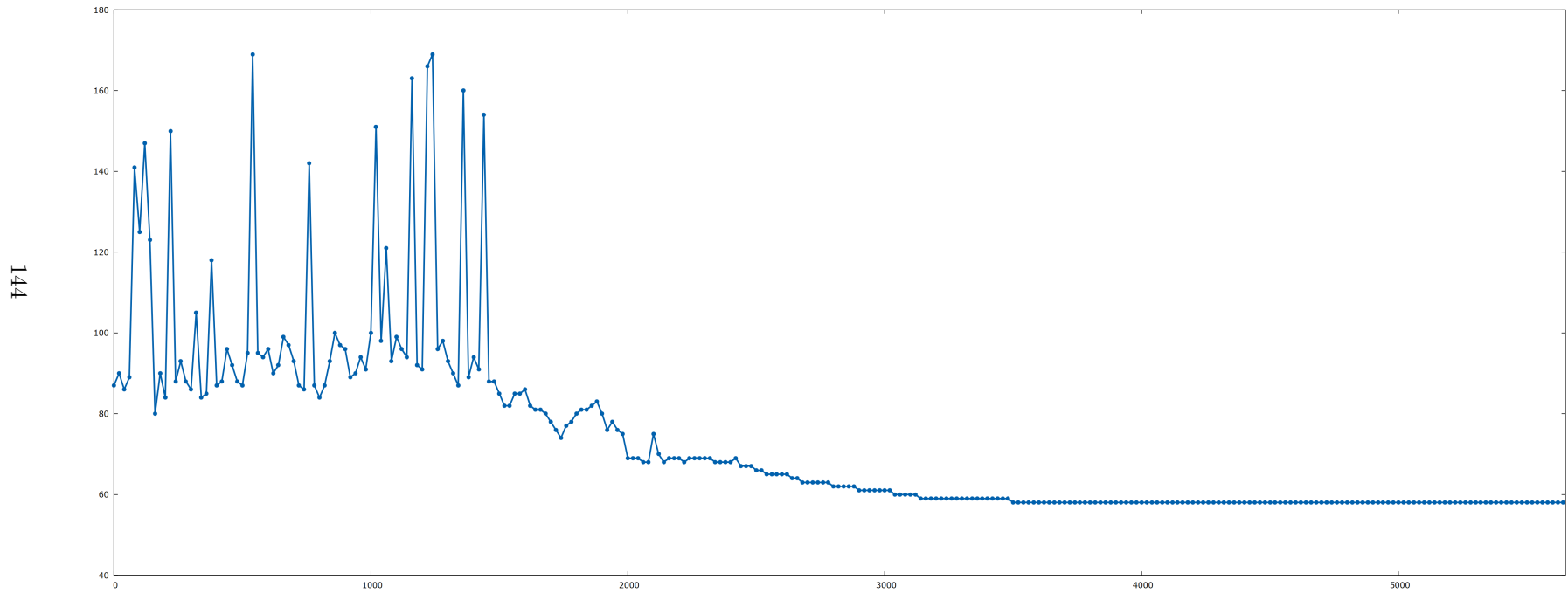


Figura 5.17: Soluciones visitadas con Recocido Simulado para 40 actividades

### Mejor planeación encontrada con Búsqueda Tabú

Para la planeación mostrada en la Figura 5.15 se tiene en la Figura 5.18 la mejor planeación encontrada con Búsqueda Tabú con un costo de 67 unidades de tiempo, interrumpiendo a lo más 1 vez cada actividad, con lo cual, se planean 78 subactividades.

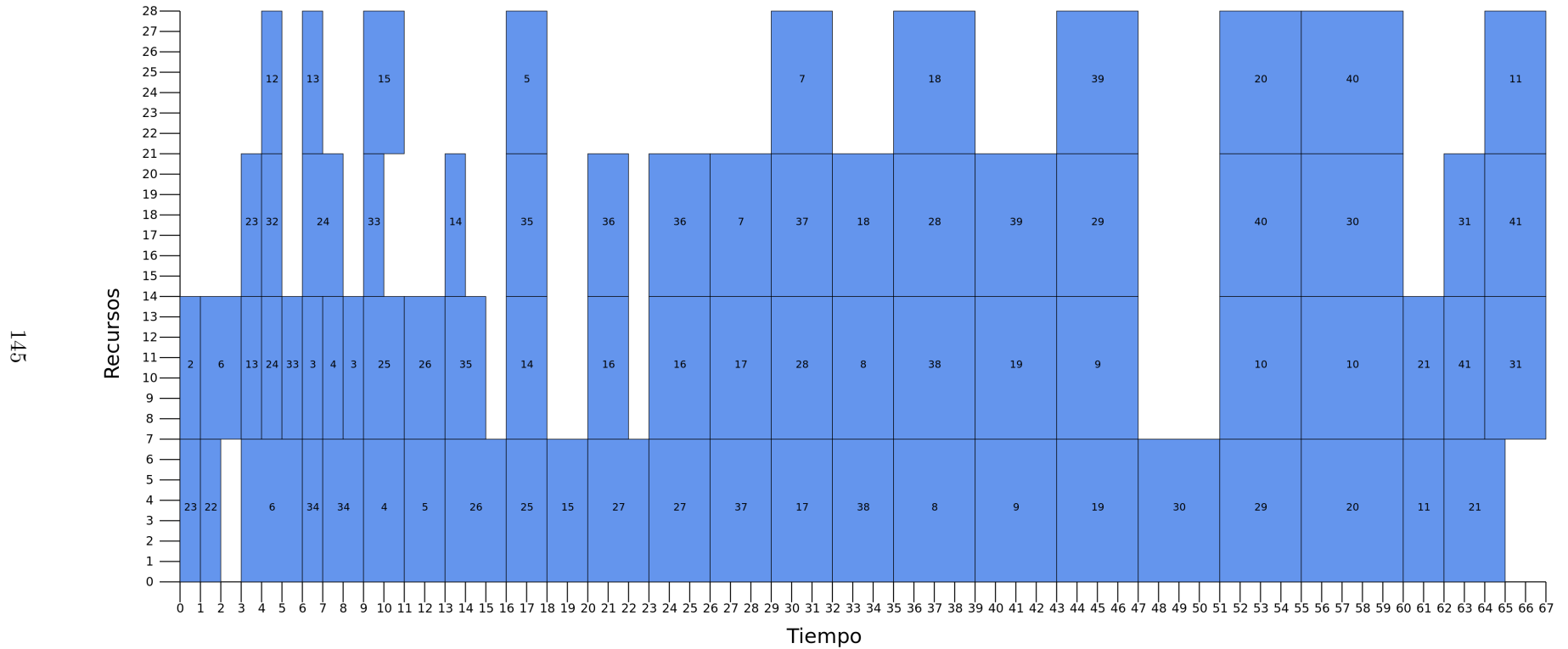


Figura 5.18: Mejor planeación encontrada con Búsqueda Tabú para 40 actividades

En la Figura 5.19 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.18.

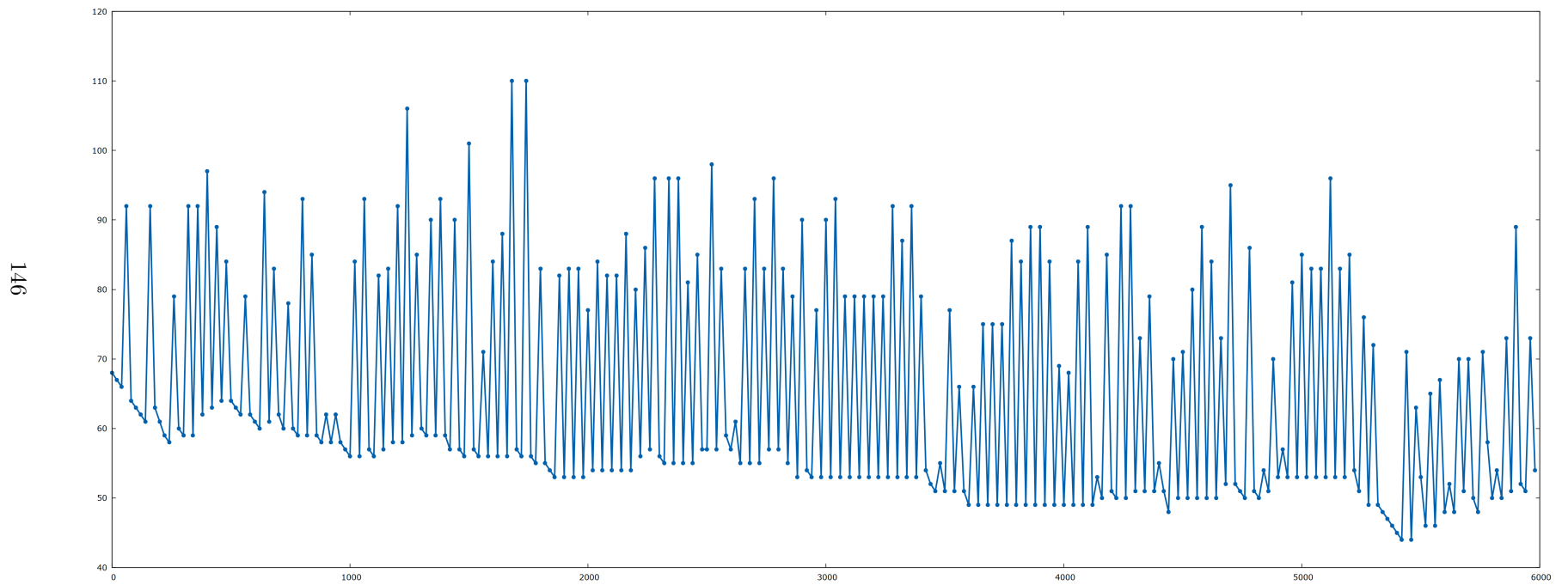


Figura 5.19: Soluciones visitadas con Búsqueda Tabú para 40 actividades



### 5.2.4. Ejemplar de 80 actividades

Para el ejemplar de 80 actividades se muestra en la Figura 5.20 la planeación óptima con un costo de 90 unidades de tiempo.

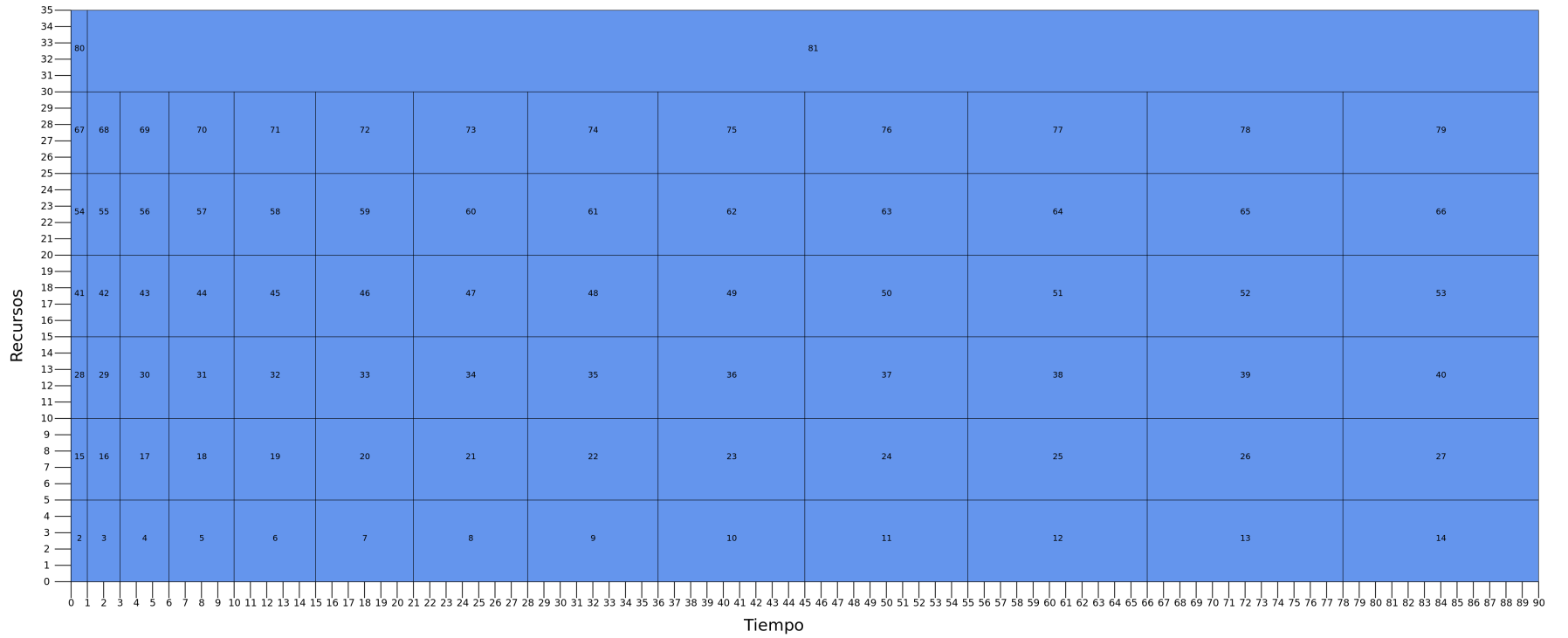


Figura 5.20: Planeación óptima para 80 actividades

### Mejor planeación encontrada con Recocido Simulado

Para la planeación mostrada en la Figura 5.20 se tiene en la Figura 5.21 la mejor planeación encontrada con Recocido Simulado con un costo de 126 unidades de tiempo, interrumpiendo a lo más 1 vez cada actividad, con lo cual, se planean 155 subactividades.

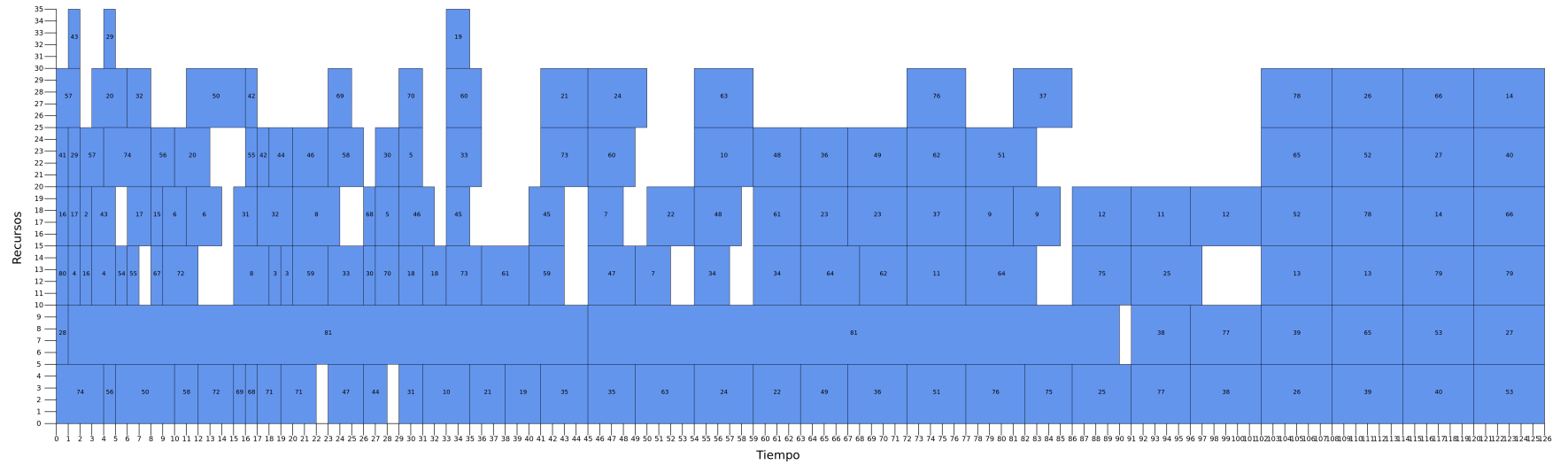


Figura 5.21: Mejor planeación encontrada con Recocido Simulado para 80 actividades

En la Figura 5.22 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en la Figura 5.21.

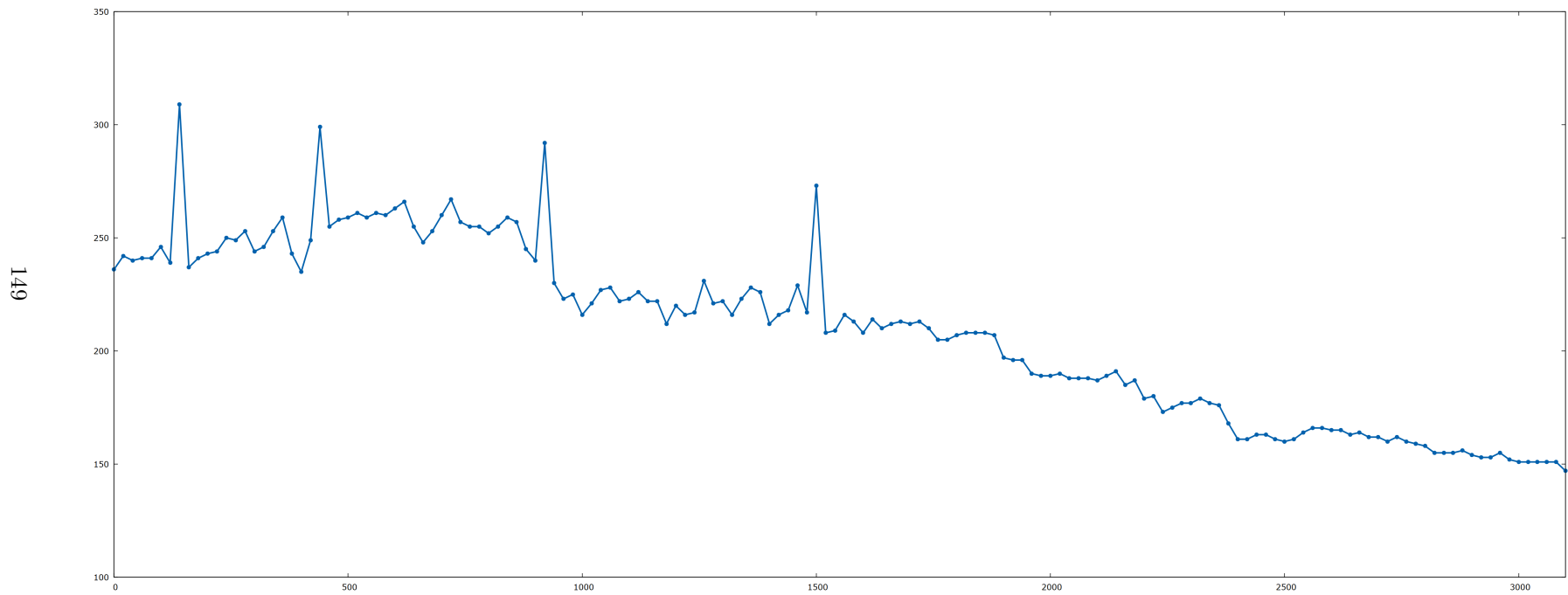


Figura 5.22: Soluciones visitadas con Recocido Simulado para 80 actividades

### Mejor planeación encontrada con Búsqueda Tabú

Para la planeación mostrada en la Figura 5.20, debido a la longitud de la planeación, se muestra en las Figuras 5.23 y 5.24 la primera mitad y segunda mitad respectivamente de la mejor planeación encontrada con Búsqueda Tabú con un costo de 176 unidades de tiempo, interrumpiendo a lo más 1 vez cada actividad, con lo cual, se planean 155 subactividades.

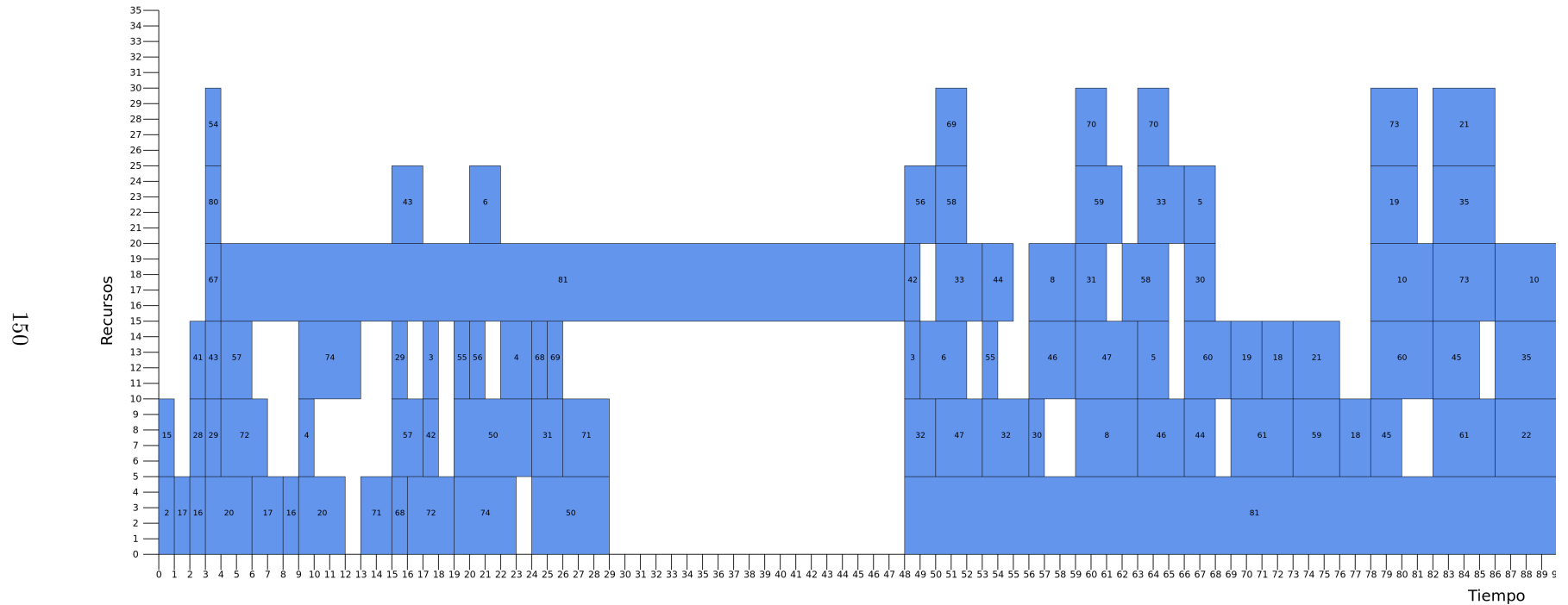


Figura 5.23: Mejor planeación encontrada con Búsqueda Tabú para 80 actividades

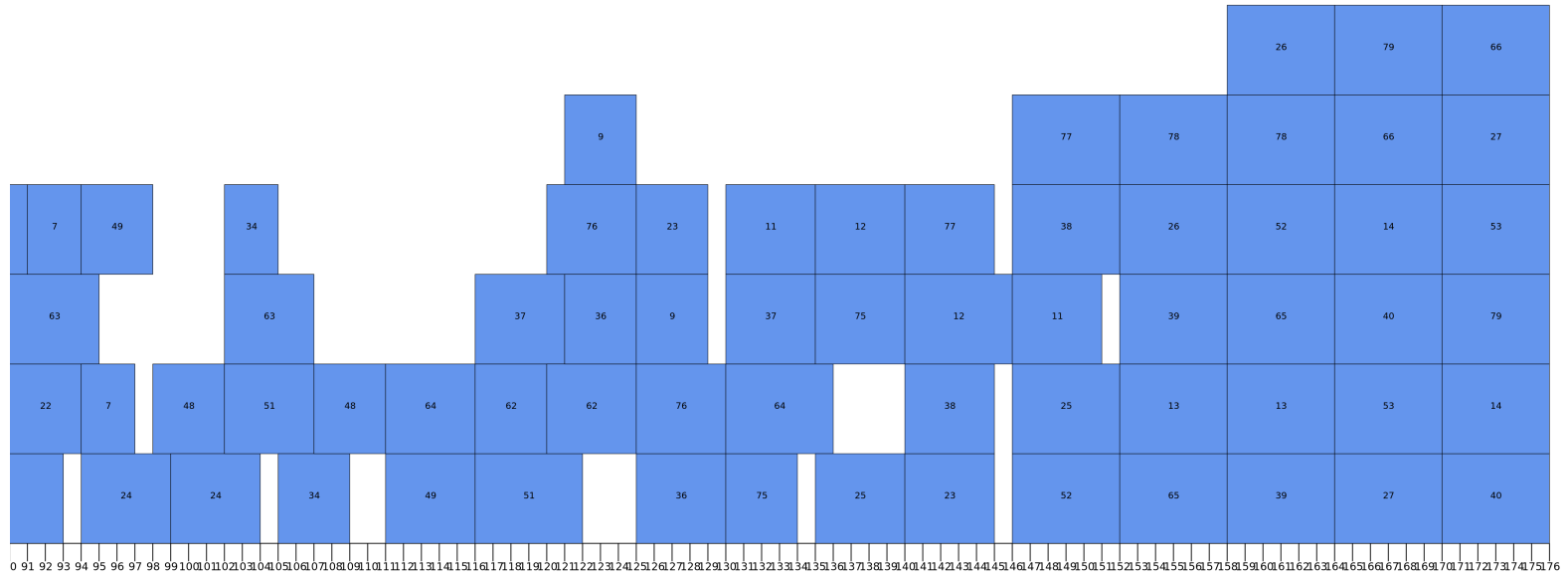


Figura 5.24: Mejor planeación encontrada con Búsqueda Tabú para 80 actividades

En la Figura 5.25 se muestra el costo de las soluciones visitadas durante la ejecución de la metaheurística para la planeación mostrada en las Figuras 5.23 y 5.24.

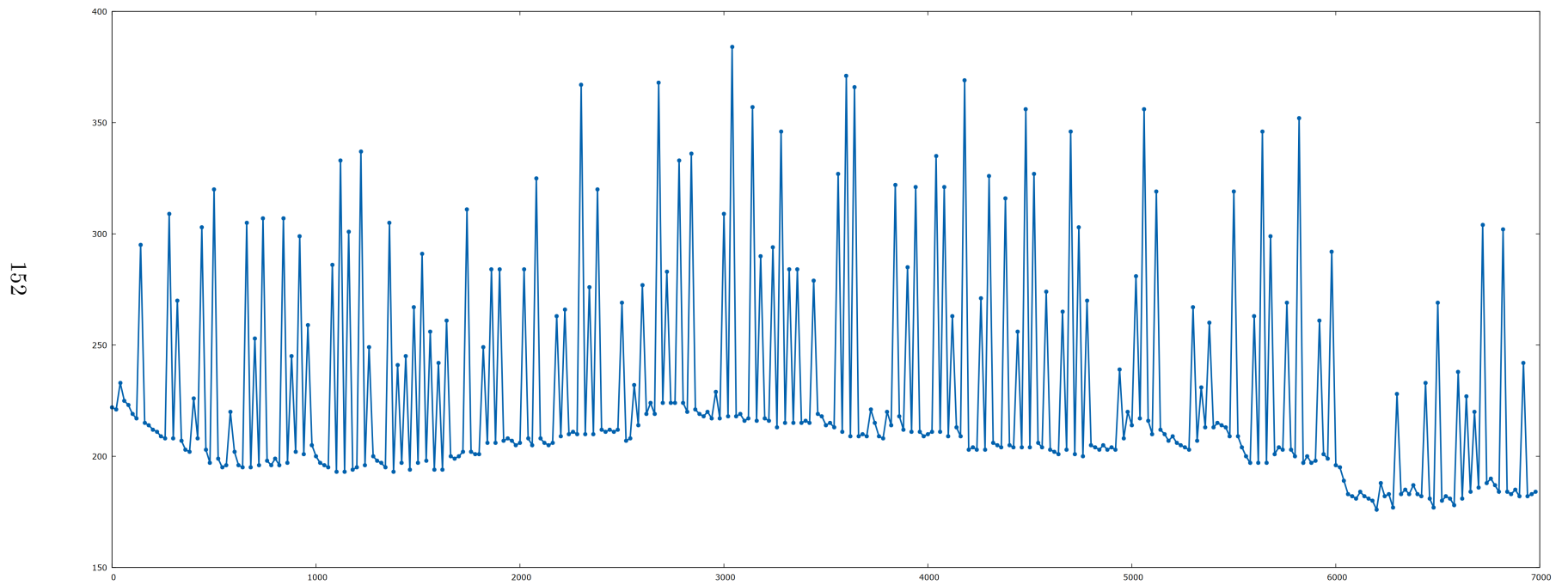


Figura 5.25: Soluciones visitadas con Búsqueda Tabú para 80 actividades

### 5.3. Análisis

De acuerdo a los resultados obtenidos en todos los ejemplares, la metaheurística de Recocido Simulado logró mejores resultados que la metaheurística de Búsqueda Tabú.

Analizando el desempeño de ambas, para la Búsqueda Tabú, aunque en cada tiempo se toma una solución distinta, con el fin de poder diversificar la búsqueda y tener un comportamiento oscilatorio, fue más difícil encontrar los parámetros adecuados para obtener mejores soluciones y esto es porque al seguir utilizando la búsqueda local aún con el manejo de memoria, salir de una región con soluciones pobres toma bastante tiempo, por lo que, para encontrar soluciones atractivas, se deben realizar más iteraciones.

Por el otro lado, en el Recocido Simulado, se encontraron mejores soluciones por la ventaja de usar más aleatoriedad que la Búsqueda Tabú; esto es, poder elegir peores soluciones basándose en una probabilidad causa que se visiten bastantes regiones en una ejecución y que por lo tanto, al llegar a una región con soluciones pobres y aunque se tenga una temperatura baja, existe la probabilidad de moverse a otra región en menos tiempo, teniendo una búsqueda con mayor diversificación.

Por lo tanto, lo que marca la diferencia entre los resultados de ambas metaheurísticas es la capacidad de cambiar entre regiones del espacio de búsqueda, esto gracias a la mayor aleatoriedad usada en el Recocido Simulado. Es común que a la metaheurística de Búsqueda Tabú se le refiera como una metaheurística determinista por el hecho de basar sus movimientos en decisiones que no involucran probabilidad, lo que hace es tomar las decisiones con base en la solución actual, el estado de la memoria y el vecindario actual.

Un factor importante que mejora el tiempo que tardan ambas metaheurísticas es la cantidad de interrupciones, como se muestra en el Cuadro 5.3, si se interrumpen las actividades en subactividades de duración uno, el tamaño de los ejemplares sobre los cuales se tiene que realizar la búsqueda crece de manera significativa.

n	s(n)
10	60
20	120
40	200
50	300
60	350
70	480
80	630

Cuadro 5.3: Número de subactividades de duración uno por ejemplar

Otra forma de ver cómo crece el número de actividades en los ejemplares usando interrupciones se muestra en la Figura 5.26, al realizar cada vez una interrupción de más, la cantidad de actividades se incrementa bastante.

Aunque el Recocido Simulado obtiene mejores resultados para el problema de interés, no significa que sea una mejor metaheurística en todos los problemas a los que se pueda aplicar que la Búsqueda Tabú, simplemente, los parámetros encontrados en la práctica funcionan mejor para el Recocido Simulado y esto se debe a que tiene una mayor libertad de implementación; es decir, los parámetros no

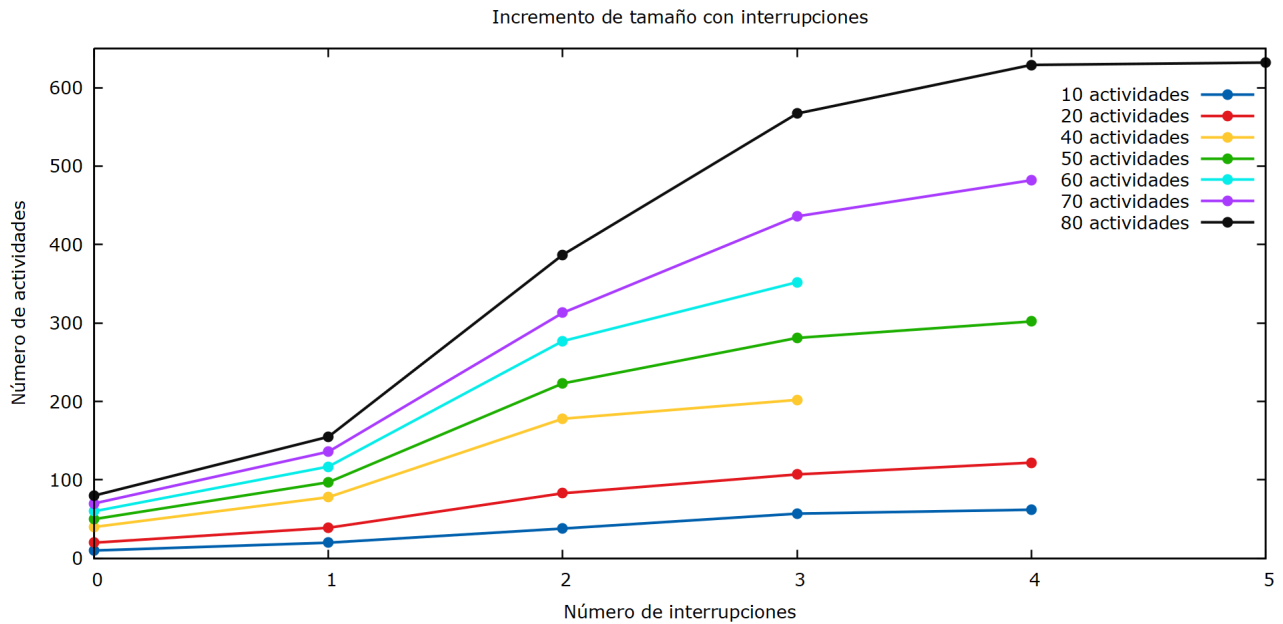


Figura 5.26: Incremento de actividades de acuerdo al número de interrupciones

necesitan estar tan definidos como los de la Búsqueda Tabú. Por ejemplo, en la Búsqueda Tabú, se debe definir específicamente cual es la estrategia para diversificar la búsqueda, en cambio, para el Recocido Simulado, basta con manejar el parámetro de la temperatura, con la cual se diversifica la búsqueda. En cuanto a las estrategias de intensificación, para el Recocido Simulado es más fácil, porque, con el manejo de la temperatura cuando es baja y la cantidad de soluciones vecinas a explorar es suficiente para enfocar la búsqueda en alguna región, mientras que en la Búsqueda Tabú, para intensificar la búsqueda es necesario definir el tiempo tabú y lo más complicado, cómo obtener conocimiento de la memoria y cómo aplicarlo correctamente al problema.



# Conclusiones

Ante los problemas de optimización que son NP-Difíciles, se han estudiado y diseñado estrategias de aproximación como las metaheurísticas, que inspiradas en comportamientos de la naturaleza, físicos y sociales, logran buscar soluciones a estos problemas utilizando un nivel alto de abstracción, es decir, los detalles específicos de; cómo representar la solución, las soluciones iniciales, la búsqueda en el espacio de soluciones y las estrategias de diversificación e intensificación, deben ser diseñados por el experto en el problema, pero algo aún más importante de obtener, son los parámetros correctos, lo cual es una tarea complicada y que depende del comportamiento de la metaheurística en la práctica.

En cuanto a las metaheurísticas de Recocido Simulado y Búsqueda Tabú, comparten el uso de la búsqueda local, una utilizando la probabilidad para la toma de decisiones y otra basándose en el conocimiento adquirido por la memoria, aunque ambas son completamente distintas, fue interesante compararlas por la razón de comparar una estrategia totalmente probabilista a una estrategia determinista y por tanto, comparar si puede ser mejor tomar una solución peor con alguna probabilidad o tomar sólo soluciones basándose en el conocimiento adquirido en la búsqueda.

Para el PRCPSP se logró implementar el Recocido Simulado y la Búsqueda Tabú, además, se realizó una forma de crear ejemplares de manera aleatoria pero controlada, con el objetivo de poder usarlos en las metaheurísticas, pues de éstos se conoce la planeación óptima. Como se ve en la comparación, el Recocido Simulado obtuvo mejor desempeño gracias al mayor uso de aleatoriedad en la elección de soluciones, con la cual, se logró tener mayor diversidad en soluciones, pues es más sencillo dar saltos hacia otras regiones del espacio de soluciones, que a comparación de la Búsqueda Tabú, una vez que entraba en alguna región, salir de ésta toma mayor tiempo. Aún con los resultados satisfactorios para el Recocido Simulado, ninguna metaheurística funciona en general mejor que otra, sólo que encontrar los parámetros adecuados en ambas es complicado y como se nota en los resultados, los parámetros para el Recocido Simulado otorgan mayor libertad a la búsqueda y por tanto, recorrer el espacio de soluciones es más sencillo.

Aunque como se menciona en la investigación, pese a las ventajas de realizar interrupciones, el costo que tiene realizar esto afecta el tamaño de los ejemplares, como se ve en la comparación de las metaheurísticas, realizar bastantes interrupciones a las actividades de un ejemplar provoca realizar una búsqueda sobre ejemplares que tienen un tamaño considerablemente mayor al original, sin embargo, la ventaja principal es que las actividades al ser más pequeñas, es posible tener planeaciones más compactas, donde los recursos se puedan utilizar una mayor cantidad de tiempo.

Finalmente, saber elegir, diseñar e implementar metaheurísticas para problemas tan difíciles como el PRCPSP que están presentes en la era moderna, es absolutamente relevante, ya que identificar bien el problema junto a una buena estrategia conduce a encontrar buenas soluciones al problema y por tanto, alcanzar los objetivos planteados del problema, por lo tanto, el estudio de metaheurísticas es fundamental para la optimización de problemas.

# Bibliografía

- [1] Blazewicz, J., Lenstra, J., Rinnooy, A. (1983). *Scheduling subject to resource constraints: classification and complexity*. Discrete Applied Mathematics 5, 11–24.
- [2] Demeuleester, E., Herroelen, W. (1996). *An efficient optimal procedure for the preemptive resource-constrained project scheduling problem*. European Journal of Operational Research 90, 334-348.
- [3] Ballestín, F., Valls, V., Quintanilla, S. (2008). *Pre-emption in resource-constrained project scheduling*. European Journal of Operational Research 189, 1136-1152.
- [4] French, S.(1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Ellis Hordwood.
- [5] Bouleimen, K., Lecocq, H. (2003). *A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version*. European Journal of Operational Research 149, 268-281.
- [6] Eglese, R. (1990). *Simulated annealing: A tool for operational research*. European Journal of Operational Research 46, 271-281.
- [7] Gendreau, M., Potvin, J. (2010). *Handbook of Metaheuristics*.(2a ed.). Boston, E.U.A: Springer.
- [8] Glover, F. (1990). *Tabu Search: A Tutorial*. Interfaces 20, 74-94.
- [9] Gonzalez, T. (2007). *Handbook of Approximation Algorithms and Metaheuristics*.(2a ed.). Santa Barbara, E.U.A: Chapman & Hall/CRC.
- [10] Brucker, P. Knust, S. (2012). *Complex Scheduling*.(2a ed.). Heidelberg, Berlin: Springer.
- [11] Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems*.(5a ed.). Heidelberg, Berlin: Springer.
- [12] Kolisch, R. Padman, R. (2001). *An integrated survey of deterministic project scheduling*. Omega 29, 249-272.
- [13] Brucker, P. (2007). *Scheduling Algorithms*.(5a ed.). Heidelberg, Berlin: Springer.
- [14] Glover, F. Laguna, M. Pardalos, P. Du, D. Graham, R.L. (2013). *Tabu search: effective strategies for hard problems in analytics and computational science*. Handbook of Combinatorial Optimization. (2a ed.). 3261-3362.
- [15] Brucker, P. (2002). *Scheduling and constraint propagation*. Discrete Applied Mathematics 123,

227-256.

- [16] Talbi, E. (2009). *Metaheuristics: From Design to Implementation*. New Jersey, E.U.A: John Wiley & Sons, Inc.
- [17] Rothlauf, F. (2011). *Design of Modern Heuristics*. Heidelberg, Berlin: Springer.
- [18] Papadimitriou, C., Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. New Jersey, E.U.A: Prentice Hall.
- [19] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti, A., Protasi, M. (1999). *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Heidelberg, Berlin: Springer.
- [20] Du, K., Swamy, M. (2016). *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*. Heidelberg, Berlin: Springer.
- [21] Birattari, M. (2005). *Tuning Metaheuristics: A Machine Learning Perspective*. Heidelberg, Berlin: Springer.

# Índice alfabético

- actividad sin terminar, 26
- actividades ficticias, 21
- algoritmos de aproximación, 6
- algoritmos exactos, 6
- Alternativa con retraso, 27
  - mínima, 27
  - ruta crítica restante
    - cota inferior, 28
- AOA, 15
- AON, 15
  
- Branch and Bound, 28
- Búsqueda local, 7
- Búsqueda Tabú, 98
  
- conflicto de recursos, 27
- Conjunto de retraso, 27
- conjunto de vecinos admisibles, 98
- conjunto elegible, 26
- criterio de aspiración, 98
  
- Diversificación, 11
  
- espacio de soluciones, 4
- Estrategias de búsqueda, 10
  - métodos de búsqueda local, 10
  - métodos de recombinación, 10
  
- función objetivo, 9
  
- heurísticas, 6
  - construcción, 6
  - de mejora, 6
  
- Intensificación, 11
  
- localidad, 4
  
- memoria basada en frecuencias, 99
- memoria basada en lo reciente, 98
  
- metaheurísticas, 8
  - basadas en población, 13
  - basadas en una solución, 11
- modelo matemático, 3
- movimiento, 98
  - tabú, 98
  
- planeación, 14
- planeación de proyectos, 14
  - componentes, 14
- planeación en máquinas, 16
- planeación parcial, 26
  - dominada, 28
  - semi-activa, 26
- planeada inmediatamente, 27
- PRCPSP, 24
- problema de optimización, 4
  - combinatoria, 4
  
- RCPSP, 21
- Recocido Simulado, 44
- recursos no renovables, 14
- recursos renovables, 14
- reglas de dominio, 26
- ruta crítica restante, 28
  
- solución, 4
  - factible, 5
  - inicial, 10
  - representación, 9
  - vecina, 4
  - vecindario, 4
- subactividad, 24
  
- variables de decisión, 4
  
- óptimo global, 4
- óptimo local, 5

# Glosario

- actividades ficticias** Actividades en la planeación que no consumen tiempo ni recursos. 21
- AOA** Activity-On-Arc. 15
- AON** Activity-On-Node. 15
- conflicto de recursos** No se tiene la suficiente disponibilidad de los recursos para atender la demanda. 27
- criterio de término** Condición necesaria que se tiene que alcanzar para terminar la ejecución de una metaheurística. 8
- heurísticas subordinadas** Heurísticas guiadas por alguna estrategia de mayor nivel. 8
- interrupciones** Número de veces que se tiene que dividir una subactividad. 24
- modelo matemático** Representación matemática del problema junto a sus variables de decisión, restricciones. 3
- parámetros del problema** Datos (variables) que no posean valores específicos al problema. 3
- planeación** Asignar tiempo de finalización y unidades de algunos recursos a una actividad. 21
- postcondiciones** Condiciones esperadas que debe cumplir una solución al problema. 3
- PRCPSP** Pre-emptive Resource-Constrained Project Scheduling Problem. 24
- precondiciones** Condiciones bien especificadas sobre el número y tipo de datos del ejemplar de un problema. 3
- RCPSP** Resource-Constrained Project Scheduling Problem. 21
- recombinar soluciones** Combinar características de soluciones en la creación de otra solución. 10
- recursos renovables** Recursos que pueden ser utilizados en cualquier momento de la planeación siempre que exista disponibilidad. 14
- restricciones** Límites sobre el valor y relaciones que pueden tomar las variables en un problema. 3
- soluciones descendientes** Soluciones creadas a partir de recombinar algunas soluciones. 10