

Sistemas Inteligentes

Taller: Métodos de Búsqueda: ascenso a la colina y templado simulado

Integrantes:

Camilo Andres Pinilla

Nicolas Ricardo Enciso

Daniel Esteban Rodriguez

Problema:

Multimecánicas S.A. es una empresa metalmecánica que se dedica a hacer montajes para la industria alimenticia. En su último contrato se le adjudicó el diseño y construcción de un reactor químico para la industria lechera. Después del diseño y análisis se determinó que los siguientes rectángulos son piezas requeridas para su construcción.

Tabla 1. Dimensiones de las piezas necesarias para fabricar el reactor

Rectángulo	Dimensiones (cm x cm)
1	80 x 30
2	50 x 10
3	20 x 20
4	10 x 10
5	90 x 40
6	60 x 20
7	40 x 30
8	50 x 70
9	30 x 20
10	40 x 20

Como es una industria alimenticia se debe emplear acero inoxidable. Comercialmente cada lámina disponible tiene una dimensión de 1m x 1m. Dado su elevado costo es necesario minimizar el desperdicio de material a la vez que se produce la mayor cantidad de piezas anotadas anteriormente.

Encuentre las piezas a fabricar, su ubicación y orientación dentro de una lámina, indicando el área desperdiciada.

CONSIDERACIONES ESPECIALES:

- No existen pérdidas debidas al corte.
- La orientación sólo puede ser horizontal o vertical.
- Note que no todas las piezas caben en la lámina.
- Solo se necesita una unidad de cada rectángulo.

Modele e implemente una solución al problema:

1. Utilizando el método de Ascenso a la Colina (Hill-Climbing)
2. Utilizando Templado Simulado (Simulated Annealing)

Compare los resultados obtenidos por las soluciones implementadas usando los dos enfoques.

Solución

Hill Climbing Desarrollo Manual

Se toma la primera lámina de 100 cm x 100 cm y en ella se ubica la mayor cantidad de piezas posibles.

La 1ª en ser ubicada es la de mayor dimensión, la 5 de dimensiones 90 x 40 ubicada horizontalmente, quedando un espacio libre vertical de 10 x 40 y uno horizontal de 100 x 60.

Se ubica la 2ª pieza en dimensión, la 8 de dimensiones de 50 x 70 ubicada horizontalmente, quedando espacios libres verticales de 10 x 40 y 30 x 50 y uno horizontal de 100 x 10.

Se prueba la 3ª pieza en dimensión, la 1 con dimensiones 80 x 30, esta no puede ser ubicada en el espacio disponible.

Se prueba la 4ª pieza en dimensión, la 6 de dimensiones 60 x 20, esta no puede ser ubicada en el espacio disponible.

Se ubica la 5ª pieza en dimensión, la 7 de dimensiones 40 x 30 ubicada verticalmente, quedando espacios libres verticales de 10 x 40 y espacios horizontal de 100 x 10 y 30 x 10.

Se prueba la 6ª pieza en dimensión, la 10 de dimensiones 40 x 20, esta no puede ser ubicada en el espacio disponible.

Se prueba la 7ª pieza en dimensión, la 9 de dimensiones 30 x 20, esta no puede ser ubicada en el espacio disponible.

Se ubica la 8ª pieza en dimensión, la 2 de dimensiones 50 x 10 ubicada horizontalmente, quedando espacios libres verticales de 10 x 40 y espacios horizontal de 50 x 10 y 30 x 10.

Se ubica la 9ª pieza en dimensión, la 3 de dimensiones 20 x 20 ubicada horizontalmente, quedando espacios libres verticales de 10 x 40 y espacios horizontal de 30 x 10 y 10 x 10.

Se ubica la 10ª pieza en dimensión, la 4 de dimensiones 10 x 10 ubicada horizontalmente, quedando espacios libres verticales de 10 x 40 y espacios horizontal de 30 x 10.

Se toma entonces la segunda pieza de 100 x 100 y se repite el proceso con las piezas restantes.

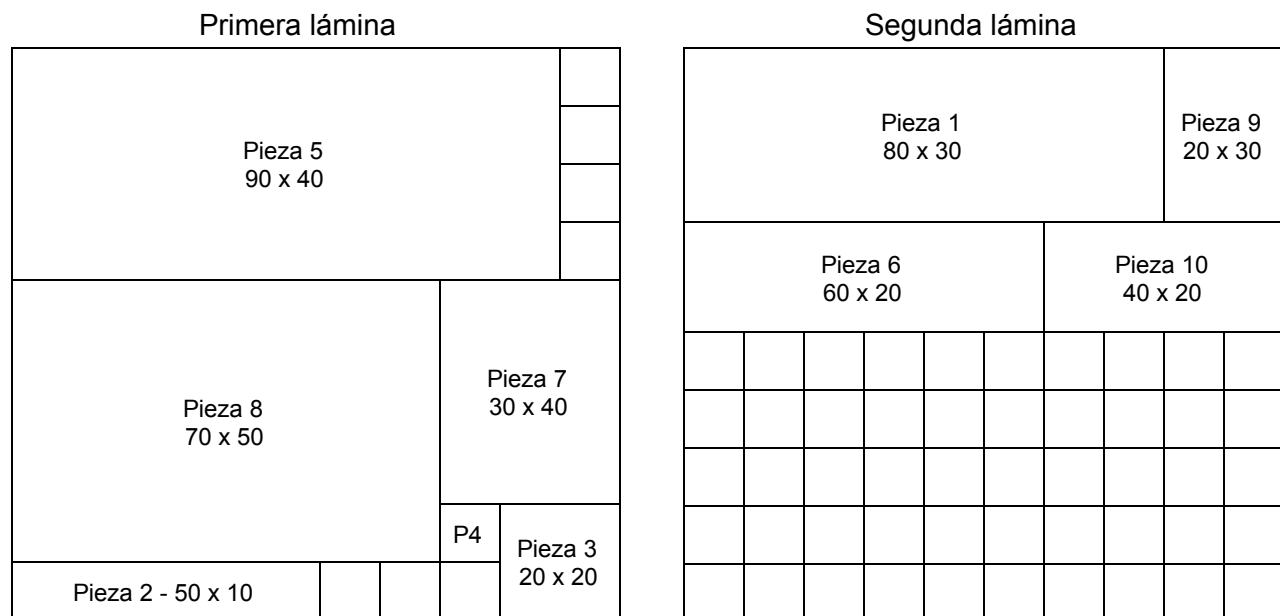
La 1ª en ser ubicada es la de mayor dimensión de las restantes, la 1 de dimensiones 80 x 30 ubicada horizontalmente, quedando un espacio libre vertical de 20 x 30 y uno horizontal de 100 x 70.

Se ubica la 2ª pieza en dimensión de las restantes, la 6 de dimensiones de 60 x 20 ubicada horizontalmente, quedando espacios libres verticales de 20 x 30 y horizontales de 100 x 50 y 40 x 20.

Se ubica la 3ª pieza en dimensión de las restantes, la 10 de dimensiones de 40 x 20 ubicada horizontalmente, quedando espacios libres verticales de 20 x 30 y horizontales de 100 x 50.

Se ubica la 4ª pieza en dimensión de las restantes, la 9 de dimensiones de 30 x 20 ubicada verticalmente, quedando espacios libres horizontales de 100 x 50.

De este modo se ubican todas las piezas, en dos láminas, en la primera se ubican 6 piezas (5, 8, 7, 2, 3, 4) y en la segunda las 4 piezas restantes (1, 6, 10, 9),



Hill Climbing Programado

Se realiza la implementación del algoritmo en Python, y se obtiene el siguiente resultado:

Se crea una clase Piece, para guardar cada pieza creada, esta tiene un estado de disponibilidad (si esta disponible o ya fue asignada), un número como identificador, sus dimensiones, las coordenadas relativas (si se encuentra como una sub pieza de otra pieza más

grande), un arreglo por si está contiene sub piezas, al igual que una matriz de contención o ubicación de cada una de esas sub piezas.

Se crea un arreglo con todas las piezas, necesarias para la fabricación.

Se crea la lámina (pieza grande en la cual ubicamos cada una de las pequeñas)

Se recorre el arreglo de las sub piezas insertandolas en la lámina, para esto se verifica cada una de las casillas de la matriz que conforma la lámina, y en cada casilla si se encuentra una casilla con 0 se prueba a poner la pieza en esa posición, si es posible, esta se marca como no disponible se cambian las coordenadas relativas en la lámina y en el arreglo de la lámina se inserta como una sub pieza al igual que se marca la ubicación de la misma en la matriz de contención, si no es posible realizar la asignación, se pueden realizar dos acciones:

1. caso 1 se continúa recorriendo el arreglo buscando otro espacio libre en el cual pueda ser asignada.
2. caso 2 se rota la pieza y se intenta nuevamente, de no ser posible se deja en su posición inicial y continua recorriendo el arreglo en busca de otra ubicación.

si al terminar de recorrer todo el arreglo de la lámina, no se logra ubicar se reporta como fallo.

Si la pieza pudo ser ubicada se continúa con la siguiente en el arreglo, si no pudo ser ubicada se guarda en un arreglo de piezas que faltan.

Al terminar de probar todas las piezas se verifica si quedan piezas pendientes por asignar, de ser así, se crea una nueva lámina y se repite el proceso.

Como todas las piezas y la lámina están en múltiplos de 10, se optó por escribir los números en múltiplos de 10, para facilitar el recorrido y la presentación por consola, al ejecutar este programa obtenemos los siguientes resultados.

Sin rotas las piezas:

```
lamina: 0
Piezas: 1 2 3 4 5 6 9
01 01 01 01 01 01 01 01 03 03
01 01 01 01 01 01 01 01 03 03
01 01 01 01 01 01 01 01 04 --
02 02 02 02 02 -- -- -- --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
06 06 06 06 06 06 09 09 09 --
06 06 06 06 06 06 09 09 09 --
```

```
lamina: 1
Piezas: 7 8 10
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
10 10 10 10 08 08 08 08 08 --
10 10 10 10 08 08 08 08 08 --
-- -- -- -- 08 08 08 08 08 --
-- -- -- -- 08 08 08 08 08 --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
```

Rotando las piezas:

```
lamina: 0
Piezas: 1 2 3 4 5 6
01 01 01 01 01 01 01 01 02 04
01 01 01 01 01 01 01 01 02 --
01 01 01 01 01 01 01 01 02 --
03 03 06 06 06 06 06 06 02 --
03 03 06 06 06 06 06 06 02 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
-- -- -- -- -- -- -- -- -- --
```

```
lamina: 1
Piezas: 7 8 9 10
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
09 09 09 -- 08 08 08 08 08 --
09 09 09 -- 08 08 08 08 08 --
10 10 10 10 08 08 08 08 08 --
10 10 10 10 08 08 08 08 08 --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
```

Temple Simulado Programado

Para este se decidió tomar como valor de temple, el orden de las piezas cada una se insertará de forma aleatoria, la aleatoriedad se realizará en el momento de insertar, de este modo si no todas caben, se cambiara el orden antes de proseguir con la siguiente lámina; la ejecución del programa es la misma, que en hill climbing.

Sin rotar las piezas:

```
lamina: 0
Piezas: 3 7 4 6 9 5 2 10
03 03 07 07 07 07 04 09 09 09
03 03 07 07 07 07 -- 09 09 09
-- -- 07 07 07 07 10 10 10 10
06 06 06 06 06 06 10 10 10 10
06 06 06 06 06 06 -- -- -- --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 05 --
02 02 02 02 02 -- -- -- -- --
```

```
lamina: 1
Piezas: 1 8
01 01 01 01 01 01 01 01 -- --
01 01 01 01 01 01 01 01 -- --
01 01 01 01 01 01 01 01 -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
```

Rotando las piezas (primera ejecución):

```

lamina: 0
Piezas: 6 10 7 4 3 5 2 9
06 06 06 06 06 06 10 10 10 10
06 06 06 06 06 06 10 10 10 10
07 07 07 07 04 03 03 09 09 02
07 07 07 07 -- 03 03 09 09 02
07 07 07 07 -- -- -- 09 09 02
05 05 05 05 05 05 05 05 05 02
05 05 05 05 05 05 05 05 05 02
05 05 05 05 05 05 05 05 --
05 05 05 05 05 05 05 05 --
-- -- -- -- -- -- -- -- --

```

```

lamina: 1
Piezas: 1 8
01 01 01 01 01 01 01 01 -- --
01 01 01 01 01 01 01 01 -- --
01 01 01 01 01 01 01 01 -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --
08 08 08 08 08 -- -- -- -- --

```

Rotando las piezas (segunda ejecución):

```

lamina: 0
Piezas: 4 5 1 3 6 9 2
04 05 05 05 05 05 05 05 05 05
-- 05 05 05 05 05 05 05 05 05
-- 05 05 05 05 05 05 05 05 05
-- 05 05 05 05 05 05 05 05 05
01 01 01 01 01 01 01 01 03 03
01 01 01 01 01 01 01 01 03 03
01 01 01 01 01 01 01 01 09 09
06 06 06 06 06 06 -- -- 09 09
06 06 06 06 06 06 -- -- 09 09
02 02 02 02 02 -- -- -- -- --

```

```

lamina: 1
Piezas: 10 8 7
10 10 10 10 08 08 08 08 08 --
10 10 10 10 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
07 07 07 07 08 08 08 08 08 --
-- -- -- -- 08 08 08 08 08 --
-- -- -- -- 08 08 08 08 08 --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --
-- -- -- -- -- -- -- -- -- --

```

Como podemos ver, al cambiar el orden de inserción, difícilmente se tendrán nuevamente los mismos resultados.