

# Descripción del Algoritmo

En este documento se presenta la información necesaria para implementar el algoritmo para la resolución del problema Bin Packing, mediante la metaheurística Variable Neighborhood Search (VNS).

## Variable neighborhood search

La metaheurística consiste en tres pasos:

1. **Shaking:** A la solución actual se le aplican ***k*th Neighborhood** de forma aleatoria, generando una **solución alpha**.
2. **Local Search:** Se ocupa el algoritmo de *Best First improvement*, en el cual consiste buscar el mejor movimiento para la **solución alpha**, generando otra **solución beta**.
3. **Move or not:** En esta paso se verifica si la **solución beta** es mejor que la **actual**, entonces esta es aceptada reiniciando el ciclo desde el inicio, caso contrario se sigue con el *k* siguiente.

A continuación se deja un pseudocódigo para mejor comprensión de este procedimiento.

---

**Algorithm 1: VNS**

---

**Result:**  $S^*$  best state

$S = \text{MBS}(1)$ ; Estado actual se inicializa mediante el método MBS;

$S^*$ ; Guarda el mejor estado ;

$k = 1$ ;

**while**  $k > k_{\max}$  **do**

$S' = \text{SHAKING}(S, k)$ ; Se aplica el shaking retornando otro estado;

$S'' = \text{LOCAL\_SEARCH}(S')$ ; Se aplica la búsqueda local al estado  $S'$ ;

**if**  $m(S'') < m(S)$  **then**

$S^* = S''$ ;

$k = 1$ ;

**else**

$k = k + 1$ ;

**end**

$S = S''$ ;

**end**

**return**  $S^*$ ;

---

Figura 1: Pseudocódigo VNS.

Para el problema se va ocupar la siguiente función objetivo que se busca maximizar en el proceso de la búsqueda local (*local search*).

$$f(x) = \sum_{\alpha}^m (l(\alpha))^2$$

m: La cantidad de bin.

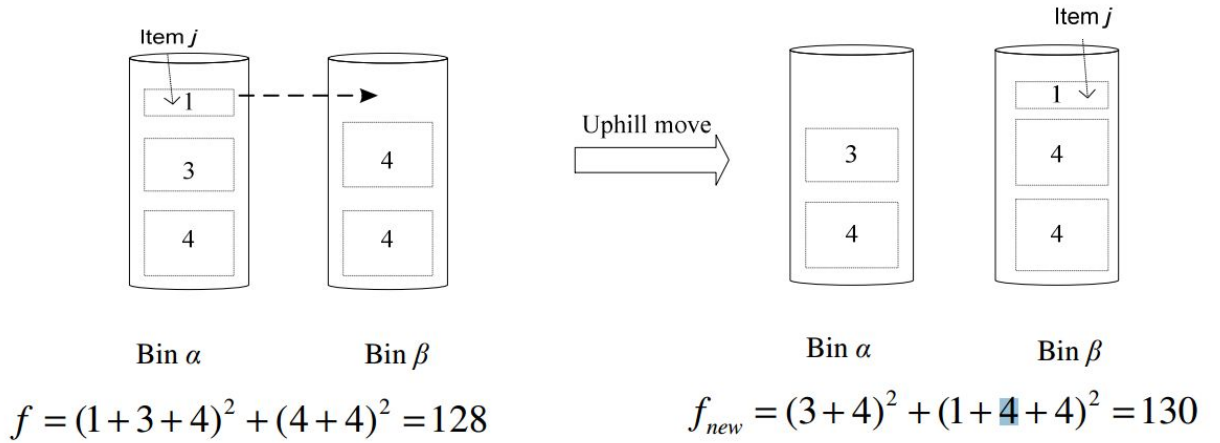
$l(a)$ : La suma de cada objeto en el contenedor (bin).

## Shaking

Para este proceso se debe seguir:

1. Se debe crear una lista  $Z'$  de todos los elementos de los contenedores y mantener el orden de mayor a menor.
2. Se selecciona un elemento  $i$  al azar de la lista  $Z'$ .
3. Todos los posibles movimiento de  $i$  son guardados con su correspondiente restricciones:
  - 3.1. **Transferir**: Este proceso contempla mover un elemento de un contenedor A al B como se puede ver en la Figura 1. El elemento  $i$  se mueve al contenedor B solo si el contenedor contiene espacio libre para poder recibir el elemento  $i$ .
  - 3.2. **Swap**: El proceso hace el cambio del elemento  $i$  con otro elemento  $j$ . Se debe cumplir que los contenedor de los elemento sea distinto, además de que el elemento  $i$  pueda estar en el contenedor B sin el elemento  $j$  y el caso contrario también debe cumplirse.
4. Si no hay más posibles movimientos para el elemento  $i$ , este es removido de la lista  $Z'$  y se vuelve a elegir un elemento  $i$  al azar de la lista  $Z'$ . En caso que haya movimientos posibles, se elige un movimiento al azar y se aplica.

Este proceso se repite  $k$  veces o hasta que no hayan elementos en la lista  $Z'$ .



$$\Delta f = (l_{\alpha} - t_j)^2 + (l_{\beta} + t_j)^2 - l_{\alpha}^2 - l_{\beta}^2$$

Figura 2: Movimiento de Transferencia.

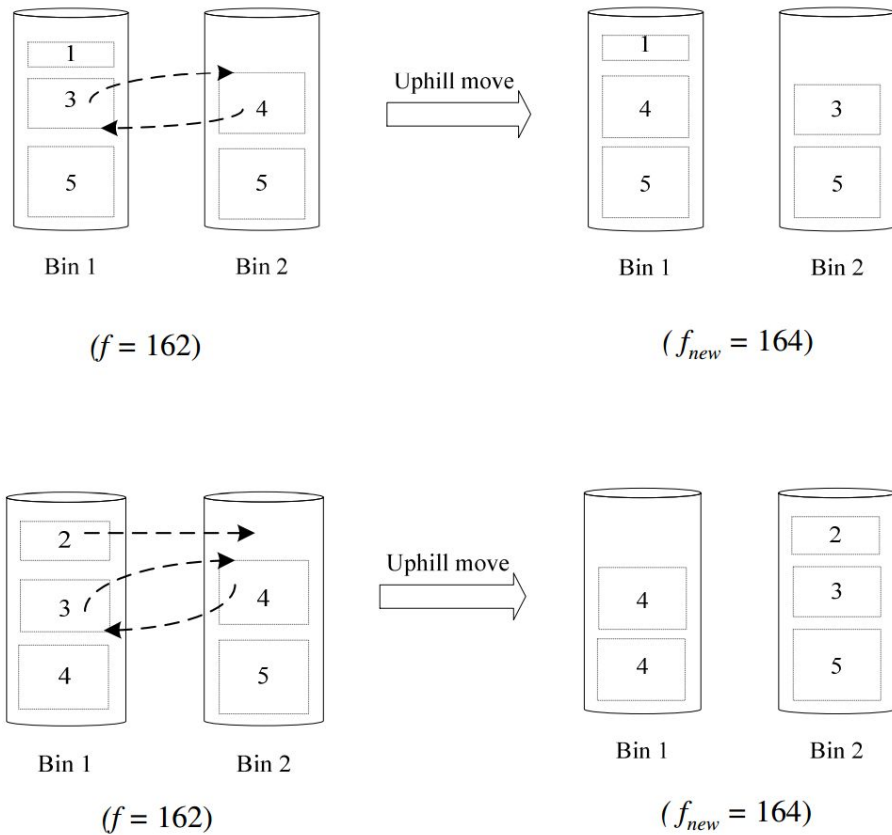


Figura 3: Movimiento Swap.

## Local Search

En este paso se busca el mejor movimiento posible para la solución, este es calculado de mediante la siguiente forma para el caso de transferir y swap:

$$\Delta f = [l(\alpha) - t_i]^2 + [l(\beta) + t_i]^2 - (l(\alpha))^2 - (l(\beta))^2$$

para el swap:

$$\Delta f = [l(\alpha) - t_i + t_j]^2 + [l(\beta) + t_i - t_j]^2 - (l(\alpha))^2 - (l(\beta))^2$$

Para el proceso de esta búsqueda del mejor movimiento:

1. Se obtiene una lista  $Z'$ , de los todos elementos de los contenedores que tengan espacio libre y se mantiene el orden de esta lista de mayor a menor.
2. **Transferir:** Se recorre la lista desde el final al inicio de esta, se debe cumplir que al transferir un elemento  $i$  al contenedor A, este debe tener el espacio suficiente, si es así, se calcula la función  $f$  de transferencia y se guarda siempre la de mayor valor.
3. **Swap:** Se deben seguir estos pasos:
  - 3.1.  $r = q - 1$ , donde  $q$  es el largo de la lista  $Z'$ .

- 3.2. Se itera hasta que el elemento  $r$  y  $q$  tengan distinto tamaño o valor, si no  $r$  es decrementado. (Elementos del mismo tamaño no cambian la función objetivo)
- 3.3. Se empieza desde  $r$  hasta el inicio de la lista  $Z'$ . Se debe cumplir que los contenedor de los elemento sea distinto, además de que el elemento  $i$  pueda estar en el contenedor B sin el elemento  $j$  y el caso contrario también debe cumplirse. Se calcula la  $f$  para guardar el swap con mayor valor.
4. Se aplica el mejor movimiento (mayor valor) entre la transferencia y swap.