

**Algoritmos metaheurísticos para el problema de ubicación de  
instalaciones P-Mediana no capacitado**

**Santiago José Guzmán Villamarín  
Faber Duban Garcés Gómez**

**Anexos algoritmos metaheurísticos para el problema de ubicación  
de instalaciones P-mediana no capacitada**

**DIRECTOR: DRA. MARTHA ELIANA MENDOZA BECERRA  
CO-DIRECTOR: DR. CARLOS ALBERTO COBOS**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
GRUPO DE I+D EN TECNOLOGÍAS DE LA INFORMACIÓN (GTI)  
LÍNEA INVESTIGACIÓN: SISTEMAS INTELIGENTES  
POPAYÁN  
Noviembre 2021**

## Contenido

Anexo A: Búsqueda Armónica Binaria Simplificada (SBHS).....	3
1. Generación de una nueva solución .....	3
2. Reparación.....	4
Calculo de menores distancias .....	4
Activación vértice .....	5
Desactivación vértice .....	5
3. Evaluación.....	6
4. Búsqueda local.....	6
5. Reemplazo .....	7
6. Selección del mejor .....	8
Anexo B: Búsqueda de Organismos Simbióticos Híbridos (HSOS) .....	8
1. Procesos de fase.....	9
2. Mutualismo .....	9
3. Comensalismo.....	10
4. Parasitismo .....	11
5. Armonía.....	11

## Tabla Figuras

Figura 1. Algoritmo SBHS .....	3
Figura 2. Generación de una nueva solución .....	3
Figura 3. Reparación de la solución .....	4
Figura 4. Calcular menores distancias .....	4
Figura 5. Obtener vértice para activar .....	5
Figura 6. Obtener vértice para desactivar .....	6
Figura 7. Evaluación de la solución.....	6
Figura 8. Búsqueda local .....	7
Figura 9. Reemplazo de solución .....	7
Figura 10. Selección del mejor.....	8
Figura 11. Algoritmo HSOS.....	8
Figura 12. Procesos de fases.....	9
Figura 13. Fase de Mutualismo .....	10
Figura 14. Fase de Comensalismo.....	10
Figura 15. Fase de Parasitismo .....	11
Figura 16. Fase de Armonía.....	12

## Anexo A: Búsqueda Armónica Binaria Simplificada (SBHS)

Al algoritmo de SBHS se hicieron modificaciones en los métodos de inicialización, la construcción de una nueva solución, en el método de reparación, de evaluación, además de que se adicione una búsqueda local, y se mantuvo base los métodos de reemplazo y selección del mejor (Ver Figura 1).

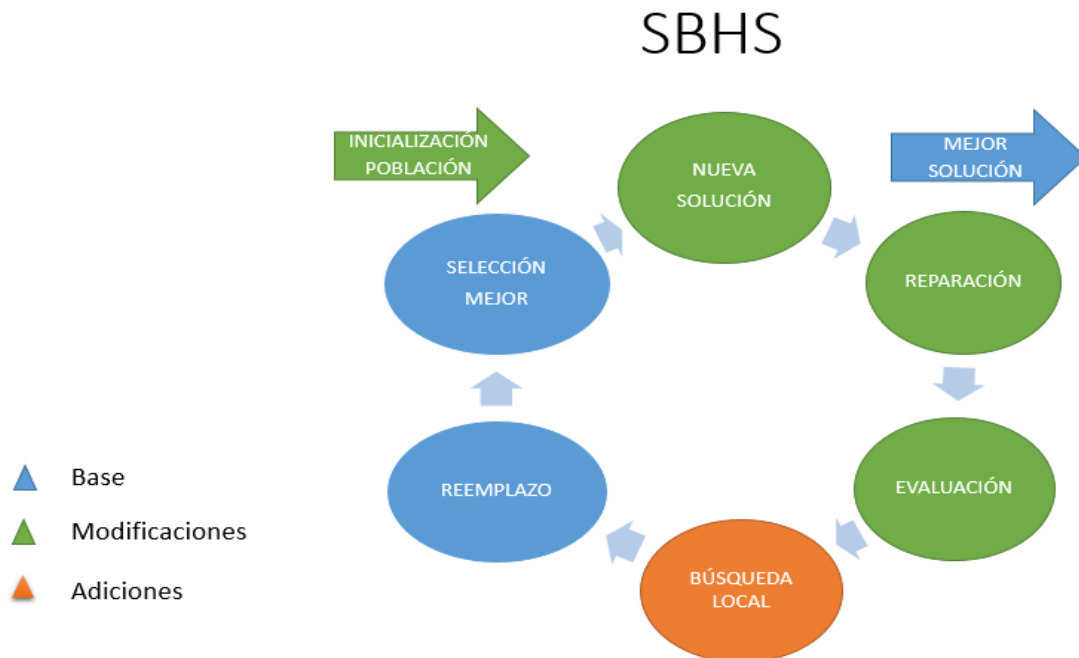


Figura 1. Algoritmo SBHS

### 1. Generación de una nueva solución

Se genera aleatoriamente a partir de una probabilidad del 70%, a partir de esto dado a que no se ajusta al problema, esta debe ser reparada y evaluada, y de esta forma conocer que solución es mejor que otra según el valor de la función objetivo (Ver Figura 2).

## GENERACIÓN DE UNA NUEVA SOLUCIÓN

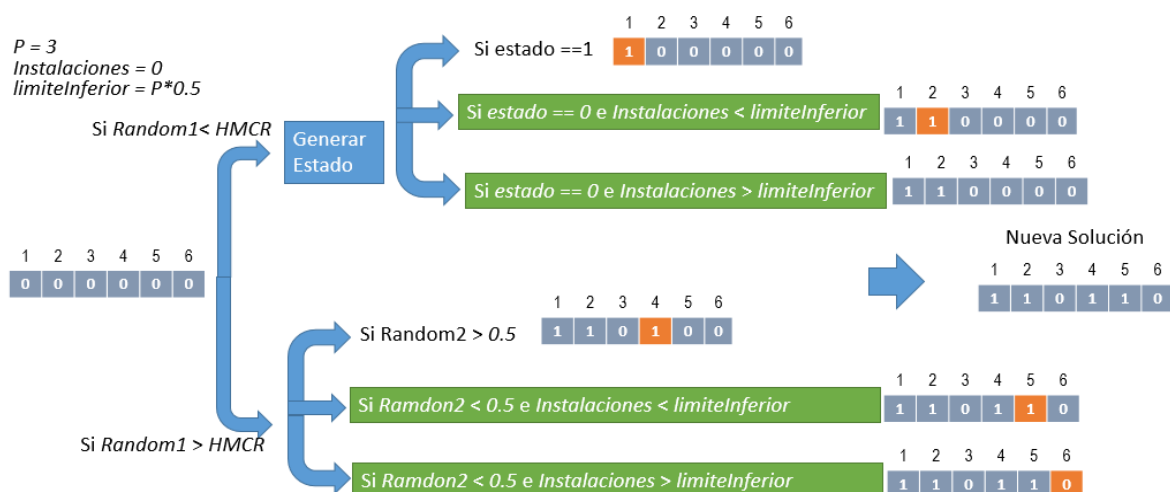


Figura 2. Generación de una nueva solución

## 2. Reparación

Dado que esta solución debe ser reparada ya que tiene 4 vértices activados y no cumple con el número de instalaciones a ubicar  $P = 3$ , se aplica un proceso de obtener un vértice para desactivar y de esta forma obtener una solución reparada (Ver Figura 3).

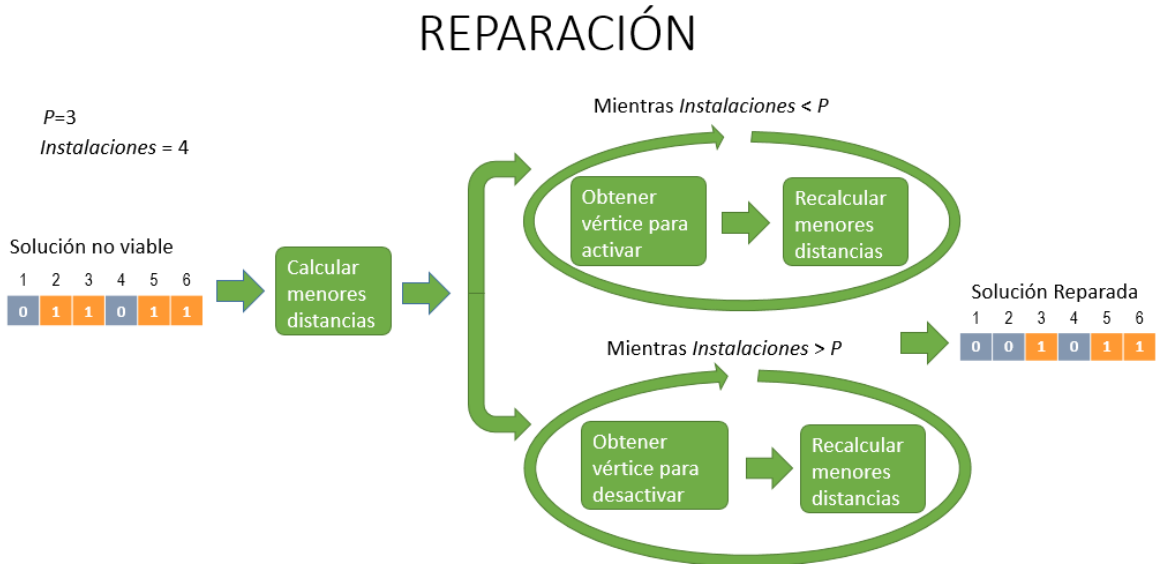


Figura 3. Reparación de la solución

### Calculo de menores distancias

Para la reparación se realiza el cálculo de menores distancias donde se toma los vértices definidos como instalaciones  $j$  y haciendo uso de la matriz de Floyd se calcula para cada punto de demanda  $i$  la menor distancia hacia las instalaciones, obteniendo así el vector de menores distancias para cada demanda (ver Figura 4).



Figura 4. Calcular menores distancias

### Activación vértice

Para el proceso de obtener un vértice para activar, se toma al conjunto  $x$  de vértices donde no se ha ubicado una instalación y se para cada vértice  $j$  se realiza una resta igual a  $g$  entre la distancia de cada demanda a dicho vértice y el valor de menor distancia  $\delta$ , luego se suma para cada vértice posible a ubicar aquellos valores de pérdida menores a cero y se selecciona aquel que presente la menor pérdida para posteriormente activarlo como una instalación (Ver Figura 5).

## REPARACIÓN

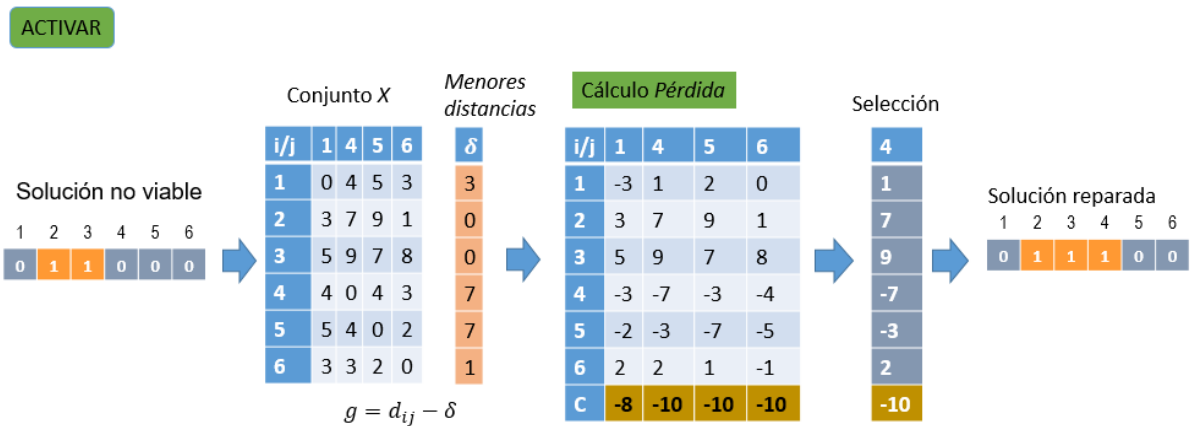


Figura 5. Obtener vértice para activar

### Desactivación vértice

En caso donde se tenga una solución con más instalaciones de las que se desea ubicar, se realiza el proceso de obtener un vértice para desactivar, en donde se toma el conjunto de vértices activados y para cada uno de ellos se realiza un cálculo de reubicación, definiendo para cada una de las demandas  $i$  la distancia más cercana hacia las demás instalaciones  $j$  diferentes a la actual, luego con esta matriz de reubicación y el vector de menores distancias  $\delta$  se calcula para cada instalación  $j$  y demandas  $i$ , el cálculo de ganancia  $g$  que tendría la función objetivo al eliminar cierto vértice, se suman estos cálculos de ganancia para cada instalación y aquella que presente el menor valor de aumento a la función objetivo se desactiva obteniendo una solución viable (Ver Figura 6).

# REPARACIÓN

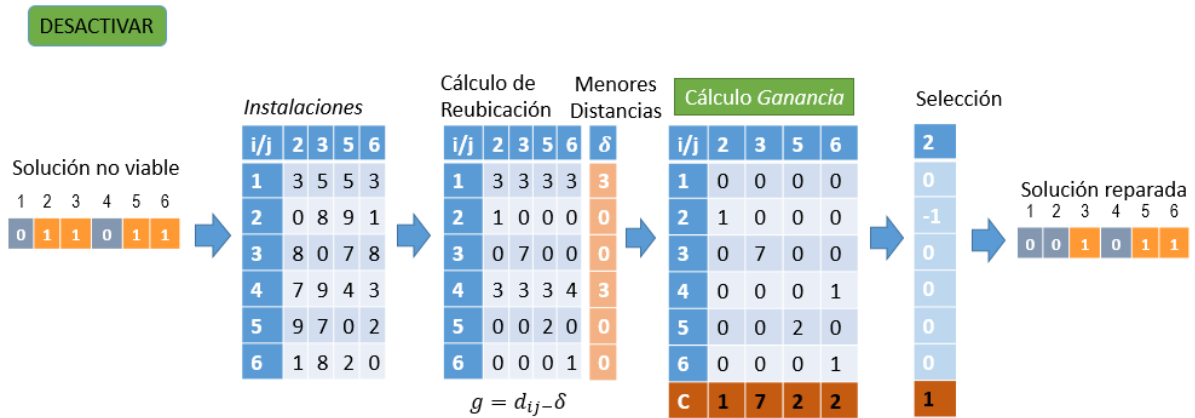


Figura 6. Obtener vértice para desactivar

## 3. Evaluación

Nuestra solución reparada posteriormente es evaluada con el fin de obtener el valor de evaluación de la función objetivo, para esto se toma de nuestra solución los vértices activados como instalaciones y por medio del uso de la matriz de Floyd calculamos para cada una de las demandas  $i$  cual es la menor distancia entre las distancias a las instalaciones  $j$ , luego sumamos cada una de estas menores distancias obteniendo nuestro valor de evaluación de función objetivo  $f_o$  (Ver Figura 7).

# EVALUACIÓN

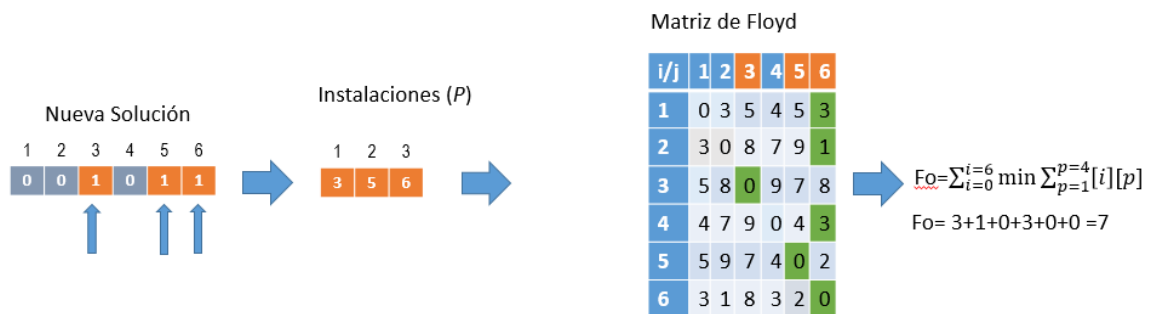


Figura 7. Evaluación de la solución

## 4. Búsqueda local

Parte de la construcción de 2 soluciones, la mejor de la población y la nueva solución, generando un cruce en cada uno de sus estados de forma uniforme, esta solución cruce debe ser reparada y evaluada para ser comparada con la nueva solución, si el valor de la función objetivo es mejor, está la modificara (Ver Figura 8).

# BÚSQUEDA LOCAL

## OPERADOR DE CRUCE

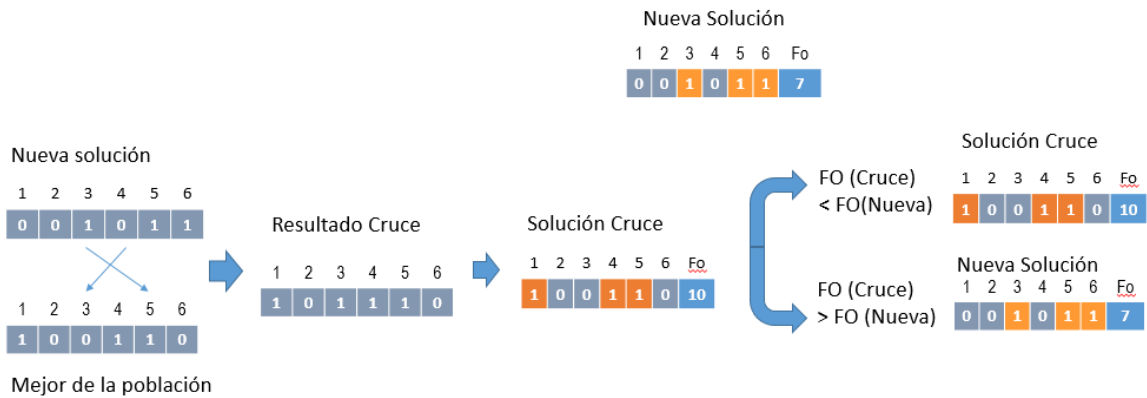


Figura 8. Búsqueda local

## 5. Reemplazo

A esta solución que ha obtenido un proceso de reparación, evaluación y aplicado un método de búsqueda local, es comparada con el peor de la población, si esta es mejor, reemplazara a la peor solución de la población (Ver Figura 9).

# REEMPLAZO

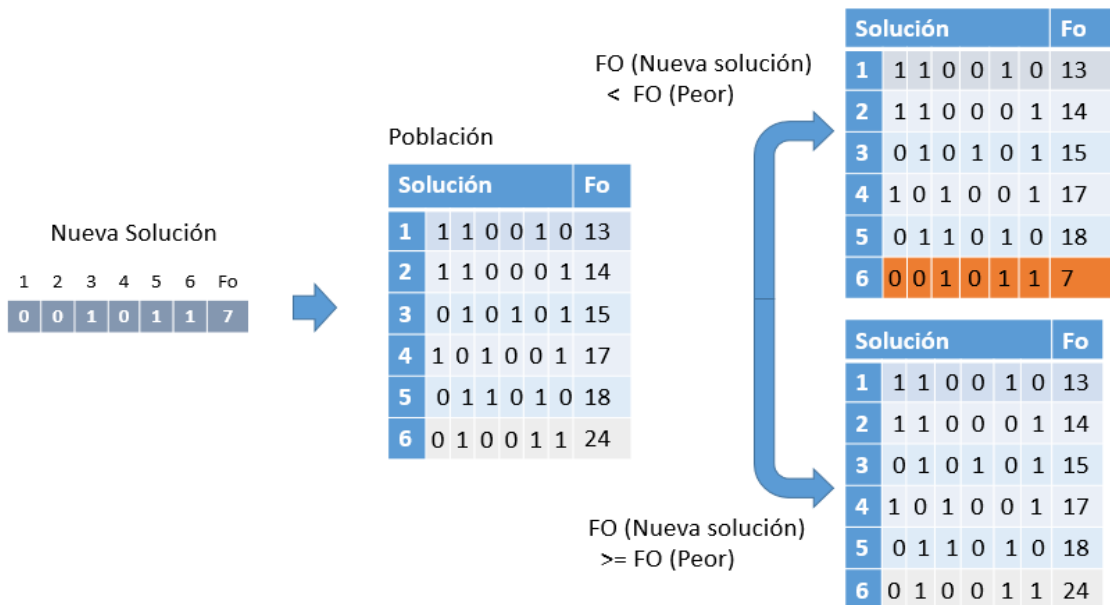


Figura 9. Reemplazo de solución

## 6. Selección del mejor

Para finalizar se selecciona a la mejor de la población, aplicando un ordenamiento según el valor de la función objetivo (Ver Figura 10).

# SELECCIÓN DEL MEJOR



Figura 10. Selección del mejor

## Anexo B: Búsqueda de Organismos Simbióticos Híbridos (HSOS)

En el algoritmo de HSOS, se hicieron modificaciones en el método de inicialización, usando el mismo que el usado en SBHS, además se hicieron modificaciones en las fases naturales de mutualismo, comensalismo, parasitismo y armonía (ver Figura 11)

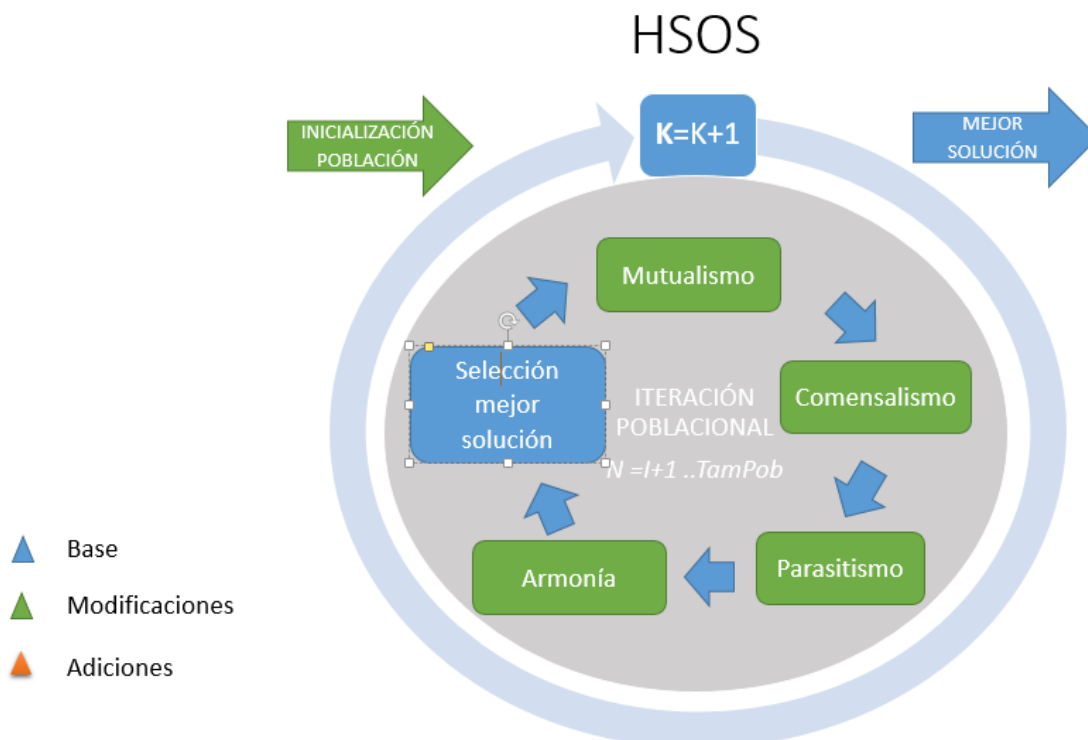


Figura 11. Algoritmo HSOS



## 1. Procesos de fase

En la iteración poblacional cada una de nuestras soluciones entra a los distintos procesos de fase, donde en cada uno de ellas pasa por los métodos de evolución de fase, luego por los métodos anteriormente vistos para *SBHS*, los cuales son reparación, evaluación y reemplazo, realizando modificaciones en los métodos de evolución y reemplazo y entrando la búsqueda local como adicción al cada una de las fases (Ver Figura 12).



Figura 12. Procesos de fases

## 2. Mutualismo

En el método de evolución de la fase de Mutualismo, tomamos nuestra solución actual y además seleccionamos aleatoriamente una solución de la población distinta a la actual, luego generamos un vector *mutual* a partir de la probabilidad del 50% que cada vértice tome el estado ya sea del vértice común de la solución *actual* o de la solución *seleccionada aleatoriamente*, luego se crea una solución *A* *aleatoriamente* y se altera cada uno de los vértices de acuerdo a la igualdad de estado entre los vértices de la solución *mutual* y el mejor de la población, de la misma forma generamos aleatoriamente una solución *B* y alteramos sus vértices de acuerdo a la igualdad de estados entre la solución *actual* y la solución *A* *alterada* previamente (Ver Figura 13).

# MUTUALISMO

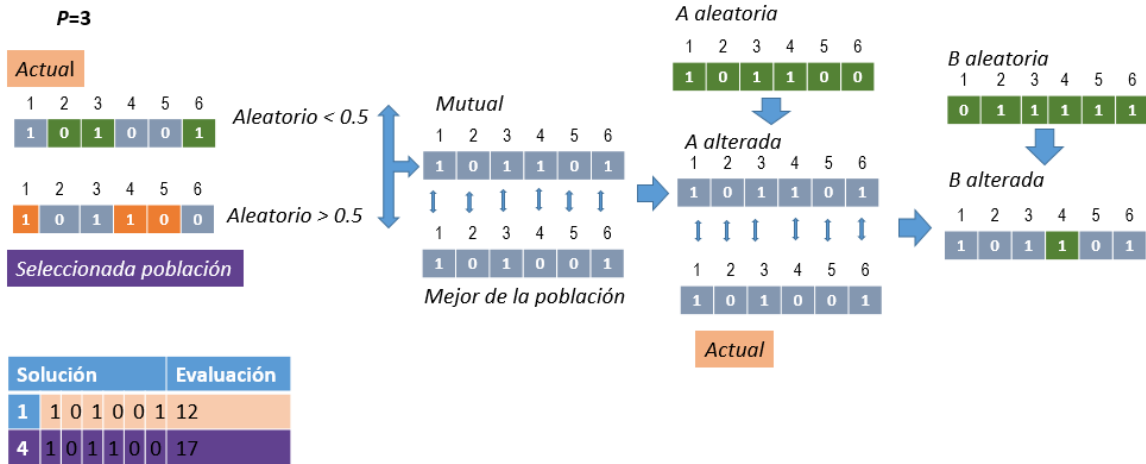


Figura 13. Fase de Mutualismo

## 3. Comensalismo

Luego se crea una solución *C aleatoriamente* y se altera cada uno de los vértices de acuerdo a la igualdad de estado entre los vértices de la solución seleccionada aleatoriamente de la población y el mejor de la población, de la misma forma generamos aleatoriamente una solución *D* y alteramos sus vértices de acuerdo a la igualdad de estados entre la solución actual y la solución *C alterada* previamente (Ver Figura 14).

# COMENSALISMO

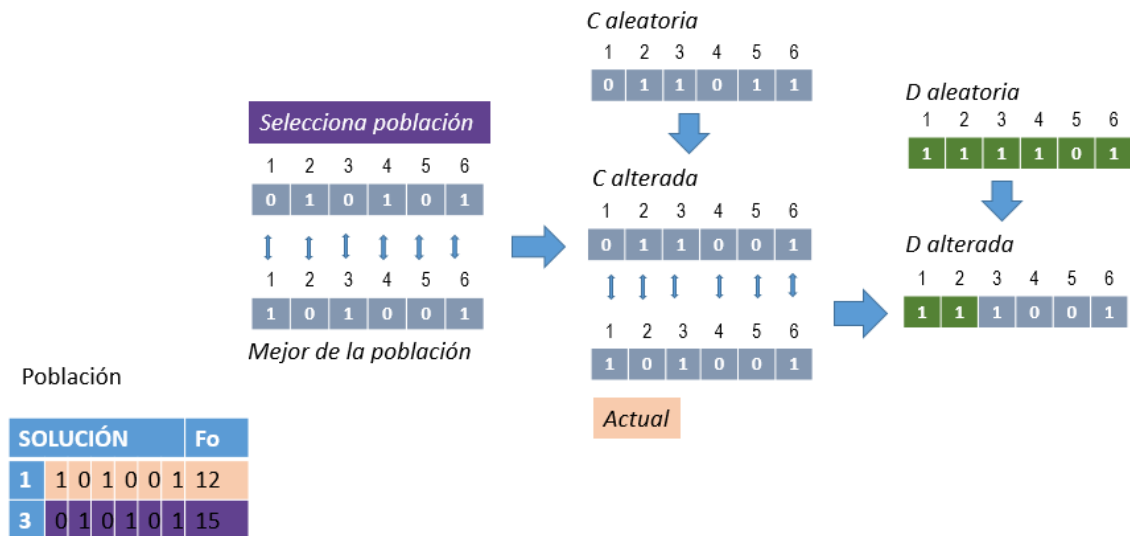


Figura 14. Fase de Comensalismo

#### 4. Parasitismo

En el método de evolución de la fase de parasitismo el proceso de fase de parasitismo, tomamos nuestra solución actual y generamos a partir de ella una copia, luego seleccionamos aleatoriamente un número de vértices y les realizamos un proceso donde invertimos su estado obteniendo una solución *parasito* (Ver Figura 15).

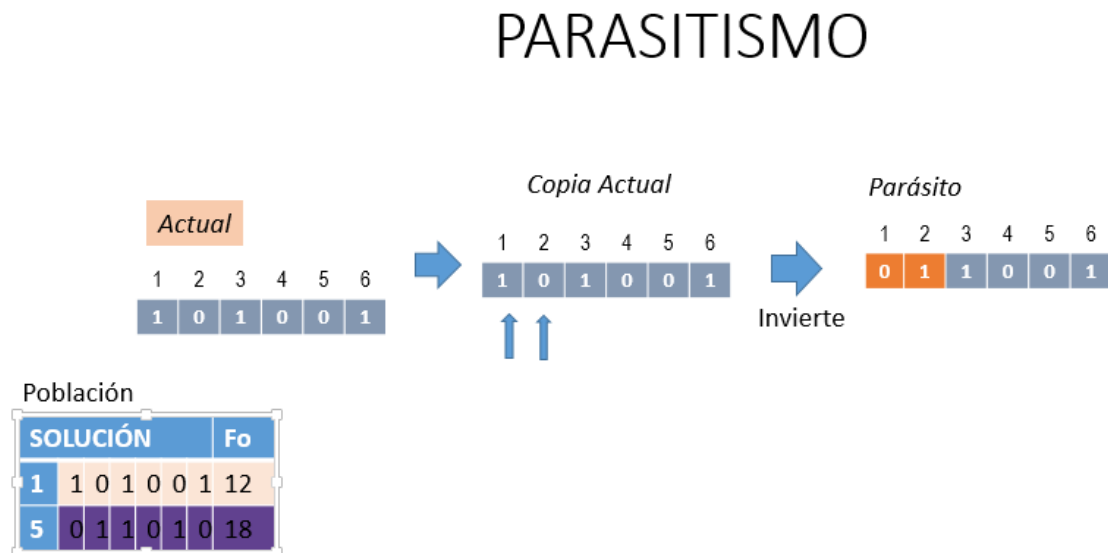


Figura 15. Fase de Parasitismo

#### 5. Armonía

En la evolución de la fase de armonía creamos una solución vacía llamada *Neko* para cada uno de sus vértices realizamos un proceso donde generamos un *aleatorio1* entre 0 y 1, si el valor de este aleatorio es menor o igual a la tasa de consideración de la memoria armónica *HMCR*, para el vértice actual se tomara el estado del mismo vértice de una solución selecciona aleatoriamente de la población, luego de acuerdo a un segundo aleatorio, si este es mayor a la tasa de ajuste de tono *PAR* el estado obtenido se mantendrá por el contrario si el aleatorio es menor o igual al *PAR*, se recurrirá a un tercer aleatorio que de acuerdo a una probabilidad del 50% definirá si se mantiene el estado del vértice o cambia. Por otro lado si nuestro primer aleatorio es mayor a *HMCR*, se seleccionará aleatoriamente un 1 o 0 para definir el estado del vértice si es 0 el estado será de apagado y si es 1 de activado es decir nuestro vértice de demanda pasara a ser una instalación. De la misma forma se realiza este procedimiento para cada uno de los vértices de nuestra solución *Neko* obteniendo una solución completa (Ver Figura 16).

# ARMONÍA

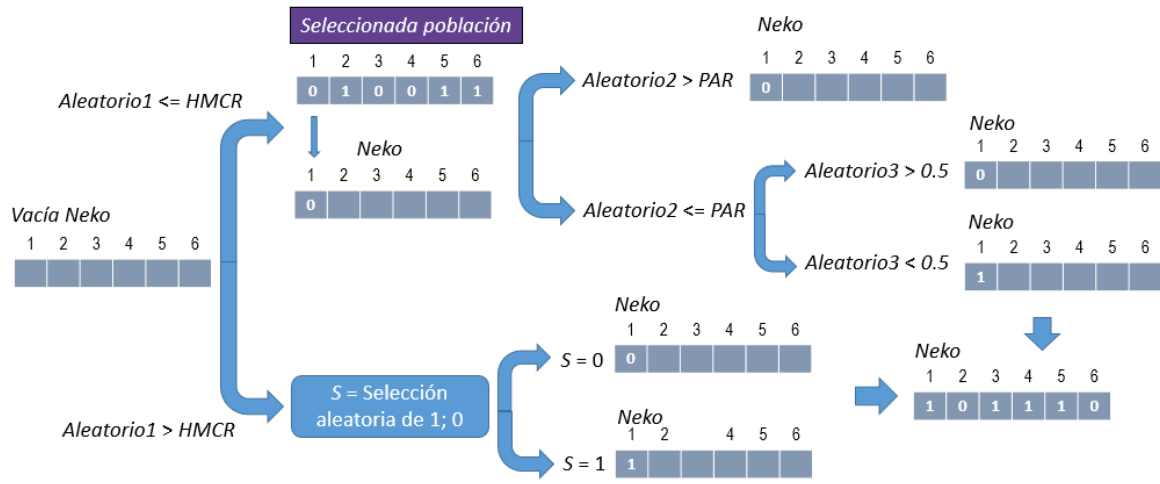


Figura 16. Fase de Armonía