



5. DISEÑO FUNCIONAL

En este apartado vamos a detallar los diagramas funcionales que han constituido la base para la posterior implantación informática de la metaheurística.

5.1. Diseño funcional del algoritmo generado

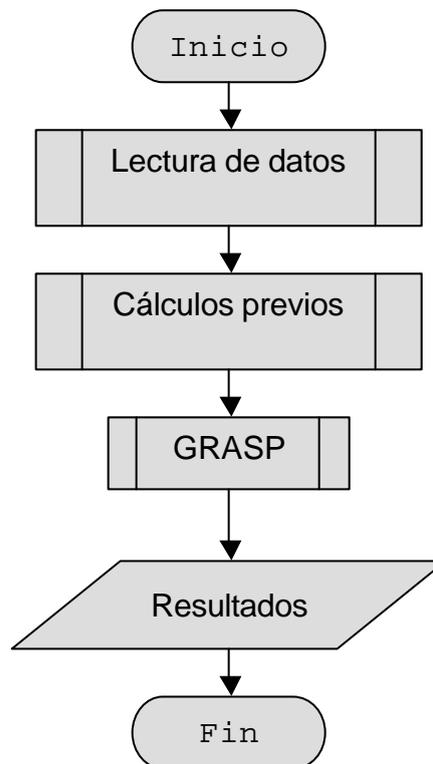


Figura 5.1: Diagrama de flujo del algoritmo.

5.1.1. Módulo de lectura de datos

Con este módulo el algoritmo toma todos los datos necesarios para efectuar las operaciones programadas. Dicha información es la siguiente:

- Número de trabajos a procesar.
- Instantes de comienzo y finalización de cada uno de los trabajos.
- Peso de cada uno de los trabajos.
- Número de máquinas
- Número de clases de trabajo

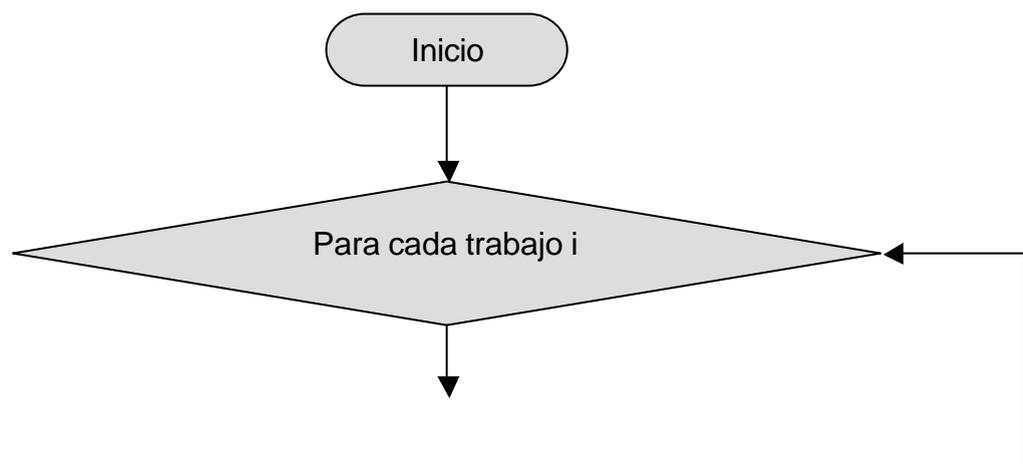


- Número de clases de máquinas
- Número de iteraciones a completar.
- Valor de α o tamaño de la RCL (según caso).

5.1.2. Módulo de cálculos previos

Una vez que se dispone de toda la información necesaria para el correcto funcionamiento del algoritmo, es recomendable la realización de una serie de cálculos previos. Con ello nos evitaremos tener que realizarlos reiteradamente durante el desarrollo de las iteraciones.

Estos cálculos consisten en identificar los pedidos que se solapan en algún instante durante sus procesamientos. Hemos de tener en cuenta que los movimientos a estudiar siempre involucrarán trabajos que se superponen entre sí y, según las características del problema, una misma máquina no puede procesar dos trabajos en un mismo intervalo de tiempo. Por lo tanto conocer a priori los solapamientos entre las tareas nos permitirá desarrollar un algoritmo mucho más rápido, pues sabremos qué trabajos podrán ser asignados a las máquinas en cada instante. La forma de proceder es la siguiente:



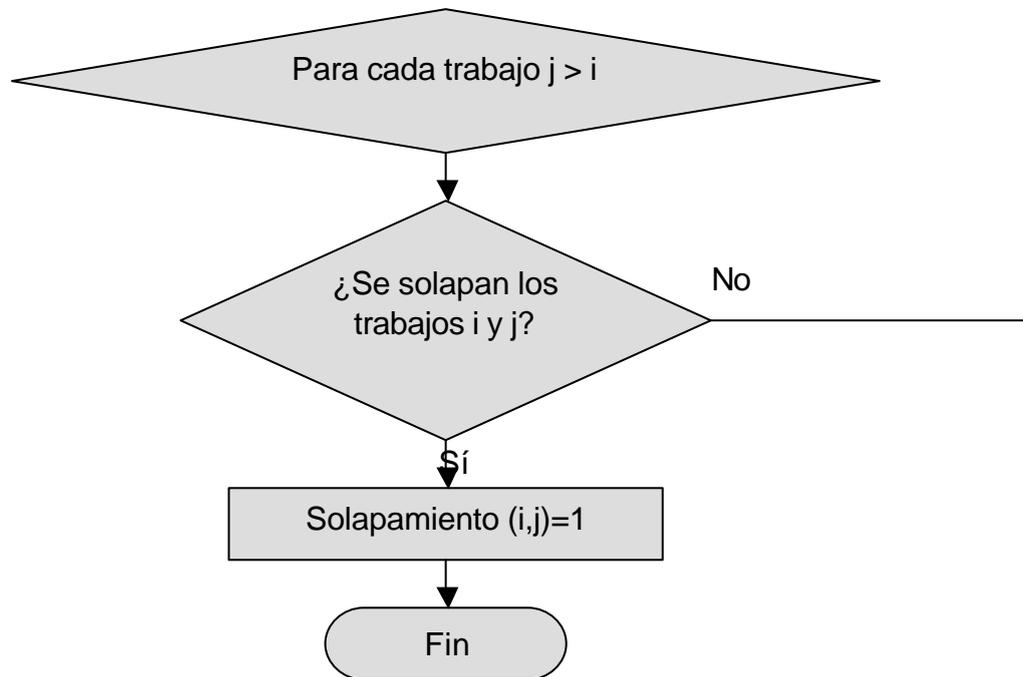
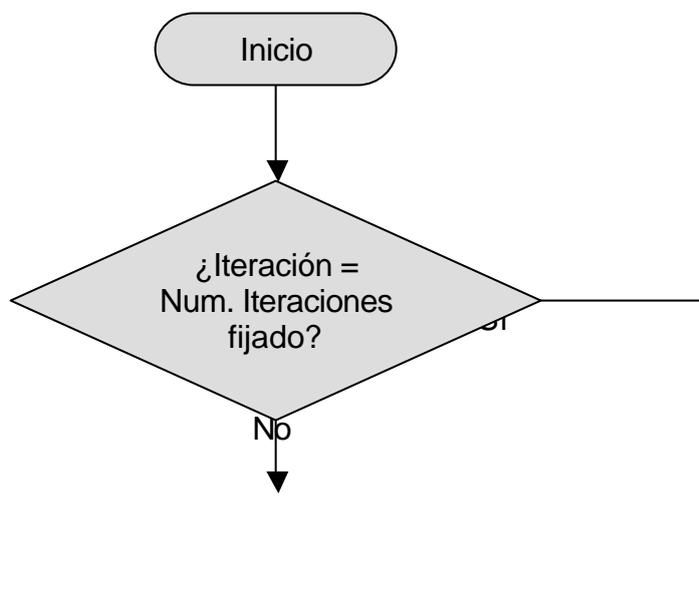


Figura 5.2: Diagrama de flujo del módulo de cálculos previos.

5.1.3. Módulo de la metaheurística GRASP

Según se ha descrito, cada iteración del GRASP consta de dos fases distintas: una primera fase en la que se construye la solución de partida y una segunda en la que, mediante una metodología de búsqueda local, se trata de mejorar la solución construida en la fase anterior.

Según esto, el diagrama funcional de este módulo adquiere la siguiente forma:



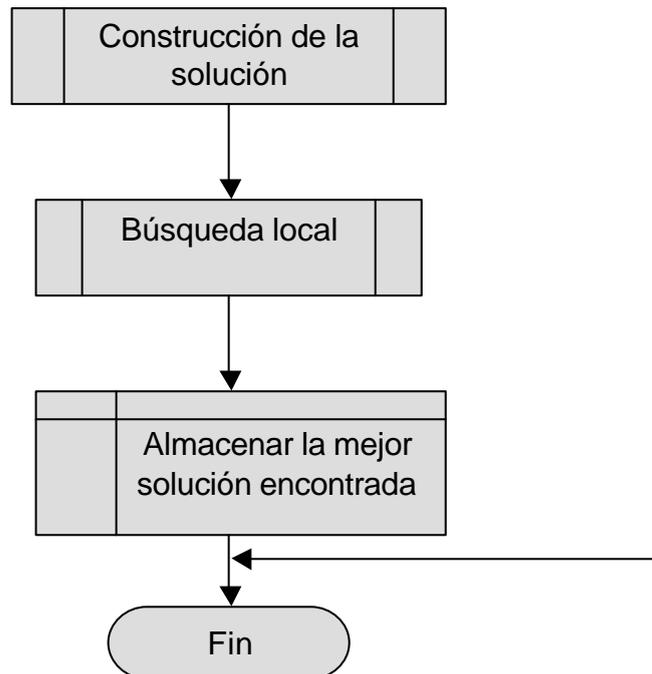
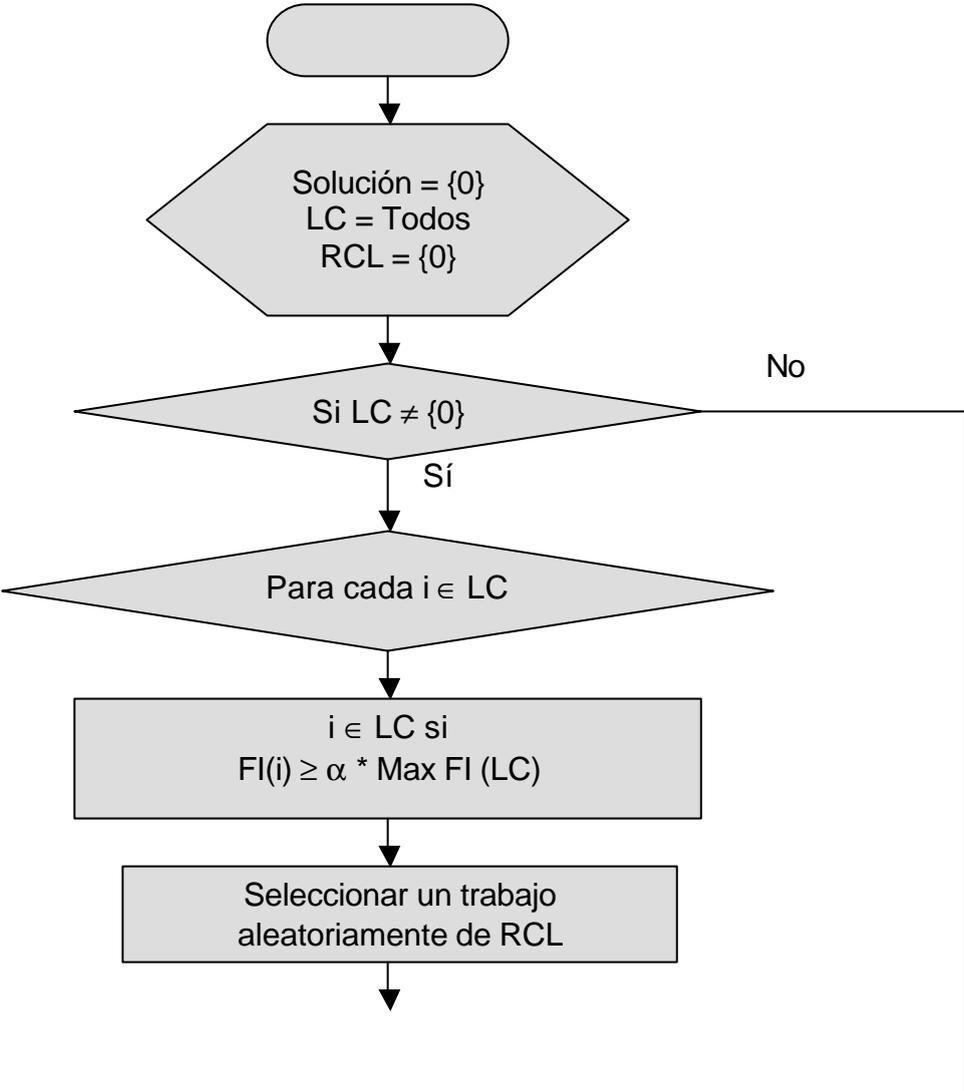


Figura 5.2: Diagrama de flujo de la metaheurística GRASP.

El procedimiento seguido en la fase de construcción depende de la vía de investigación utilizada:

1. Se genera la lista de candidatos (LC) con aquellos trabajos que, no perteneciendo a la solución, no presentan superposición con alguno de los elementos de la misma. A partir de este conjunto, se forma una lista restringida de candidatos (RCL) con las tareas cuyo valor de la función índice (FI) esté dentro del porcentaje α del valor mayor de todos los candidatos. Posteriormente se selecciona aleatoriamente un candidato que, según su tipo, se asigna a una máquina determinada. El proceso termina cuando se agota el número de candidatos.



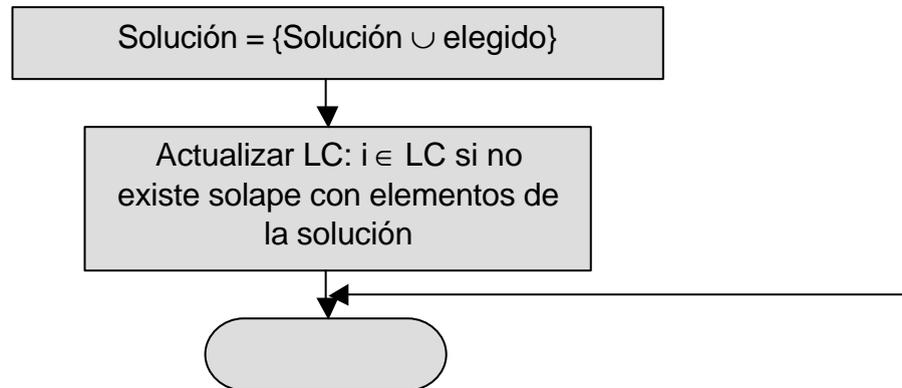


Figura 5.3: Diagrama de flujo de la fase de construcción con RCL dependiente de a .

2. Partiendo de una lista en la que los trabajos no pertenecientes a la solución están ordenados con relación a su FI, se estudian los solapamientos existentes entre dichos trabajos y los pertenecientes a la solución de forma que se construye la LC eliminando de la lista inicial aquellos trabajos solapados con algún elemento de la solución. La RCL se formará con los candidatos de mayor FI, de forma que sólo entrarán en ella un número de candidatos igual al tamaño prefijado en el módulo de lectura. Así, por ejemplo, si el tamaño de la RCL se ha fijado en el inicio en cuatro unidades, ésta se constituirá con los cuatro candidatos cuyos FI sean los mayores. Una vez diseñada la RCL se seleccionará aleatoriamente un candidato que, según su tipo, se asignará a una máquina determinada. Al igual que en el caso anterior, el proceso termina cuando se agota la LC.

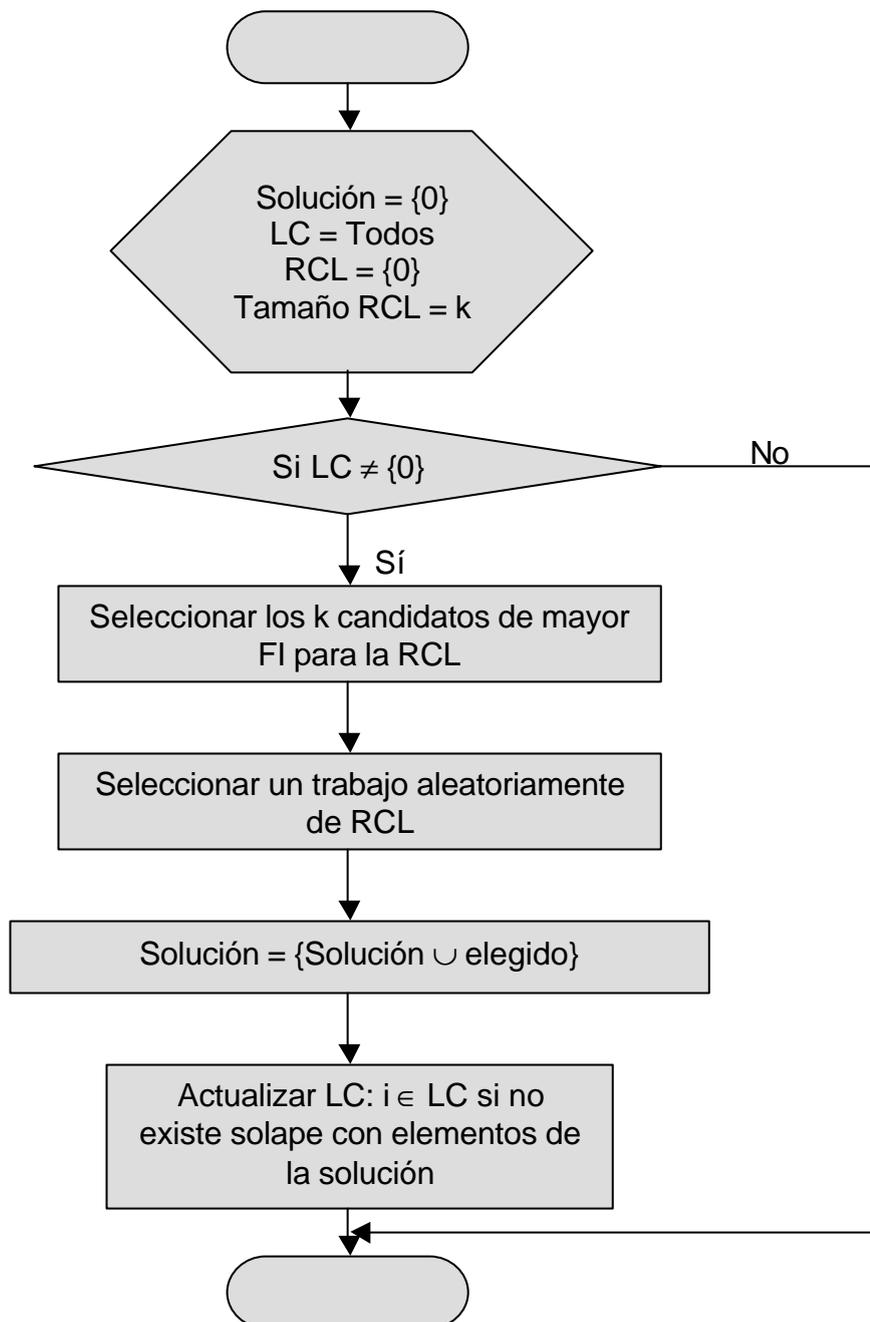




Figura 5.4: Diagrama de flujo de la fase de construcción con tamaño de RCL fijo.

En la fase de búsqueda local se realiza una exploración de la vecindad de la solución construida en la fase anterior para elegir el mejor movimiento posible, siempre que dicho movimiento nos lleve a una situación mejor que la actual. Para cada solución encontrada, se estudia si se mejora el mayor valor obtenido, en cuyo caso se almacena como posible solución del algoritmo. El proceso finaliza cuando no se encuentra una solución vecina que mejore la función objetivo del problema.

El diagrama de flujo, en este caso, tiene la siguiente forma:



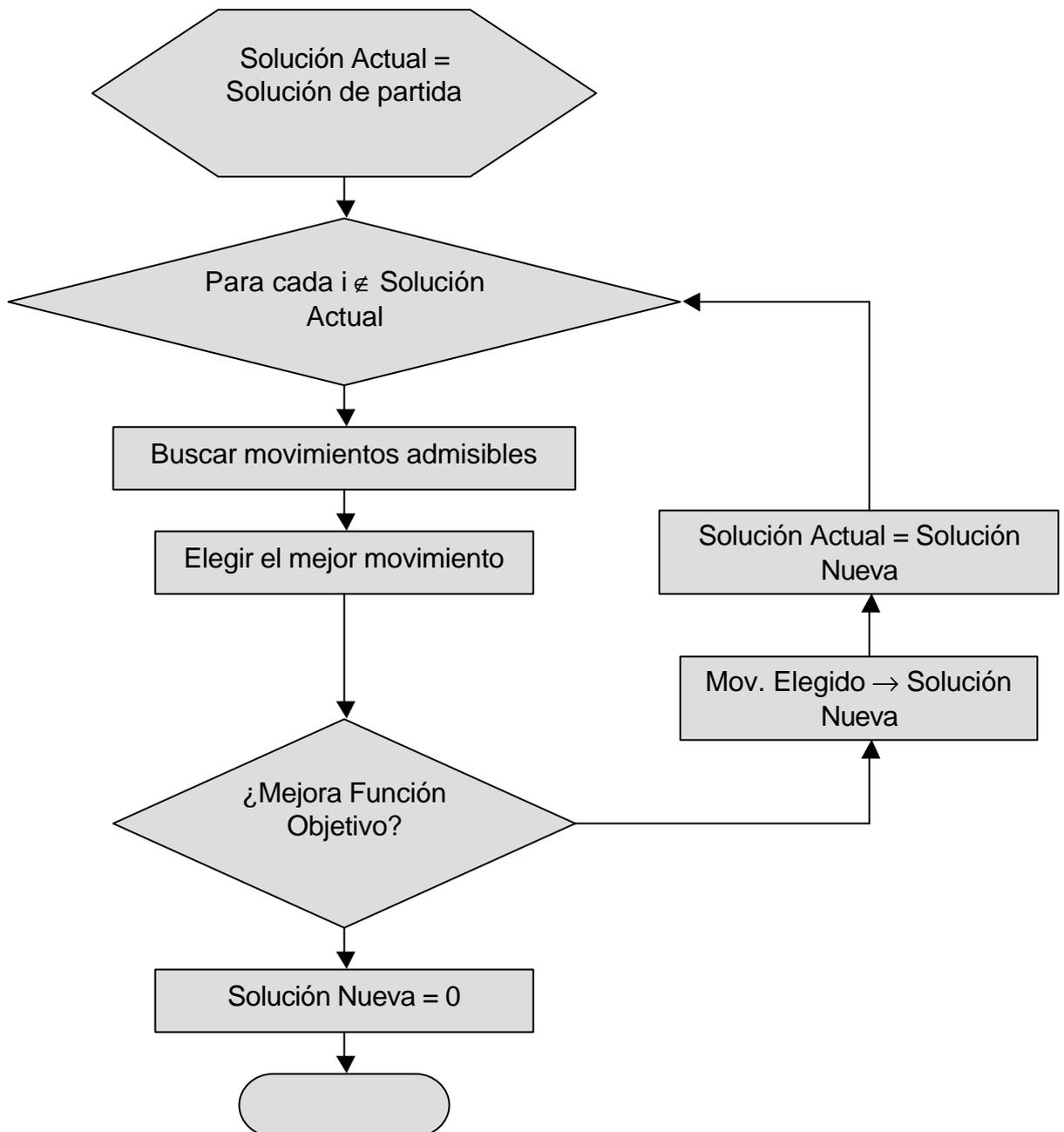


Figura 5.5: Diagrama de flujo de la búsqueda local.

En el módulo “buscar soluciones admisibles” se realizan las siguiente operaciones:

Se estudia en primer lugar si algún trabajo i no perteneciente a la *Solución Actual* puede añadirse a ella. Para ello se debe cumplir que:



$$\sum_{j \in \text{Sol. Actual}} \text{Solapamiento}(i, j) \times q$$

Siendo q el número de máquinas pertenecientes al tipo compatible con la clase a la que pertenece el trabajo i .

A continuación se identifican los posibles movimientos de intercambio entre cada trabajo $i \hat{I}$ Solución Actual y cada trabajo $j \hat{I}$ Solución Actual. Este proceso de búsqueda puede escribirse de la siguiente forma:

" $i \hat{I}$ Solución Actual y " $j \hat{I}$ Solución Actual, Movimiento $i \ll j$ Admisible si:

$$\sum_{k \in \text{Sol. Actual} \neq j} \text{Solapamiento}(i, k) \times q$$

