Capítulo 8

Scatter Search y Path Relinking

8.1 Scatter Search

Scatter Search o búsqueda esparcida, es un algoritmo poblacional que construye soluciones mediante la combinación de otras soluciones.

Originalmente fué propuesto para combinar reglas de decisión en el contexto de programación entera.

La idea era generar nuevas reglas mediante una combinación pesada de reglas existentes. Después se extendió a una combinación pesada de restricciones para reflejar qué tanto se violaban ciertas restricciones (*surrogate constraints*).

Opera sobre un conjunto de puntos, llamados puntos de referencia (*reference points*), que constituyen "buenas" soluciones obtenidas anteriormente.

El algoritmo genera sistemáticamente combinaciones de estos puntos de referencia (soluciones) para obtener nuevos puntos (soluciones).

Las combinaciones son formas generalizadas de combinaciones lineales, acompañadas de procesos adaptativos para garantizar condiciones de factibilidad.

El algoritmo prosigue de la siguiente forma (ver tabla 8.1):

1. Genera un conjunto inicial de soluciones que garantice cierto nivel de diversidad. Selecciona un subconjunto de las mejores soluciones, en cuanto a evaluación de función objetivo y en cuanto a diversidad.

2. Crea nuevas soluciones mediante combinaciones de subconjuntos de las soluciones de referencia. Se buscan soluciones tanto dentro como fuera de las regiones convexas de las soluciones de referencia y su posible modificación para hacerlas aceptables (soluciones factibles).

3. Extrae las mejores soluciones y añádelas al conjunto de soluciones de referencia.

El proceso consiste de 5 métodos:

1. Generación de diversificación: genera una colección de soluciones diversas. Estas son típicamente 10 veces más que el conjunto de referencia.

2. Mejoramiento: transforma una solución en otra u otras mejores. Puede ser porque se violen ciertas restricciones.

3. Actualización del conjunto de referencia: construye y mantiene un conjunto con las N mejores soluciones (donde N típicamente es pequeño, alrededor de 20, y el concepto de mejor no es solo en términos de evaluación de la función objetivo, sino también en términos de diversidad).

4. Generación de un subconjunto: considerando el conjunto de referencia, generar un conjunto de posibles soluciones combinadas (lo más común es considerar todos los pares en las soluciones de referencia).

5. Combinación: transformar el subconjunto generado en una o más soluciones combinadas. Cada vez que se genera una nueva solución combinada, si ésta es mejor que la peor del conjunto de referencia, se reemplaza y se vuelven a generar los sobconjuntos necesarios.

La figura 8.1 muestra un ejemplo en donde el conjunto inicial de soluciones de referencias es {A, B, C}. Después de una combinación de A y B se produce

Tabla 8.1: Algoritmo de Scatter Search.

P = *Diversifación* para constuir $|P|$ soluciones
Sea *ConjRef* N soluciones diversas de P
Ordena las soluciones en *ConjRef* de acuerdo a la función objetivo
Sea *SolsNuevas* = True
while *SolsNuevas* **do**
    Sea *NuevosConj* todos los pares en *ConjRef* que incluya
    al menos una solución nueva.
    Sea *SolsNuevas* = False
    while *NuevosConj* $\neq \emptyset$ **do**
        Selecciona el siguiente subconjunto s en *NuevosConj*
        Sea x = Combinación(s)
        if ($x \notin$ *ConjRef* $\wedge f(x) < f(x_N)$ **then**
            $x_N = x$ y reordena *ConjRef*
            sea *SolsNuevas* = True
        endif
        elimina a s de *NuevosConj*

procedimiento Diversificación
$P \leftarrow \emptyset$
Construye una solución x
Until $|P|$ = *TamañoP*
    If $x \notin P$, **Then** $P = P \cup x$
    Else elimina x
regresa P

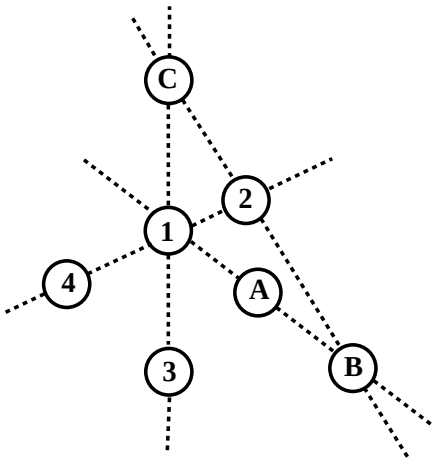

Figura 8.1: Ejemplo de generación de nuevos puntos con *scatter search*.

1 (de hecho se puede producir un segmento de línea, y de ahí seleccionar a 1). De la misma forma se generan las nuevas soluciones 2, 3 y 4.

Dependiendo de la implementación el número de combinaciones del conjunto referencia puede ser muy grande.

8.2 Path Relinking

El proceso de generar combinaciones de soluciones lineales de soluciones de referencia se puede ver como generar caminos entre soluciones que nos lleven a puntos entre o más alla de las soluciones.

Podemos llevar esto a un concepto un poco más general de creación de combinaciones de soluciones.

Un camino entre soluciones en un espacio vecino va a generar soluciones que compartan una serie de atributos con las soluciones originales. Estos atributos pueden ser ligas, nodos, valores de variables, etc.

La idea de *path relinking* es buscar soluciones que compartan atributos con antiguas soluciones con la esperanza de obtener mejores soluciones.

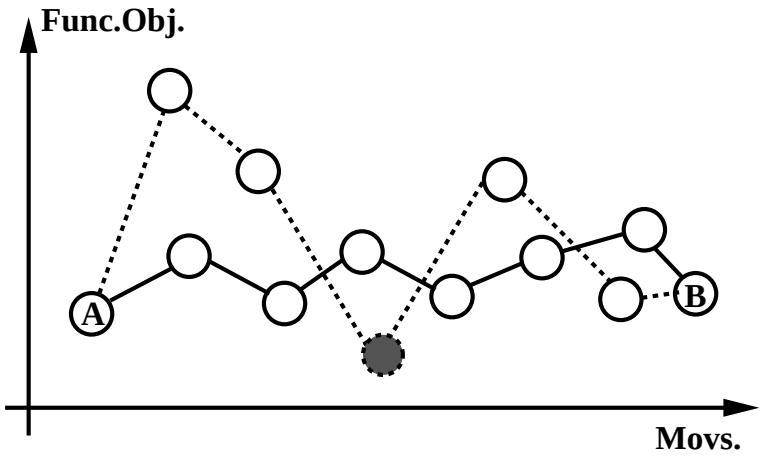

Figura 8.2: Ejemplo de *path relinking* donde la línea continua es un camino entre soluciones y la línea punteada es un camino generado para unir las dos soluciones. El nodo sombreado es una mejor solución que las anteriores.

Path relinking construye nuevas soluciones explotando las trayectorias que conectan a las soluciones buenas, empezando con alguna de las soluciones, llamada la solución de iniciación (*initiating solution*), y generando un camino en la vecindad del espacio que lleva a otras soluciones, llamadas las soluciones guías (*guiding solutions*).

Muchas veces se introducen ideas de búsqueda Tabú para poder realizar estos caminos y no repetir caminos anteriores.

La figura 8.2 muestra un ejemplo de *path relinking* donde a partir de un camino inicial entre dos soluciones (A y B) en línea continua se construye un nuevo camino para unir estas dos soluciones. La idea es encontrar soluciones difíciles de obtener por búsqueda "ciega", al incorporar atributos de soluciones conocidas.

También es posible generar caminos considerando la combinación de atributos de varias soluciones (*multiparent path*) o hacer "tunelaje" (*tunneling*) al tratar de unir dos soluciones en vecindades diferentes.

Path relinking se ha utilizado para mejorar los procedimientos de re-inicio, como por ejemplo, GRASP.

En GRASP la construcción de una solución es independiente de la construcción de la siguiente solución. No se guarda una historia.

Se han propuesto dos posibles aplicaciones de *path relinking* en combinación con GRASP:

- utilizar *path relinking* como un proceso de optimización posterior a todos los pares de soluciones "elite" encontradas con GRASP.
- utilizar *path relinking* como una estrategia de intensificación aplicada a cada solución obtenida después de la fase de búsqueda local.

Las dos estrategias mantienen un conjunto pequeño de soluciones "elite" seleccionadas después de cada proceso de búsqueda local.

Aplicar *path relinking* después de búsqueda local, en general ha dado mejores resultados. En este caso se toma la solución X de búsqueda local y una solución Y seleccionada aleatoriamente de las soluciones "elite".

Se obtienen los conjuntos de movimientos que podrian aplicarse para llegar de una solución inicial a la solución guía.

Se toma la mejor solución en esta trayectoria y se considera como candidatra a ingresar a las soluciones "elite".

Existe en esta y en muchos otras estrategias para resolver problemas de optimización propuestas para paralilizar estos algoritmos.