```
Grafo dirigido simple
                                             Situaciones posibles en el problema
                                            Inicial
                                                        En el que comienza la búsqueda
                                            Meta Cada uno de los estados solución
                                            Heurístico Información asociada a cada nodo para facilitar la búsqueda del estado meta óptimo
Espacio de búsqueda
                                                 Acción elemental que permite cambiar de estado
                             Operadores
                                                Estado origen - Estado destino - Coste
                                                           n Factor de ramificación
                                                                                                  Nº medio de sucesores de los nodos del árbol de búsqueda
                              Tamaño del problema
                                                           p Profundidad de la solución Nº de arcos desde el nodo inicial hasta la solución, que se supone única
                            Determina el orden en el que se recorre el espacio de búsqueda
                                                  Completitud Siempre encuentra solución
                                                 Admisibilidad Siempre encuentra una solución óptima
                                                               Variante del AGBG que usa una función de evaluación f(n) que indica lo prometedor que es el nodo
                                                              f(n) Función heurística de evaluación
                                                            ABIERTA Se ordena por f(n)
                            A Primero el mejor (BF)

√ Completo (en grafos finitos)

                                                              X No admisible
                                     Especializción de BF
                                                                     f* = coste del camino más corto desde el nodo inicial a los nodos objetivos, condicionado a pasar por n
                                                                  g* = coste del camino más corto que existe desde el nodo inicial a n
                                    f^*(n) = g^*(n) + h^*(n)
                                                                   h* = coste del camino más corto que existe desde el nodo objetivo más cercano a n
                                                                   C* = f*(inicial) = h*(inicial) = coste de la solución óptima
                                                                f = estimación de f*. Criterio para ordenar ABIERTA
                                                              g = coste del camino desde el nodo inicial a n
                                                                                                                                                                     nótona si V(n,n'), h(n) \le k(n,n') + h(n') (k(n,n') es el menor coste para ir de n a n')
                                    f(n) = g(n) + h(n)
                                                                                                                                                            es consistente si V(n,n'), h(n) \le c(n,n') + h(n') (c(n,n') es el coste de la regla que lleva de n a n', \infty si no existe)
                                                               h = función heurística que estima el coste para llegar a meta pasando por n
                                                                                                                                                             todo heurístico monótono es admisible
                                     Condición necesaria para expandir n es que f(n) \le C^*
                                                                                                     Si q no está en TABLA_A, se mete. Se marca n como mejor padre, se calcula g(q) = g(n) + k(n,q), y es introducido en ABIERTA según f(q)
                                                                                                    Si q estaba en TABLA_A y su lista de hijos es vacía (no expandido), rectificar g(q). Si es así, reorientar al nuevo padre en TABLA_A, y evaluar un posible cambio de orden en ABJERTA.
                                     Al expandir n y generar cada uno de sus hijos (q)
                                                                                                                                                                                          Si hay que rectificar, habrá que evaluar recursivamente en descendientes para actualizar TABLA_A, y de ser necesario, cambiar orden en ABIERTA
                                                                                                                                                                                           rectificar g(q)??
                                                                                                    Si q estaba en TABLA_A y su lista de hijos no es vacía (ya expandido)
                                     (i) Completo (en grafos localmente finitos)
                                     Admisible (siempre que h sea admisible: Vn, h(n) ≤ h*(n))
                                                                                            Cuando A* dedica demasiado tiempo discriminando caminos cuyos costes no varían significativamente, 
se puede ganar en eficiencia a costa de perder en admisibilidad
                                                                                                                             f(n) = (1-w)g(n) + w \cdot h(n), 0 \le w \le 1
                                                                                                                             w = 0 ≡ coste uniforme
                                                                                                                             w = 1/2 ≡ A* estándar
                                                                                                                             w = 1 \equiv BF
                                                                                                                             El mejor w se obtiene de forma experimental
                                                                                                                             Un algoritmo es ε-admisible si siempre encuentra una solución con un coste ≤ (1+ ε)C*
                                                                                                                                                               los pesos cambian a lo largo del proceso de búsqueda
                                                                                                                                                                h tiene mayor peso al principio de la búsqueda. En las proximidades de la solución se pasa a búsqueda en anchura
                                     Relajación de condiciones de optimalidad
                                                                                                                                                              f(n) = g(n) + h(n) + \epsilon(1 - d(n) / N)h(n)
                                                                                                                             Ponderación dinámica
                                                                                                                                                               d(n) = profundidad del nodo n
                                                                                                                                                              N = cota superior de profundidad de la mejor solución
                                                                                            Algoritmos ε-admi
                                                                                                                                                             Si h es admisible, la ponderación dinámica es ε-admisible
                                                                                                                                           Utilizan una función heurística adicional para seleccionar nodos de una sublista de ABIERTA (nodos con f \leq (1+\epsilon) min(f en ABIERTA) llamada FOCAL
                                                                                                                                           Estrategia idéntica que A*, salvo que se desarrollan primero los nodos de FOCAL según los valores del segundo heurístico h'
                                                                                                                                         h' = estimación del esfuerzo computacional para llegar desde n a la solución
                                                                                                                                         Si h es admisible, Αε* es ε-admi
                                                                                      Iterative Deepening A* (1985)
Estragia de control
                                                                                     Amplia búsqueda en profundidad iterativa
                                                                                     En cada iteración, coste límite para la búsqueda (min(f) de los nodos descartados)
                                                                                     ABIERTA pila

√ Completo

                                                                                     (i) Admisible (si h es adm
                                                                                     Coste temporal O(n P)
                                                                                                           O(n • p)
                                     Conservación de memoria
                                                                                      Limita el tamaño de TABLA_A a un valor prefijado
                                                                                     Si hay que añadir un nodo a TABLA_A y no hay espacio, se elimina de ABIERTA y de TABLA_A el nodo con mayor f en ABIERTA
                                                                                     Recuerda en cada nodo, el mejor f de los hijos olvidados de ese nodo
                                                                                     ABIERTA lista ordenada según f
                                                                                     (i) Completo (si la memoria disponible es suficiente para almacenar el camino a la solución menos profunda)
                                                                                     Admisible (si tiene suficiente memoria para almacenar el camino a la solución menos profunda)
                                               Toman decisiones de forma irrevocable en la exploración, sin considerar alternativas
                                              X No completos
                           / Voraces
                                               X No admisibles
                                              Son muy eficientes por lo que se suelen usar en tiempo real
                                                        Cada estado es un subconjunto de soluciones del problema original
                                                        El nodo raíz representa todas las posibles soluciones al problema original
                                                        Un nodo hoja contiene una solución
                          Ramificación y poda
                                                     Cada nodo no terminal tendrá asociado un heurístico que es cota inferior del coste de la mejor solución
                                                        Si un nodo rebasa el menor de los costes de los nodos hoja encontrados, el nodo se poda
                                                        La estrategia de control acostumbra a ser en profundidad por ahorro de memoria, pero valen otras
                                                       (i) Admisible (si recorre el espacio de búsqueda (salvo lo podado) hasta vaciar ABIERTA)
                                                   Se usa si el nodo meta tiene toda la información necesaria para la solución
                                                   Realiza cambios del estado actual para ir a un estado vecino para acabar en un estado meta tras una serie de iteraciones
                                                   A los estados se les asocia una función objetivo
                                                                              Para calcular los estados vecinos y decidir a cual moverse
                                                                             Criterio de aceptación Para decidir si hay que moverse
                                                                                                              Cuando se cumple, se da por finalizada la búsqueda
                                                                              Selección aleatoria entre los vecinos del nodo actual
                                                                              El criterio de aceptación permite admitir con cierta probabilidad, transiciones a estados que empeoren el valor de la función objetivo. Esta probabilidad depende de la "temperatura" y el "incremento de energía"
                                                                             Al principio la "temperatura" tiene un valor alto y va decreciendo a lo largo del proceso
La probabilidad de llegar a una buena solución es mayor si la temperatura inicial es alta y el enfriamiento
es lento
                                                  Temple simulado
                          Búsqueda local
                                                                             Requiere un ajuste apropiado de parámetros (T y enfriamiento) para que resulte eficaz y obtenga una buena solución
                                                                              El criterio de aceptación es que la solución vecina sea mejor o igual que la solución actual
                                                                                                 Óptimos locales Si se alcanza uno, el algoritmo finaliza sin encontrar el óptimo global
                                                  Máximo gradiente
                                                                                                Regiones planas
                                                                                                                      La búsqueda es aleatoria porque ningún vecino mejora al actual
                                                                                              Crestas
                                                                                                     mo de memoria para evitar la generación de algunos vecin
                                                                                            Criterio de aspiración

Aquel por el que se aplican excepciones (p. ej. admitir nodos que mejoren al mejor nodo encontrado hasta el momento)
                                                   Búsqueda tabú
                                                                          Lista tabú
                                                                                            Requiere un ajuste apropiado de parámetros según el problema (tamaño de lista tabú, criterio de
                                                                                            construcción, criterio de aspiración, ...)
```

Búsquedas informadas