

Consulte debates, estadísticas y perfiles de autores de esta publicación en: <https://www.researchgate.net/publication/228346477>

Búsqueda tabú

Libro · Julio 2008

DOI: 10.1007/978-1-4615-6089-0

CITAS

1.767

LECTURAS

8.660

3 autores:



Fred Glover

Universidad de Colorado en Boulder

623 PUBLICACIONES 51.335 CITAS

VER EL PERFIL



manuel laguna

Universidad de Colorado en Boulder

190 PUBLICACIONES 16.242 CITAS

VER EL PERFIL



Rafael Martí

universidad de valencia

226 PUBLICACIONES 11.212 CITAS

VER EL PERFIL

Principios de la búsqueda tabú

FRED GLOVER, MANUEL LAGUNA Escuela

de Negocios de Leeds, Universidad de Colorado, Campus Box 419, Boulder, CO 80309

Fred.Glover@colorado.edu, Laguna@colorado.edu

RAFAEL MARTÍ Dpto.

de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Valencia, Dr.

Moliner 50, 46100 Burjassot (Valencia) España. Rafael.Marti@uv.es

Abstracto

La búsqueda tabú es una metaheurística que guía un procedimiento de búsqueda heurística local para explorar el espacio de la solución más allá del óptimo local. Uno de los componentes principales de Tabu Search es el uso de memoria adaptativa, que crea un comportamiento de búsqueda más flexible. Las estrategias basadas en la memoria son, por lo tanto, el sello distintivo de los enfoques de búsqueda tabú, fundados en la búsqueda de "principios integradores", mediante los cuales se combinan apropiadamente formas alternativas de memoria con estrategias efectivas para explotarlas. Un hallazgo novedoso es que tales principios son a veces lo suficientemente potentes como para producir un comportamiento eficaz de resolución de problemas por sí solos, con una dependencia insignificante de la memoria. Sin embargo, en una amplia gama de escenarios de problemas, el uso estratégico de la memoria puede marcar diferencias dramáticas en la capacidad para resolver problemas. Los enfoques de búsqueda tabú pura e híbrida han establecido nuevos récords en la búsqueda de mejores soluciones a problemas de planificación y programación de la producción, asignación de recursos, diseño de redes, enrutamiento, análisis financiero, telecomunicaciones, planificación de carteras, gestión de la cadena de suministro, modelado basado en agentes y diseño de procesos comerciales. , previsión, aprendizaje automático, minería de datos, biocomputación, diseño molecular, gestión forestal y planificación de recursos, entre muchas otras áreas.

Palabras clave: resolución inteligente de problemas, estructuras de memoria, programación de memoria adaptativa.

1. Introducción

El término búsqueda tabú fue acuñado en el mismo artículo que introdujo el término metaheurístico (Glover, 1986). La búsqueda tabú (TS) se basa en la premisa de que la resolución de problemas, para calificar como inteligente, debe incorporar memoria adaptativa y exploración receptiva. La función de memoria adaptativa de TS permite implementar procedimientos que sean capaces de buscar el espacio de solución de forma económica y efectiva. Dado que las elecciones locales están guiadas por la información recopilada durante la búsqueda, TS contrasta con los diseños sin memoria que dependen en gran medida de procesos semialeatorios que implementan una forma de muestreo. El énfasis en la exploración responsiva (y por tanto en el propósito) en la búsqueda tabú, ya sea en una implementación determinista o probabilística, se deriva de la suposición de que una mala elección estratégica a menudo puede generar resultados. Más información que una buena elección aleatoria.

TS se puede aplicar directamente a prácticamente cualquier tipo de problema de optimización. Podemos plantear la mayoría de estos problemas de la siguiente forma, donde "optimizar" significa minimizar o maximizar:

$$\begin{array}{l} \text{Optimizar } f(x) \\ \text{sujeto a } x \\ X \end{array}$$

La función $f(x)$ puede ser lineal, no lineal o incluso estocástica, y el conjunto X resume las restricciones sobre el vector de variables de decisión x . De manera similar, las restricciones pueden incluir desigualdades lineales, no lineales o estocásticas, y pueden obligar a todos o algunos de los componentes de x a recibir valores discretos.

Si bien esta representación es útil para discutir una serie de consideraciones de resolución de problemas, enfatizar que en muchas aplicaciones de optimización combinatoria, el problema de interés puede no ser formulada fácilmente como una función objetiva sujeta a un conjunto de restricciones. El requisito $x \in X$ para , Por ejemplo, pueden especificar condiciones lógicas o interconexiones que serían difíciles de formular matemáticamente, pero que sería mejor dejarlas como estipulaciones verbales que luego pueden codificarse como reglas.

La técnica TS se está convirtiendo rápidamente en el método de elección para diseñar procedimientos de solución para problemas difíciles. Problemas de optimización combinatoria. Un examen exhaustivo de esta metodología se puede encontrar en el libro de Glover y Laguna (1997). Los éxitos generalizados en aplicaciones prácticas de optimización han estimulado un rápido crecimiento del método como medio para identificar de manera eficiente soluciones de muy alta calidad. Los métodos TS también se han utilizado para crear procedimientos híbridos con otros métodos heurísticos y algorítmicos, para proporcionar soluciones mejoradas a problemas de programación. secuenciación, asignación de recursos, planificación de inversiones, telecomunicaciones y muchas otras áreas. Alguno En la Tabla 1 se muestra la diversidad de aplicaciones de búsqueda tabú.

TABLA 1. APLICACIONES ILUSTRATIVAS DE BÚSQUEDA TABÚ	
<p>Planificación</p> <p>Fabricación de celdas de tiempo de flujo</p> <p>Programación de procesadores heterogéneos</p> <p>Planificación de la fuerza laboral</p> <p>Listado</p> <p>Programación de máquinas</p> <p>Programación del taller de flujo</p> <p>Programación de taller de trabajo</p> <p>Secuenciación y procesamiento por lotes</p> <p>Diseño</p> <p>Diseño asistido por ordenador</p> <p>Redes tolerantes a fallas</p> <p>Diseño de redes de transporte</p> <p>Planificación del espacio arquitectónico</p> <p>Coherencia del diagrama</p> <p>Diseño de red de carga fija</p> <p>Problemas de corte irregular</p> <p>Planificación del diseño</p> <p>Lógica e Inteligencia Artificial</p> <p>Máxima Satisfabilidad</p> <p>Lógica probabilística</p> <p>Reconocimiento/clasificación de patrones</p> <p>Procesamiento de datos</p> <p>Agrupación</p> <p>Discriminación estadística</p> <p>Entrenamiento de redes neuronales</p> <p>Diseño de redes neuronales</p> <p>Ubicación y asignación</p> <p>Ubicación/asignación de productos múltiples</p> <p>Asignación cuadrática</p> <p>Semiasignación cuadrática</p> <p>Asignación generalizada multinivel</p> <p>Problemas de brechas a gran escala</p> <p>Tecnología</p> <p>Inversión sísmica</p> <p>Distribución de energía eléctrica</p> <p>Ingeniería de Diseño Estructural</p> <p>Elipsoides de volumen mínimo</p> <p>Construcción de la estación espacial</p> <p>Colocación de celdas del circuito</p> <p>Exploración petrolera costa afuera</p>	<p>Telecomunicaciones</p> <p>Enrutamiento de llamadas</p> <p>Embalaje de ancho de banda</p> <p>Ubicación de las instalaciones centrales</p> <p>Asignación de ruta</p> <p>Diseño de Redes para Servicios</p> <p>Planificación de descuentos para clientes</p> <p>Arquitectura inmune al fracaso</p> <p>Redes ópticas síncronas</p> <p>Producción, Inventario e Inversión</p> <p>Gestión de la cadena de suministro</p> <p>Fabricación flexible</p> <p>Producción justo a tiempo</p> <p>MRP capacitado</p> <p>Selección de piezas</p> <p>Planificación de inventario de artículos múltiples</p> <p>Adquisición de descuento por volumen</p> <p>Optimización de la cartera de proyectos</p> <p>Enrutamiento</p> <p>Rutas para vehículos</p> <p>Enrutamiento capacitado</p> <p>Enrutamiento de ventana de tiempo</p> <p>Enrutamiento multimodo</p> <p>Enrutamiento de flota mixta</p> <p>Vendedor viajero</p> <p>Comprador ambulante</p> <p>Programación de convoyes</p> <p>Optimización de gráficos</p> <p>Partición de gráficos</p> <p>Coloración de gráficos</p> <p>Partición de camarillas</p> <p>Problemas de camarilla máxima</p> <p>Gráficos máximos del planificador</p> <p>Optimización combinatorial general</p> <p>Programación cero uno</p> <p>Optimización de carga fija</p> <p>Programación no lineal no convexa</p> <p>Redes de todo o nada</p> <p>Programación binivel</p> <p>Optimización discreta multiobjetivo</p> <p>Optimización general de enteros mixtos</p>

El énfasis de la búsqueda tabú en la memoria adaptativa permite explotar los tipos de estrategias que subyacen a lo mejor de la resolución humana de problemas, en lugar de limitarse a imitar los procesos. Se encuentran en órdenes inferiores de fenómenos y comportamientos naturales. Los elementos básicos de la búsqueda tabú tienen varias características importantes, resumidas en la Tabla 2. La búsqueda tabú se ocupa de encontrar formas nuevas y más efectivas de aprovechar los conceptos incorporados en la Tabla 2, y de identificar Principios asociados que pueden ampliar los fundamentos de la búsqueda inteligente.

TABLA 2. PRINCIPALES CARACTERÍSTICAS DE BÚSQUEDA TABÚ Memoria
<p>adaptable</p> <p>Selectividad (incluido el olvido estratégico)</p> <p>Abstracción y descomposición (a través de la memoria explícita y atributiva)</p> <p>Momento:</p> <ul style="list-style-type: none"> actualidad de los eventos frecuencia de los eventos diferenciación entre corto y largo plazo <p>Calidad e impacto:</p> <ul style="list-style-type: none"> Atractivo relativo de las opciones alternativas. Magnitud de los cambios en la estructura o restricciones. relaciones <p>Contexto:</p> <ul style="list-style-type: none"> interdependencia regional interdependencia estructural interdependencia secuencial <p>Exploración receptiva</p> <p>Restricciones e incentivos impuestos estratégicamente</p> <ul style="list-style-type: none"> (condiciones tabú y niveles de aspiración) <p>Enfoque concentrado en buenas regiones y buenas características de solución.</p> <ul style="list-style-type: none"> (procesos de intensificación) <p>Caracterizar y explorar nuevas regiones prometedoras (procesos</p> <ul style="list-style-type: none"> de diversificación) <p>Patrones de búsqueda no monótonos</p> <ul style="list-style-type: none"> (oscilación estratégica) <p>Integración y ampliación de soluciones</p> <ul style="list-style-type: none"> (revinculación de rutas)

En este capítulo describiremos algunos aspectos clave de esta metodología, como el uso de estructuras de memoria. y estrategias de búsqueda, e ilustrarlas en una implementación para resolver el problema de ordenamiento lineal.

2. Estructuras de la memoria

La búsqueda tabú comienza de la misma manera que la búsqueda local o de vecindario ordinaria, procediendo de forma iterativa desde un punto (solución) a otro hasta que se cumpla un criterio de terminación elegido. Cada solución x tiene una vecindad asociada $N(x)$

X y cada solución $x' \in N(x)$ se alcanza desde x mediante una operación llamado movimiento.

Podemos contrastar TS con un método de descenso simple donde el objetivo es minimizar $f(x)$. Este método sólo permite movimientos a soluciones vecinas que mejoran el valor de la función objetivo actual y termina cuando no se pueden encontrar soluciones mejoradas. La x final obtenida mediante un método de descenso se llama local.

óptimo, ya que es al menos tan bueno o mejor que todas las soluciones de su entorno. lo evidente
La desventaja de un método de descenso es que dicho óptimo local en la mayoría de los casos no será un óptimo global,
es decir, normalmente no minimizará $f(x)$ sobre todo $x \in X$.

La búsqueda tabú permite movimientos que deterioran el valor actual de la función objetivo, pero los movimientos se eligen de una vecindad modificada $N^*(x)$. Las estructuras de la memoria a corto y largo plazo son responsables de la composición específica de $N^*(x)$. En otras palabras, la vecindad modificada es el resultado de mantener una historia selectiva de los estados encontrados durante la búsqueda. En las estrategias TS basadas en el corto plazo. Consideraciones generales, $N^*(x)$ característicamente es un subconjunto de $N(x)$, y la clasificación tabú sirve para identificar elementos de $N(x)$ excluidos de $N^*(x)$. En las estrategias de TS que incluyen consideraciones a más largo plazo, $N^*(x)$ también puede ampliarse para incluir soluciones que normalmente no se encuentran en $N(x)$, como soluciones encontradas y evaluadas en búsquedas anteriores, o identificadas como vecinas de alta calidad de estas soluciones pasadas. . Caracterizado en De esta manera, TS puede verse como un método de vecindad dinámica. Esto significa que la vecindad de x No es un conjunto estático, sino un conjunto que puede cambiar según el historial de la búsqueda.

La estructura de una vecindad en la búsqueda tabú difiere de la utilizada en la búsqueda local de una manera adicional, al abarcar los tipos de movimientos utilizados en procesos constructivos y destructivos (donde las bases para tales movimientos se denominan, en consecuencia, vecindades constructivas y vecindades destructivas) . Estos usos ampliados del concepto de barrio refuerzan una perspectiva fundamental de TS, que consiste en definir vecindarios de manera dinámica que puedan incluir la consideración en serie o simultánea de múltiples tipos de movimientos.

TS utiliza la memoria atributiva con fines de orientación (es decir, para calcular $N^*(x)$). En lugar de registrar soluciones completas, las estructuras de memoria atributiva se basan en el registro de atributos. Este tipo de memoria registra información sobre las propiedades (atributos) de la solución que cambian al pasar de una solución a otra. Los enfoques de memoria atributiva más comunes son la memoria basada en lo reciente y la memoria basada en la frecuencia. Recency, como su nombre indica, realiza un seguimiento de los atributos de las soluciones que han cambiado durante el pasado reciente. La frecuencia generalmente consiste en proporciones sobre el número de iteraciones que tiene un determinado atributo. cambia o no (dependiendo de si se trata de una frecuencia de transición o de residencia). En las Tablas 3 y 4, respectivamente, se muestran algunos ejemplos de memoria basada en lo reciente y en la frecuencia.

Contexto	Atributos	Para grabar la última vez...
Problemas binarios	Índice de variables (i)	variable cambió su valor de 0 a 1 o de 1 a 0 (dependiendo de su valor actual).
Secuenciación de trabajos	Índice de empleo (j) el empleo j cambió de posición.	
	Índice de empleo (j) y puesto (p) puesto j ocupado puesto p.	
	Par de índices de empleo (i, j) el empleo i intercambia puestos con el empleo j.	
Graficos	Índice de arco (i) arco i se agregó a la solución actual. arco i fue eliminado de la solución actual.	

Tabla 3. Ejemplos de memoria basada en lo reciente

Contexto	Medida de residencia	Medida de transición Número
Problemas binarios	Número de veces que a la variable i se le ha asignado el valor 1.	de veces que la variable i ha cambiado de valor.
Secuenciación de trabajos	Número de veces que el trabajo j ha ocupado la posición p.	Número de veces que el trabajo i ha intercambiado puestos con el trabajo j.
	Valor medio de la función objetivo cuando el trabajo j ocupa la posición p.	Número de veces que el trabajo j se ha movido a una posición anterior en la secuencia.
Graficos	Número de veces que arco i ha sido parte de la solución actual.	Número de veces que se ha eliminado el arco i de la solución actual cuando se ha agregado el arco j .
	Valor promedio de la función objetivo cuando el arco i es parte de la solución.	Número de veces que se ha agregado el arco i durante los movimientos de mejora.

Tabla 4. Ejemplos de memoria basada en frecuencia

Característicamente, un proceso TS basado estrictamente en estrategias de corto plazo puede permitir que una solución x sea visitada más de una vez, pero es probable que la correspondiente vecindad reducida $N^*(x)$ sea diferente cada vez. Con la inclusión de consideraciones a más largo plazo, la probabilidad de duplicar un proceso anterior

vecindario al revisar una solución y, más generalmente, al tomar decisiones que visitan repetidamente solo un subconjunto limitado de X , es prácticamente inexistente.

La memoria basada en actualidad es la estructura de memoria más común utilizada en las implementaciones de TS. Como sugiere su nombre, esta estructura de memoria realiza un seguimiento de los atributos de las soluciones que han cambiado durante el pasado reciente. Para explotar esta memoria, los atributos seleccionados que ocurren en las soluciones visitadas recientemente se etiquetan como tabú activos, y las soluciones que contienen elementos tabú activos, o combinaciones particulares de estos atributos, son las que se convierten en tabú. Esto impide que ciertas soluciones del pasado reciente pertenezcan a $N^*(x)$ y, por tanto, sean revisadas. También se impide que se visiten otras soluciones que comparten dichos atributos tabú activos. Tenga en cuenta que, si bien la clasificación tabú se refiere estrictamente a soluciones cuya visita está prohibida, en virtud de contener atributos tabú activos (o, más generalmente, al violar cierta restricción basada en estos atributos), los movimientos que conducen a tales soluciones también suelen denominarse como tabú.

La memoria basada en frecuencia proporciona un tipo de información que complementa la información proporcionada por la memoria basada en lo reciente, ampliando la base para seleccionar los movimientos preferidos. Al igual que lo reciente, la frecuencia a menudo se pondera o descompone en subclases. Además, la frecuencia se puede integrar con lo reciente para proporcionar una estructura compuesta para crear sanciones e incentivos que modifiquen las evaluaciones de movimientos.

Las frecuencias generalmente consisten en proporciones, cuyos numeradores representan recuentos expresados en dos medidas diferentes: una medida de transición (el número de iteraciones donde un atributo cambia (entra o sale) de las soluciones visitadas en una trayectoria particular, y una medida de residencia (el número de iteraciones) donde un atributo pertenece a soluciones visitadas en una trayectoria particular, o el número de instancias en las que un atributo pertenece a soluciones de un subconjunto particular. Los denominadores generalmente representan uno de tres tipos de cantidades: (1) el número total de ocurrencias de todos los eventos representados por los numeradores (como el número total de iteraciones asociadas), (2) la suma (o promedio) de los numeradores, y (3) el valor máximo del numerador. En los casos en que los numeradores representan recuentos ponderados, algunos de los cuales pueden ser negativos, el denominador (3) se expresa como un valor absoluto y el denominador (2) se expresa como una suma de valores absolutos (posiblemente desplazados por una pequeña constante para evitar un cero). denominador). Las proporciones producen frecuencias de transición que registran la frecuencia con la que cambian los atributos y las frecuencias de residencia.

que realizan un seguimiento de la frecuencia con la que los atributos son miembros de las soluciones generadas. Además de hacer referencia a dichas frecuencias, los umbrales basados únicamente en los numeradores pueden resultar útiles para indicar cuándo son apropiadas las fases de mayor diversificación.

3. Estrategias de búsqueda

El uso de la memoria reciente y de frecuencia en la búsqueda tabú generalmente cumple la función de evitar que los procesos de búsqueda se ciclen, es decir, que ejecuten interminablemente la misma secuencia de movimientos (o más generalmente, que revisen interminable y exclusivamente el mismo conjunto de soluciones). Sin embargo, en términos más generales, las diversas manifestaciones de estos tipos de memoria están diseñadas para impartir solidez o vigor adicional a la búsqueda.

Un elemento clave del marco de memoria adaptativa de la búsqueda tabú es crear un equilibrio entre la intensificación y la diversificación de la búsqueda. Las estrategias de intensificación se basan en la modificación de las reglas de elección para fomentar combinaciones de movimientos y características de solución que históricamente se consideran buenas. También pueden iniciar un regreso a regiones atractivas para buscarlas más a fondo. Las estrategias de diversificación, por otro lado, buscan incorporar nuevos atributos y combinaciones de atributos que no estaban incluidos dentro de las soluciones generadas previamente. De una forma, estas estrategias pretenden impulsar la búsqueda hacia regiones diferentes a las ya examinadas. Es importante tener presente que la intensificación y la diversificación no son mutuamente opuestas, sino que más bien se refuerzan mutuamente.

La mayoría de los tipos de estrategias de intensificación requieren un medio para identificar un conjunto de soluciones de élite como base para incorporar buenos atributos en soluciones recién creadas. La membresía en el conjunto de élite a menudo se determina estableciendo un umbral que está conectado con el valor de la función objetivo de la mejor solución encontrada durante la búsqueda. En la Figura 1 se muestra un ejemplo simple de la estrategia de intensificación. Dos variantes simples para la selección de soluciones de élite han demostrado ser bastante exitosas. Se introduce una medida de diversificación para garantizar que las soluciones registradas difieran entre sí en un grado deseado y luego se borra toda la memoria a corto plazo antes de continuar con la mejor de las soluciones registradas. El otro mantiene una lista secuencial de longitud acotada que agrega una nueva solución al final solo si es mejor que cualquiera vista anteriormente, y también se guarda la memoria a corto plazo que acompañaba a esta solución.

```

Aplicar TS memoria a corto plazo
Aplicar una estrategia de selección de
élite. hacer {
    Elija una de las soluciones de élite.
    Reanudar la memoria a corto plazo TS de la solución elegida.
    Agregue nuevas soluciones a la lista de élite cuando
corresponda. } while (iteraciones < límite y lista no vacía)

```

Figura 1. Enfoque simple de intensificación de TS.

La diversificación se crea automáticamente en el ST (hasta cierto punto) mediante funciones de la memoria a corto plazo, pero se ve particularmente reforzada por ciertas formas de memoria a largo plazo. Las estrategias de diversificación de TS a menudo se basan en modificar las reglas de elección para incorporar a la solución atributos que se utilizan con poca frecuencia. Alternativamente, pueden introducir dichos atributos aplicando periódicamente métodos que ensamblen subconjuntos de estos atributos en soluciones candidatas para continuar la búsqueda, o reiniciando parcial o totalmente el proceso de solución. Las estrategias de diversificación son particularmente útiles cuando sólo se pueden alcanzar mejores soluciones cruzando barreras o “jorobas” en la topología del espacio de soluciones.

La incorporación de reglas de elección modificadas se puede moderar utilizando la siguiente función de penalización:

$$\text{MoveValue} = \text{MoveValue} + d * \text{Penalización.}$$

Este tipo de enfoque de penalización se utiliza comúnmente en TS, donde el valor de la penalización es a menudo una función de medidas de frecuencia como las indicadas en la Tabla 2, y d es un parámetro de diversificación ajustable. Los valores d más grandes corresponden a un deseo de una mayor diversificación.

4. Diseños avanzados: oscilación estratégica y reenlace de trayectorias

Hay muchas formas en las que se puede mejorar una implementación simple de búsqueda tabú agregando elementos del término. En este artículo restringimos nuestra atención a dos de los métodos más utilizados, a saber, el estratégico. Oscilación y reenlace de trayectorias, que constituyen el núcleo de muchas programaciones de memoria adaptativa. algoritmos.

La oscilación estratégica opera orientando movimientos en relación con un nivel crítico, identificado por una etapa de construcción o un intervalo elegido de valores funcionales. Un nivel tan crítico o un límite de oscilación a menudo representa un punto donde el método normalmente se detendría. Sin embargo, en lugar de detenerse cuando se alcanza este límite, se modifican las reglas para seleccionar movimientos, para permitir que la región definida por el punto crítico nivel a cruzar. Luego, la aproximación continúa hasta una profundidad especificada más allá del límite de oscilación y gira. Se vuelve a acercarse y cruzar el límite de oscilación, esta vez desde la dirección opuesta, y el método avanza hasta un nuevo punto de inflexión (ver Figura 2).

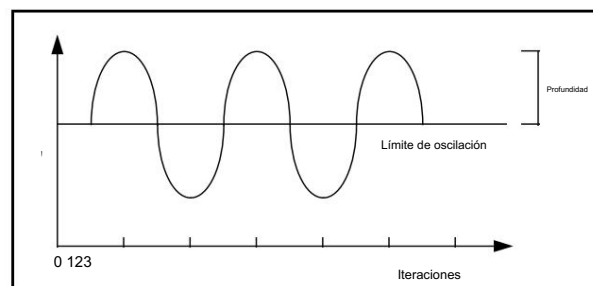


Figura 2. Oscilación estratégica

El proceso de acercarse y cruzar repetidamente el nivel crítico desde diferentes direcciones crea un comportamiento oscilatorio, que da nombre al método. El control sobre este comportamiento se establece generando evaluaciones y reglas de movimiento modificadas, dependiendo de la región navegada y la dirección de búsqueda. La posibilidad de volver sobre una trayectoria anterior se evita mediante mecanismos de búsqueda tabú estándar, como los establecidos por las funciones de memoria basadas en lo reciente y en la frecuencia.

Cuando el nivel o los valores funcionales en la Figura 2 se refieren a grados de viabilidad e inviabilidad, se puede utilizar una función con valores vectoriales asociada con un conjunto de restricciones del problema para controlar la oscilación. En este caso, controlar la búsqueda limitando esta función puede verse como manipular una parametrización del conjunto de restricciones seleccionado. Una alternativa preferida suele ser convertir la función en una función de penalización de restricción sustituta o de Lagrange, evitando funciones con valores vectoriales y permitiendo compensaciones entre grados de violación de diferentes restricciones de componentes.

Path Relinking, como estrategia para crear trayectorias de movimientos que pasan a través de soluciones de alta calidad, se propuso por primera vez en relación con la búsqueda tabú en Glover (1989). Luego, el enfoque se elaboró con mayor detalle como un medio para integrar las estrategias de intensificación y diversificación de TS, y se le dio el nombre de path relinking (PR), en Glover y Laguna (1993). Las relaciones públicas generalmente operan partiendo de una solución inicial, seleccionada de un subconjunto de soluciones de alta calidad, y generando un camino en el espacio vecinal que conduce hacia las otras soluciones del subconjunto, que se denominan soluciones guía.

soluciones. Esto se logra seleccionando movimientos que introducen atributos contenidos en las soluciones guía.

La reconexión de rutas puede considerarse una extensión del método combinado de búsqueda de dispersión (Glover y Laguna, 1993; Laguna y Martí, 2003). En lugar de producir directamente una nueva solución al combinar dos o más soluciones originales, PR genera caminos entre y más allá de las soluciones seleccionadas en el espacio de vecindad. El carácter de dichas rutas se especifica fácilmente mediante referencia a los atributos de la solución que se agregan, eliminan o modifican de otro modo mediante los movimientos ejecutados. Ejemplos de tales atributos incluyen aristas y nodos de un gráfico, posiciones de secuencia en un cronograma, vectores contenidos en soluciones básicas de programación lineal y valores de variables y funciones de variables.

El enfoque puede verse como un ejemplo extremo (altamente enfocado) de una estrategia que busca incorporar atributos de soluciones de alta calidad, creando incentivos para favorecer estos atributos en los movimientos seleccionados. Sin embargo, en lugar de utilizar un incentivo que simplemente fomente la inclusión de tales atributos, el enfoque de reconexión de caminos subordina otras consideraciones al objetivo de elegir movimientos que introduzcan los atributos de las soluciones guía, con el fin de crear una "buena composición de atributos" en el proceso. solución actual. La composición en cada paso se determina eligiendo el mejor movimiento, utilizando criterios de elección habituales, de un conjunto restringido: el conjunto de movimientos actualmente disponibles que incorporan un número máximo (o un valor ponderado máximo) de los atributos de las soluciones guía.

(Las excepciones se proporcionan según los criterios de aspiración, como se indica más adelante). El enfoque se denomina reenlace de ruta, ya sea en virtud de generar una nueva ruta entre soluciones previamente vinculadas mediante una serie de movimientos ejecutados durante una búsqueda, o generando una ruta entre soluciones previamente vinculadas. a otras soluciones pero no entre sí.

Para generar los caminos deseados, sólo es necesario seleccionar movimientos que cumplan la siguiente función: al partir de una solución iniciadora, los movimientos deben introducir progresivamente atributos aportados por una solución guía (o reducir la distancia entre los atributos de las soluciones iniciadora y guía).). Los roles de quienes inician y guían las soluciones son intercambiables; También se puede inducir a cada solución a moverse simultáneamente hacia la otra como forma de generar combinaciones. Primero considere la creación de caminos que unen dos soluciones seleccionadas x' y x'' , restringiendo la atención a la parte del camino que se encuentra 'entre' las soluciones, produciendo una secuencia de soluciones $x' = x(1), x(2), \dots, x(r) = x''$. Para reducir el número de opciones a considerar, la solución $x(i + 1)$ puede crearse a partir de $x(i)$ en cada paso eligiendo un movimiento que minimice el número de movimientos restantes para llegar a x'' . El camino reenlazado puede encontrar soluciones que tal vez no sean mejores que la solución iniciadora o guía, pero que proporcionen "puntos de acceso" fértiles para alcanzar otras soluciones, algo mejores. Por esta razón, es valioso examinar las soluciones vecinas a lo largo de un camino reenlazado y realizar un seguimiento de aquellas de alta calidad que puedan proporcionar un punto de partida para iniciar búsquedas adicionales.

Como describen Martí et al. (2004), podemos aplicar diferentes elementos de relaciones públicas para realizar diseños más elaborados. Algunos ejemplos son: reconexión simultánea, estrategia de túnel, reconexión extrapolada, múltiples soluciones de orientación, barrios constructivos o construcción de vocabulario.

5. El problema del orden lineal

Dada una matriz de pesos $E = \{e_{ij}\}_{m \times m}$, el problema de ordenamiento lineal (LOP) consiste en encontrar una permutación p de las columnas (y filas) para maximizar la suma de los pesos en la parte superior triángulo. En términos matemáticos, buscamos maximizar:

$$C_p = \sum_{i=1}^{m-1} \sum_{j=i+1}^m e_{p_i p_j}.$$

donde p_i es el índice de la columna (y fila) en la posición i en la permutación. Tenga en cuenta que en el LOP, la permutación p proporciona el orden tanto de las columnas como de las filas. Métodos de solución para este NP-duro

El problema se ha propuesto desde 1958, cuando Chenery y Watanabe esbozaron algunas ideas sobre cómo solucionar obtener soluciones para este problema (Reinelt, 1985). En esta sección describimos una implementación de búsqueda tabú (Laguna et al. 1999) para el LOP.

El LOP tiene una amplia gama de aplicaciones en varios campos. Quizás la aplicación más conocida se produce en el campo de la economía. En esta aplicación, la economía (regional o nacional) se subdivide primero en sectores. Luego, se crea una matriz de entrada/salida, en la que la entrada (i,j) representa el flujo de dinero del sector i al sector j . Los economistas suelen estar interesados en ordenar los sectores de manera que los proveedores tiendan a vienen primero, seguidos por los consumidores. Esto se logra permutando las filas y columnas de la matriz de modo que la suma de entradas por encima de la diagonal se maximice, que es el objetivo del LOP.

Las inserciones se utilizan como mecanismo principal para pasar de una solución a otra en Laguna's et al. método para el LOP. INSERT_MOVE(p, i) consiste en eliminar p_j de su posición actual j para ser insertado en la posición i (es decir, entre los sectores actuales p_{i-1} y p_i). Esta operación da como resultado el orden p' , de la siguiente manera:

$$p' = \begin{cases} (p_1, \dots, p_{i-1}, p_j, p_i, \dots, p_{j-1}, p_{j+1}, \dots, p_m) & \text{para } i < j \\ (p_1, \dots, p_{j-1}, p_i, p_{j+1}, \dots, p_j, p_{j+1}, \dots, p_m) & \text{para } i > j \end{cases}$$

La vecindad N consta de todas las permutaciones resultantes de la ejecución de movimientos de inserción generales como:

$$N = \{p: \text{INSERT_MOVE}(p, i), \text{ para } j = 1, \dots, m, i = 1, 2, \dots, j-1, j+1, \dots, m\},$$

y N se divide en m N_j barrios asociados a cada sector p_j , para $j = 1, \dots, m$.

$$N_j = \{p: \text{INSERT_MOVE}(p, i), i = 1, 2, \dots, j-1, j+1, \dots, m\}$$

A partir de una permutación p generada aleatoriamente, el procedimiento básico de TS alterna entre una fase de intensificación y una de diversificación. Una iteración de la Fase de Intensificación comienza aleatoriamente seleccionando un sector. La probabilidad de seleccionar el sector j es proporcional a su peso w_j según:

$$w_j = \frac{f(p_j)}{\sum_{j=1}^m f(p_j)}$$

Se selecciona el movimiento INSERT_MOVE(p, i) con el valor de movimiento más grande. (Tenga en cuenta que esta regla puede resultar en la selección de un movimiento que no mejora.) El movimiento se ejecuta incluso cuando el valor del movimiento no es positivo, lo que resulta en un deterioro del valor de la función objetivo actual. El sector movido se convierte tabu-activo para las iteraciones de TabuTenure y, por lo tanto, no se puede seleccionar para inserciones durante este tiempo.

El número de veces que se ha elegido mover el sector j se acumula en el valor $\text{freq}(j)$. Esta información de frecuencia se utiliza con fines de diversificación. La fase de intensificación termina después de MaxInt iteraciones consecutivas sin mejora. Antes de abandonar esta fase, se aplica un procedimiento de búsqueda local en el mismo barrio de la mejor solución encontrada (durante el periodo actual). Denotamos esta solución como $p^\#$, en contraste con p^* (la mejor solución encontrada durante toda la búsqueda). Al aplicar este procedimiento codicioso (sin restricciones tabú), se garantiza un óptimo local como resultado de la fase de intensificación.

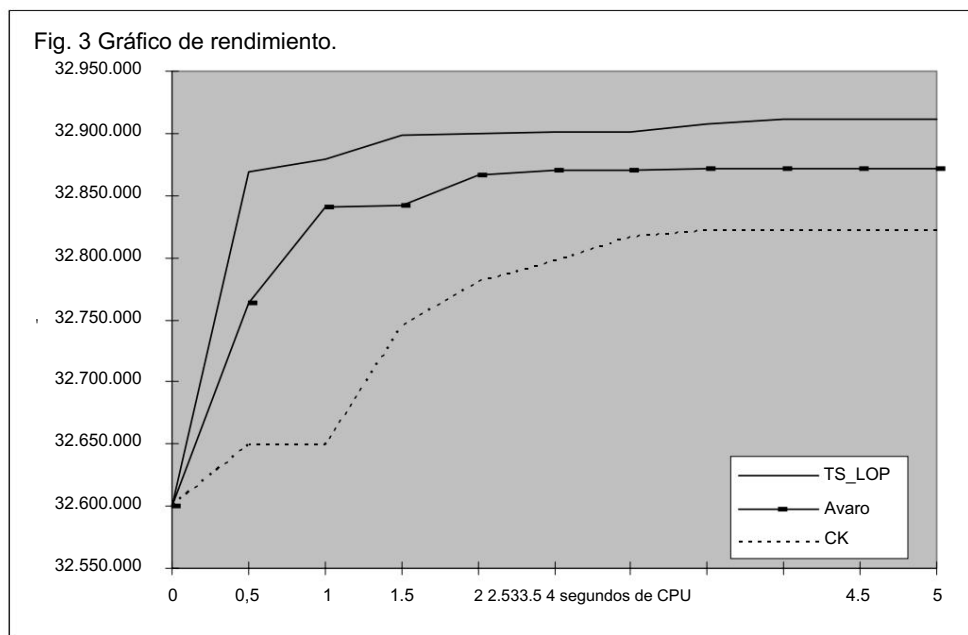
La Fase de Diversificación se realiza para las iteraciones MaxDiv . En cada iteración, se selecciona aleatoriamente un sector, donde la probabilidad de seleccionar el sector j es inversamente proporcional al conteo de frecuencia. frecuencia(j). El sector elegido se coloca en la mejor posición, según lo determinado por los valores de movimiento asociados con los movimientos de inserción en N_j . El procedimiento se detiene cuando se realizan iteraciones globales de MaxGlo sin mejorando $CE(p^*)$. Una iteración global es una aplicación de la fase de intensificación seguida de la aplicación de la fase de diversificación.

Se introduce una intensificación adicional mediante la implementación de una fase de reconexión de rutas a largo plazo. Específicamente, la mejor solución encontrada al final de una fase de intensificación $p\#$ (que no necesariamente representa p^* , la mejor solución en general) se somete a un proceso de reenlace. El proceso consiste en hacer movimientos comenzando desde $p\#$ (la solución inicial) en dirección a un conjunto de soluciones de élite (también denominadas soluciones guía). El conjunto de soluciones de élite consta de las mejores soluciones de ElitSol encontradas durante toda la búsqueda. Las inserciones utilizadas para acercar la solución iniciadora a las soluciones guía se pueden describir de la siguiente manera. Para cada sector p_j en la solución actual:

- 1) Encuentre la posición i para la cual se minimiza el valor absoluto de (j_i) , donde i es la posición que ocupa p_j en al menos una de las soluciones guía.
- 2) Realice INSERT_MOVE(p_j, i).

También se implementa una fase de diversificación a largo plazo para complementar la fase de diversificación en el procedimiento básico. La diversificación a largo plazo se aplica después de que las iteraciones globales de MaxLong hayan transcurrido sin mejorar $CE(p^*)$. Para cada sector p_j , se calcula una posición promedio redondeada $\alpha(p_j)$ utilizando las posiciones ocupadas por este sector en el conjunto de soluciones de élite y las soluciones visitadas durante la última fase de intensificación. Luego, se realizan m pasos de diversificación que insertan cada sector p_j en su posición complementaria $m - \alpha(p_j)$, es decir, INSERT_MOVE($p_j, m - (p_j)$) se ejecuta para $j = 1, \dots$, metro.

Después de la experimentación preliminar, los parámetros de búsqueda se establecen en MaxGlo = 100, MaxLong = 50, ElitSol = 4, TabuTenure = 2 MaxLong, my MaxDiv = 0,5 my ElitSol = 4. En las 49 instancias de la biblioteca LOLIB de dominio público, el método obtiene la solución óptima dentro de 1 segundo de tiempo de ejecución de la computadora en un Pentium IV a 3 Ghz . El método también se compara con un procedimiento anterior debido a Chanas y Kobylanski (1996) y un procedimiento codicioso basado en la búsqueda local N. Los métodos se ejecutaron de una manera que la mejor solución encontrada se informó cada 0,5 segundos. Estos puntos de datos se utilizaron para generar el gráfico de rendimiento en la Figura 3. El rendimiento superior de TS_LOP se hace evidente en la Figura 3.



6. El ciclo tabú y los métodos de probabilidad condicional

En esta sección, describimos la implementación y prueba del método del ciclo tabú y dos variantes del método de probabilidad condicional (Laguna, 2005). Estos métodos se describieron originalmente en Glover (1990) y nuevamente en el libro de Glover y Laguna (1997), pero han sido ignorados en gran medida en la literatura de búsqueda tabú. El método del ciclo tabú es un mecanismo de memoria a corto plazo que se basa en particionar los elementos (es decir, mover atributos) de una lista tabú. La metodología es general y capaz de acomodar memorias de búsqueda tabú de atributos múltiples, como se describe en Glover y Laguna (1997). En su forma más básica, el método del ciclo tabú divide la lista de memoria a corto plazo en grupos TabuGroups, donde el grupo k consta de elementos que se agregaron a la lista hace un rango específico de iteraciones.

Mientras que en algunas variantes de la búsqueda tabú (por ejemplo, la búsqueda tabú probabilística) es común relajar progresivamente el estado tabú de los elementos a medida que envejecen, el método del ciclo tabú, por el contrario, permite que los elementos de algunos grupos escapen por completo de su estado tabú. según determinadas frecuencias que aumentan con la edad de los grupos. El método se basa en el uso de intervalos de iteración llamados ciclos tabú, que se hacen más pequeños para los grupos más antiguos que para los más jóvenes (con la excepción de un pequeño grupo de amortiguación). Específicamente, si el grupo k tiene un ciclo tabú de iteraciones $TC(k)$, entonces en cada ocurrencia de tantas iteraciones, en promedio, los elementos del grupo k escapan de su estado tabú y son libres de ser elegidos. En otras palabras, en promedio, el grupo k se designa como LIBRE en cada iteración de $TC(k)$. Los mecanismos y estructuras de datos que son útiles para lograr esto se describen en Laguna (2005).

El método de probabilidad condicional es una variante del método del ciclo tabú que elige elementos estableciendo la probabilidad de que un grupo sea LIBRE en una iteración determinada. La probabilidad asignada al grupo k puede verse conceptualmente como la inversa del valor del ciclo tabú. Es decir, $P(k) = 1/TC(k)$.

De manera análoga al método del ciclo tabú, el grupo k es LIBRE si todos los grupos anteriores también lo son. El método emplea una probabilidad condicional, $CP(k)$, como medio para determinar si un grupo k particular puede designarse como LIBRE. Los valores de probabilidad condicional son fijos y en cada iteración el estado de un grupo está determinado por un proceso probabilístico que no se ve afectado por elecciones anteriores. En consecuencia, el enfoque ignora la posibilidad de que los valores reales del ciclo tabú puedan estar lejos de sus objetivos para algunos grupos. Esto puede suceder, por ejemplo, cuando durante varias iteraciones no se elige ningún elemento de un conjunto particular de grupos LIBRES. El método de probabilidad condicional también utiliza un grupo de buffer, para el cual ningún elemento puede escapar de su estado tabú.

Una variante del método de probabilidad condicional utiliza valores de probabilidad sustitutos para mantener el número esperado de elementos por iteración elegidos de grupos que no sean más antiguos que cualquier grupo k determinado cerca de $P(k)$. Las probabilidades sustitutas reemplazan los valores originales de $P(k)$ en la determinación de las probabilidades condicionales. Estas probabilidades sustitutas utilizan conteos cíclicos, que también se utilizan en relación con el método del ciclo tabú.

Laguna (2005) utiliza un problema de programación de una sola máquina para probar el mérito de las implementaciones del método del ciclo tabú y ambas variantes del método de probabilidad condicional. El problema consiste en minimizar la suma de los costos de preparación y las penalizaciones por retraso lineal cuando se programan n trabajos, que llegan al tiempo cero, para su procesamiento secuencial en una máquina continuamente disponible. En la literatura se han informado varias variantes de búsqueda tabú para este problema (Laguna, Barnes y Glover, 1991 y 1993; Glover y Laguna, 1991; Laguna y Glover, 1993). Se realizaron experimentos con más de 300 instancias de problemas con hasta 200 trabajos para comparar esquemas de memoria a corto plazo estáticos y dinámicos simples con una implementación de ciclo tabú (Cycle), una implementación de probabilidad condicional (C-Prob) y una implementación del condicional. método de probabilidad con probabilidades sustitutas (S-Prob). La memoria estática a corto plazo asigna una duración tabú constante a todos los atributos durante la búsqueda. La memoria dinámica a corto plazo selecciona aleatoriamente una tenencia tabú de un rango específico. Por lo tanto, la tenencia tabú asignada a un atributo en una iteración determinada puede no ser la misma que la tenencia tabú asignada a otro atributo en una iteración diferente. La Tabla 5 muestra el número de mejores soluciones encontradas por cada método en cada conjunto de 100 problemas.

Conjunto de	Estático	Dinámico	Ciclo	Problema	Problema
	2	50		C	S
problemas	0	10	9	31	65
n = 50 n = 100 n = 200	0	8	28 37	17 26	47 29

Tabla 5. Número de mejores soluciones (de 100) encontradas por cada método

Los resultados de la Tabla 5 muestran el mérito del ciclo tabú y las variantes de probabilidad condicional a medida que aumenta el tamaño del problema. Además de estos resultados, el S-Prob es capaz de encontrar 17 nuevas mejores soluciones para 20 problemas utilizados para la experimentación en Glover y Laguna (1991). Para problemas con hasta 60 puestos de trabajo, para los cuales se puede calcular un límite inferior, S-Prob produce una brecha máxima del 3,56% en relación con este límite optimista.

Estos resultados confirman que un procedimiento de búsqueda tabú basado únicamente en una lista tabú estática no es un método robusto, porque es incapaz de mantener un nivel aceptable de diversidad durante la búsqueda. La memoria dinámica de corto plazo sigue siendo una alternativa atractiva porque es fácil de implementar y proporciona un buen equilibrio entre diversificación e intensificación. Los resultados también muestran que es posible obtener mejores resultados con el esfuerzo adicional requerido para implementar el ciclo tabú o los métodos de probabilidad condicional.

Las estrategias adicionales identificadas en Glover y Laguna (1997) pueden ser valiosas para explotar otros aspectos de la intensificación y la diversificación, pero este ejemplo demuestra la importancia de manejar la memoria a corto plazo de una manera estratégica, especialmente cuando se enfrentan problemas más grandes y difíciles.

Conclusiones

El enfoque y énfasis de la búsqueda tabú tienen una serie de implicaciones para el objetivo de diseñar procedimientos de optimización mejorados. Estas oportunidades de investigación conllevan un énfasis en producir reglas sistemáticas y estratégicamente diseñadas, en lugar de seguir la política de relegar las decisiones a elecciones aleatorias, como suele estar de moda en los métodos evolutivos. Los resultados altamente atractivos proporcionados por las estructuras de memoria adaptativa que subyacen a la búsqueda tabú están produciendo un impacto evidente en el diseño de métodos metaheurísticos en general, y están motivando la aparición de nuevos híbridos de TS con otros procedimientos.

Expresiones de gratitud

Esta investigación está parcialmente financiada por el Gobierno español bajo los códigos TIC2002-10886E y TIC2003-C05-01.

Referencias

- Chanas, S. y P. Kobyanski (1996) "Un nuevo algoritmo heurístico que resuelve el problema de ordenamiento lineal", Optimización y aplicaciones computacionales, vol. 6, págs. 191-205.
- Glover, F. (1986) "Caminos futuros para la programación entera y vínculos con la inteligencia artificial", Computers and Operations Research, vol. 13, págs. 533-549.
- Glover, F., 1989. Búsqueda tabú, Parte I, ORSA Journal on Computing, vol. 1, núm. 3, 190-206.
- Glover, F., 1990. Búsqueda tabú, Parte II, ORSA Journal on Computing, vol. 2, núm. 1, 4-32.
- Glover, F. y M. Laguna (1991) "Análisis de objetivos para mejorar un método de búsqueda tabú para la programación automática", The Arabian Journal for Science and Engineering, vol. 16, núm. 2B, págs. 239-253.
- Glover, F. y M. Laguna (1993) "Tabu Search", Técnicas heurísticas modernas para problemas combinatorios, C. Reeves (ed.), Blackwell Scientific Publishing, Oxford, págs. 70-150.
- Glover, F. y Laguna, M., (1997). Búsqueda tabú, Kluwer Academic Publishers, Boston.
- Laguna, M. (2005) "Implementación y prueba del ciclo Tabú y métodos de probabilidad condicional", <http://leeds-faculty.colorado.edu/laguna/articles/tabucycle.html>
- Laguna, M., JW Barnes y F. Glover (1993) "Programación inteligente con búsqueda tabú: una aplicación a trabajos con penalizaciones por retraso lineal y tiempos y costos de configuración dependientes de la secuencia", Journal of Applied Intelligence, vol. 3, págs. 159-172.
- Laguna, M. y F. Glover (1993) "Integración del análisis de objetivos y la búsqueda tabú para sistemas de programación mejorados", Sistemas expertos con aplicaciones, vol. 6, págs. 287-297.

Laguna, M., Martí, R., 2003. Búsqueda dispersa: metodología e implementaciones en C, Kluwer Academic Publishers, Boston.

Laguna, M., Martí, R. y Campos, V. (1999), Intensificación y diversificación con soluciones de búsqueda Elite Tabu para el problema de ordenamiento lineal, Computación e investigación de operaciones, 26, 1217-1230

LOLIB (1997) <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/LOLIB/LOLIB.html>.

Martí, R., Laguna, M. y Glover, F. (2004), Principios de búsqueda dispersa, European Journal of Operational Research, de próxima publicación.

Reinelt, G. (1985) El problema del ordenamiento lineal: algoritmos y aplicaciones, investigación y exposición en matemáticas, vol. 8, HH Hofmann y R. Wille (Eds.), Heldermann Verlag Berlin.