

Búsqueda Local

Búsqueda local

- Búsqueda local es la base de muchos de los métodos usados en problemas de optimización.
- Se puede ver como un proceso iterativo que empieza en una solución y la mejora realizando modificaciones.
- Ventajas:
 - Usan muy poca memoria.
 - A menudo encuentran soluciones razonables en espacios de estados grandes o infinitos.

Búsqueda local

- Considere el siguiente problema de optimización

$$\textit{Min } f(s)$$

s.a.

$$s \in S$$

Una solución $i^* \in S$ es llamada un óptimo global si para toda $s \in S$, $f(i^*) \leq f(s)$.

Búsqueda local

- Definición: Sea (S, f) una instancia de un problema de optimización combinatoria. Un vecindario es una función $N: S \rightarrow 2^S$ que, para cada solución $s \in S$, define un conjunto de soluciones $N(s)$ que están, en cierto sentido, cerca de s .
- Una solución $s' \in N(s)$ es llamada vecina de s .

Búsqueda local

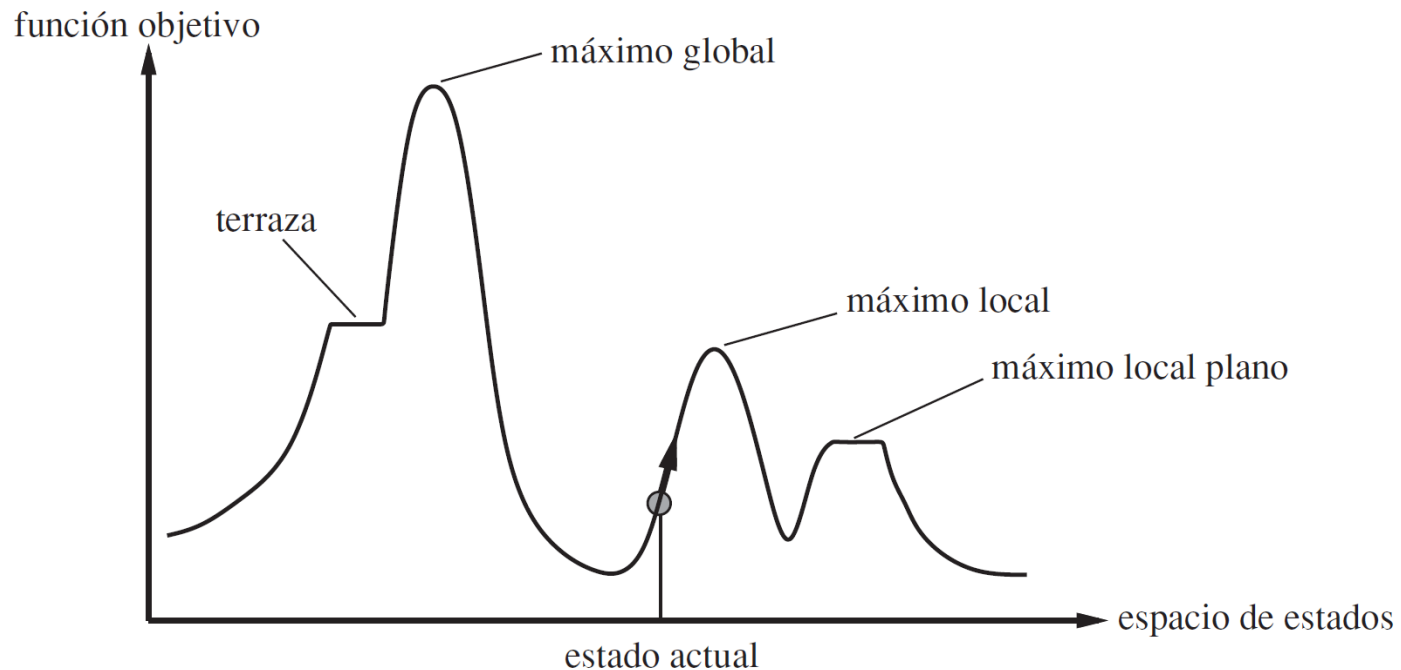
- Definición: Sea (S, f) una instancia de un problema de optimización combinatoria y sea N una función de vecindario. Una solución $i \in S$ es localmente mínima con respecto a N si *para toda* $j \in N(i)$, $f(i) \leq f(j)$.
- La optimalidad local depende por lo tanto de la función de vecindario utilizada.

Búsqueda local

- Definición: Sea (S, f) una instancia de un problema de optimización combinatoria y sea N una función de vecindario. Se dice que N es exacto si los óptimos locales (de acuerdo a N) también son óptimos globales.
- En caso de conocer un vecindario exacto, un simple algoritmo glotón (sin importar la modalidad de mejora) garantiza llegar a una solución óptima.
- Implica, sin embargo, tiempos exponenciales para explorarlo

Búsqueda local

- Un paisaje del espacio de estados unidimensional, la elevación corresponde a la función objetivo.



Búsqueda local

- Además de encontrar los objetivos, los algoritmos de búsqueda local son útiles para resolver problemas de optimización puros, en los cuales el objetivo es encontrar el mejor estado según una función objetivo.
- Dependiendo del problema analizado, el objetivo puede ser encontrar el mínimo o el máximo global.
- Los algoritmos de búsqueda exploran el paisaje.
- Un algoritmo completo encuentra un objetivo si existe.
- Un algoritmo óptimo encuentra siempre el mínimo/máximo global.

Búsqueda local

- Los algoritmos que veremos en esta parte del curso son:
 - Búsqueda de ascenso por la colina (hill-climbing).
 - Búsqueda por haz local.
 - Recocido simulado.

Ascenso en la colina

Ascenso en la colina

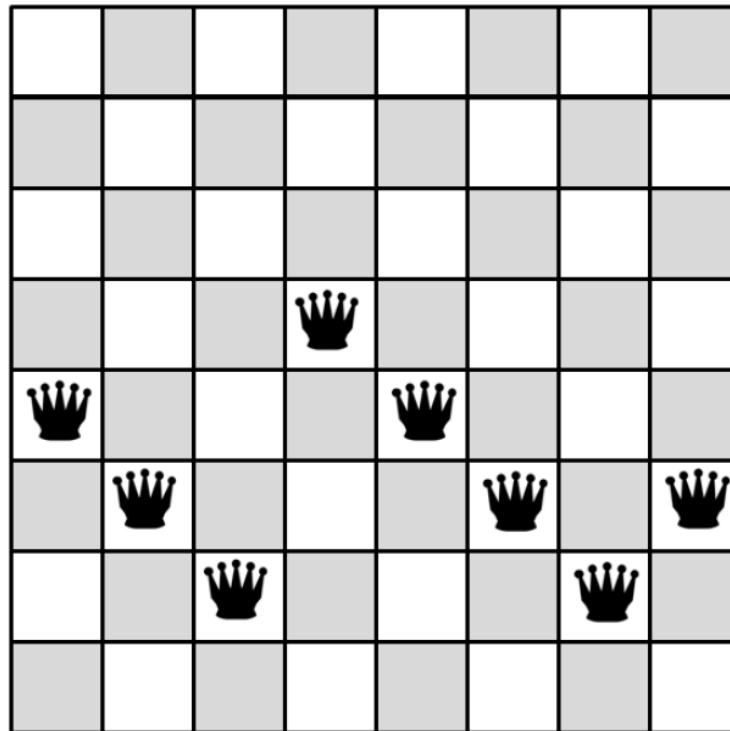
- Consiste en moverse en dirección del valor creciente, hasta alcanzar un pico en donde ningún vecino tenga valor más alto.
- Para cada estado es necesario definir a los estados vecinos y evaluarlos en base a una función que representa su costo a minimizar/maximizar.

Problema de las 8 reinas

- Formulación de estados completa. Cada estado tiene ocho reinas sobre el tablero, una por columna.
- La función sucesor devuelve todos los estados posibles generados por mover una reina a otro cuadrado en la misma columna (8 reinas x 7 espacios disponibles para cada una = 56 sucesores).
- La función heurística es el número de pares de reinas que están sobre la misma línea horizontal, vertical o diagonal (sin importar si en el estado actual no se puedan atacar por la presencia de una tercer reina entre ellas).









Ascenso en la colina

- Para el siguiente estado el costo es 17



Ascenso en la colina

- Se calcula el costo para cada estado sucesor

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14		13	16	13	16
	14	17	15		14	16	16
17		16	18	15		15	
18	14		15	15	14		16
14	14	13	17	12	14	12	18

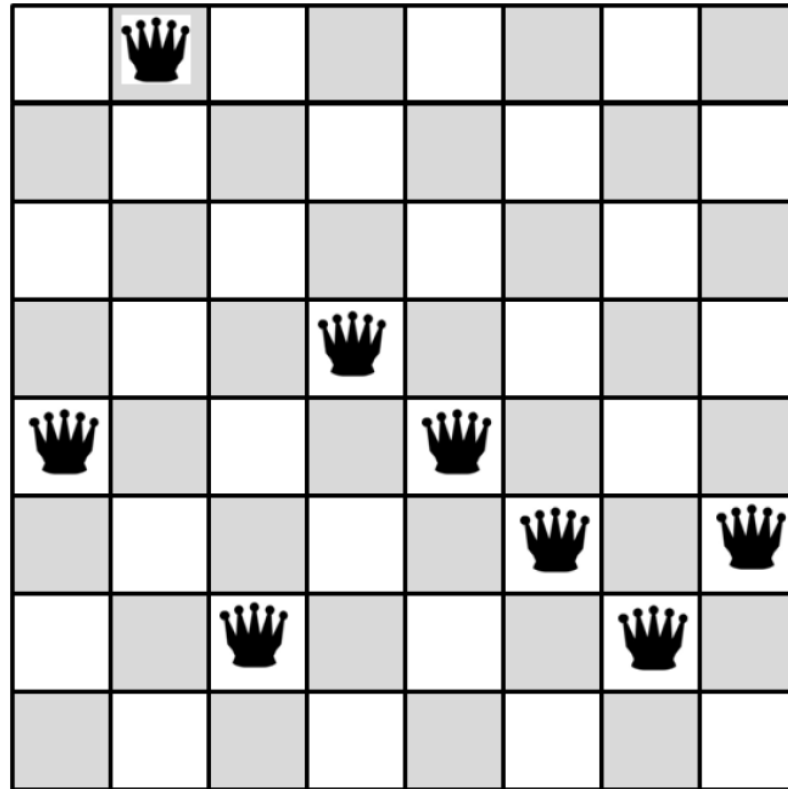
Ascenso en la colina

- Se elige el movimiento que lleve a la mejor función de costo

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	👑	13	16	13	16
👑	14	17	15	👑	14	16	16
17	👑	16	18	15	👑	15	👑
18	14	👑	15	15	14	👑	16
14	14	13	17	12	14	12	18

Ascenso en la colina

- En el estado actual el costo es $h = 12$



Ascenso en la colina

- Al ascenso de la colina se le llama también **búsqueda local voraz**.
- Este método a menudo funciona bien y progresa rápidamente hacia las soluciones.
- Sin embargo, el método a veces se atasca debido a máximos locales o mesetas.
- Para el problema de las 8 reinas, comenzando desde un estado aleatorio, este algoritmo se estanca el 86% de las veces. Sin embargo, trabaja rápidamente: en promedio usa: 4 pasos cuando tiene éxito y 3 cuando fracasa.

Ascenso en la colina con movimientos laterales

- Si se llega a un estado desde el cual no existe un estado mejor al cual moverse, pero sí existen estados con igual función de costo, entonces se realiza un movimiento a uno de esos estados.
- Se debe poner un límite al número de movimientos laterales para no caer en un bucle infinito.
- En el caso del problema de las 8 reinas, esto aumenta el éxito a 94%, con un promedio de 21 pasos cuando se tiene éxito y 64 cuando se fracasa.

Ascenso en la colina estocástica

- Escoge aleatoriamente de entre los movimiento ascendentes.
- La probabilidad de selección puede variar con la pendiente del movimiento ascendente.
- Converge más despacio, pero en algunos paisajes de estados encuentra mejores soluciones.

Ascenso en la colina primera opción

- Es un ascenso en la colina estocástica generando sucesores al azar hasta que se genera uno que es mejor que el estado actual.
- Es una buena estrategia cuando un estado tienen muchos sucesores.

Ascenso en la colina con reinicio aleatorio

- Es una serie de búsquedas desde estados iniciales generados aleatoriamente, parándose cuando se encuentra un objetivo.
- Es completo, por la razón trivial de que generará eventualmente un estado objetivo como el estado inicial.

Búsqueda por haz local

Búsqueda por haz local

- Guardar solamente un nodo en la memoria podría parecer una reacción extrema para el problema de limitaciones de memoria. Sin embargo, puede limitar el desempeño del algoritmo.
- El algoritmo de **búsqueda por haz local** guarda comienza con k estados generados aleatoriamente.
- En cada paso, se generan todos los sucesores de los k estados.
- Si alguno es un objetivo, paramos el algoritmo.
- En caso contrario, se seleccionan los k mejores sucesores de la lista completa y repetimos.

Búsqueda por haz local

- Una búsqueda por haz local con k estados podría parecerse a ejecutar k reinicios en secuencia.
- Sin embargo, los dos algoritmos son bastante diferentes.
- En una búsqueda de reinicio aleatorio, cada proceso de búsqueda se ejecuta independientemente de los demás.
- En una búsqueda por haz local, la información útil es pasada entre los k hilos paralelos de búsqueda.

Búsqueda por haz local

- Por ejemplo, si un estado genera varios sucesores buenos y los otros estados generan sucesores malos
- Entonces el efecto es que el primer estado dice a los demás, «¡Venid aquí, la hierba es más verde!»
- El algoritmo rápidamente abandona las búsquedas infructuosas y mueve sus recursos a donde se hace la mayor parte del progreso.

Búsqueda por haz local

- En su forma más simple, la búsqueda de haz local puede sufrir una carencia de diversidad entre los k estados.
- Una variante llamada **búsqueda de haz estocástica**, ayuda a aliviar este problema.
- En vez de elegir los k mejores del conjunto de sucesores candidatos, se escogen a k sucesores aleatoriamente, con la probabilidad de elegir a un sucesor como una función dependiente de su valor.

Recocido simulado

Antecedentes

- Búsqueda local que permite movimientos ascendentes para evitar óptimos locales.
- Principio fácil de entender.
- Su eficacia depende de identificar y definir cada una de las componentes.
- Cada componente tienen gran influencia en la calidad de las soluciones.
- Recocido simulado fue introducido por Kirkpatrick, Gellat y Vecchi en 1983.

Proceso físico

- Recocido denota un proceso de calentamiento de un sólido seguido de un proceso de enfriamiento en donde la temperatura se baja poco a poco
- Se inicia con un valor máximo de temperatura T , en la fase de enfriamiento del proceso de recocido, para cada valor de T debe permitirse que se alcance un equilibrio térmico.

Proceso físico

- El proceso de recocido consta de dos pasos fundamentales
 - Incrementar la temperatura del baño térmico a un valor máximo.
 - Disminuir cuidadosamente la temperatura del baño térmico hasta que las partículas se reacomoden por sí mismas en un estado de mínima energía, llamado el estado fundamental del sólido.

Proceso físico

- Al elevar la temperatura del sólido todas las partículas se reacomodan aleatoriamente
- En el estado fundamental del sólido, las partículas se acomodan en una retícula altamente estructurada y la energía del sistema es mínima.
- El estado fundamental del sólido se obtiene sólo si la temperatura máxima es suficientemente elevada y el proceso de enfriamiento es suficientemente bajo.
- El proceso de recocido puede modelarse exitosamente usando métodos de simulación

Simulación

- Metropolis et al. (1953) modelan el proceso de recocido simulando los cambios energéticos en un sistema de partículas conforme disminuye la temperatura hasta que converge a un estado estable.
- En el algoritmo de Metropolis se genera una perturbación aleatoria en el sistema y se calculan los cambios de energía resultantes: si hay una caída energética, el cambio se acepta; si hay incremento se acepta con cierta probabilidad.
- El proceso se repite durante un número predefinido de iteraciones en series decrecientes de temperaturas, hasta que el sistema esté frío.

Recocido Simulado

- La simulación del proceso de recocido puede usarse para describir un proceso de generación de sucesiones de soluciones en un POC en donde conforme avance el proceso, se vayan obteniendo mejores soluciones.
- Cada solución puede verse como un estado del sólido y los valores de la función objetivo para cada solución del POC como los niveles de energía del sólido.
- También se introduce un parámetro que hace un papel equivalente a la temperatura llamado parámetro de control.

Analogía con optimización

Simulación termodinámica	Optimización combinatoria
Sistema de estados	Soluciones factibles
Energía	Costos
Cambio de estado	Soluciones vecinas
Temperatura	Control de parámetros
Temperatura de enfriamiento	Solución heurística

Recocido simulado

- Como en los algoritmos de búsqueda local se supone la existencia de una estructura de vecindades y un mecanismo de generación.
- Definición. Sea (S,C) un caso de un POC e i y j dos soluciones con costo $C(i)$ y $C(j)$ respectivamente. Entonces el criterio, de Metrópolis, para j es:

$$p_a = \begin{cases} e^{-\frac{k \Delta E}{T}} & \Delta E > 0 \\ 1 & \Delta E \leq 0 \end{cases}$$

- $T > 0$ es el parámetro de control.
- $\Delta E = C(i) - C(j)$, con j solución actual, i solución vecina.

Recocido simulado

- Una transición es una acción combinada que transforma la solución actual en una subsecuente, acción que consta de
 - Aplicación del mecanismo de generación
 - Aplicación del criterio de aceptación.
- El mecanismo de generación corresponde al mecanismo de perturbación en el algoritmo de Metrópolis.
- El criterio de aceptación corresponde al criterio de Metrópolis (si hay caída de energía, el cambio se acepta automáticamente, si se produce un incremento energético, el cambio será aceptado con una probabilidad)

Pseudocódigo

- Sean $C(s)$ el costo de la solución actual y $V(s)$ una vecindad
- Seleccionar una solución inicial s_0
- Seleccionar una temperatura inicial $t_0 > 0$, función de reducción de temperatura
- Seleccionar un número de iteraciones n_{rep} , un criterio de parada
- REPETIR hasta criterio de parada
 - REPETIR hasta n_{rep} iteraciones
 - Seleccionar aleatoriamente una solución $s \in V(s_0)$;
 - $\delta = C(s) - C(s_0)$
 - Si $\delta < 0$ entonces $s_0 = s$
 - Si no
 - generar aleatoriamente $x \in U(0,1)$
 - si $x < \exp(-\delta/c)$ entonces $s_0 = s$
 - Actualizar temperatura

Consideraciones importantes

Es un algoritmo muy general y para implementarlo deben tomarse muchas decisiones:

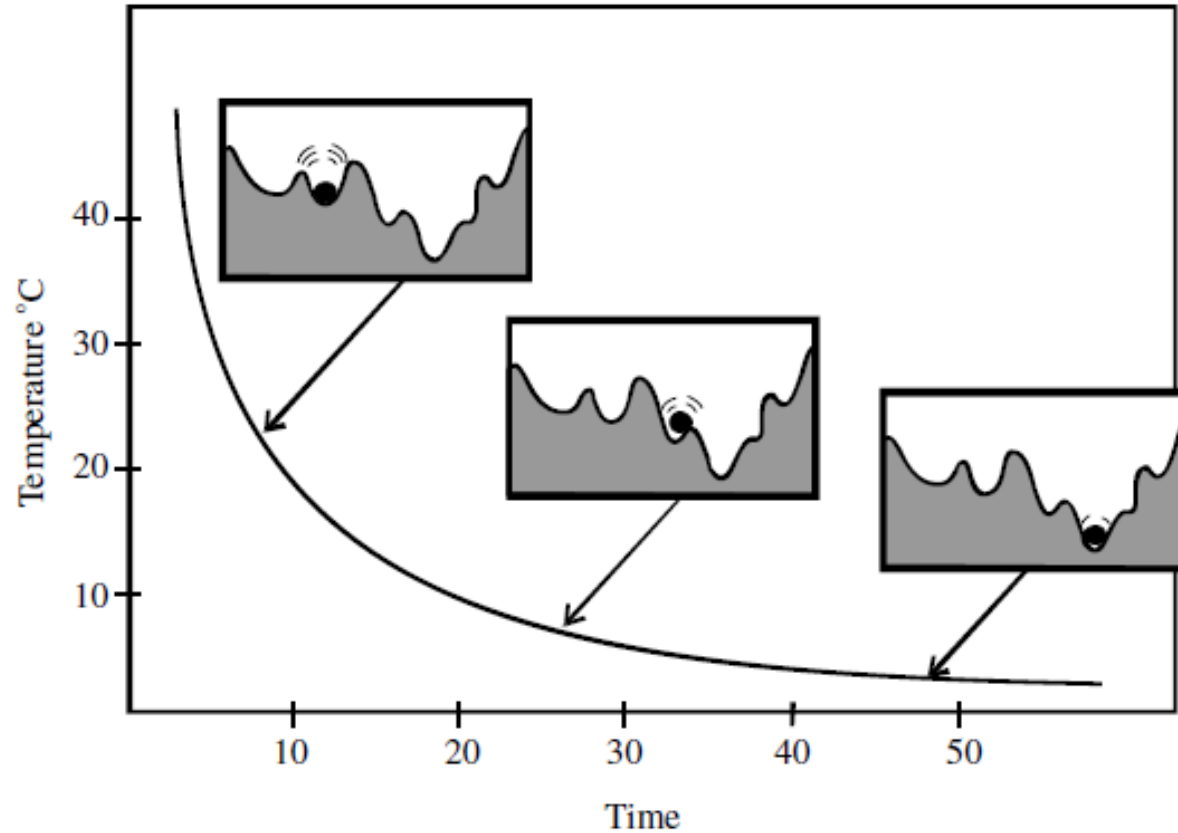
- Una descripción concisa de la representación del problema
 - Espacio de soluciones
 - Una expresión de la función de costo
 - La función de costo debe representar la efectividad de las soluciones, las cuales deben permitir fácil manipulación.
- La generación de ensayos para transformar la solución actual en una subsecuente consta de los siguientes pasos.
 - Se debe generar una nueva solución aplicando un mecanismo de generación.
 - Calcular la diferencia de costo de las soluciones
 - Decidir si se acepta o no la nueva solución.

Consideraciones importantes

El proceso de recocido requiere la especificación de los parámetros que determinan el programa de enfriamiento.

- Valor inicial de la temperatura
- Una función que especifique el enfriamiento
- La longitud de cada bloque donde permanece constante la temperatura
- Criterio de parada.

Temperatura inicial



Ventajas de su uso

- Fácil de entender e implementar
- No se requiere de un modelo matemático sólo perturbar y evaluar
- Ha demostrado su habilidad en diferentes problemas.
- La calidad de las soluciones no depende de la solución inicial.

Desventajas

- Una temperatura inicial alta permite fuertes perturbaciones, y llegar a valles profundos o a cimas muy altas.
 - ¿Qué tan fuertes debe ser las perturbaciones?
 - ¿Durante cuanto tiempo aplicar el mismo nivel de perturbación?

Desventajas

- Descenso de la temperatura inadecuado.
 - Muy despacio hace lento al algoritmo
 - Muy rápido disminuye sus probabilidades de eficacia

Desventajas

- Alta dependencia del generador de números aleatorios
 - Requiere muchos valores
 - La calidad del generador afectará la calidad de los resultados

Recocido simulado adaptable

- En este caso los parámetros que controlan la temperatura se ajustan de forma automática conforme el algoritmo avanza.
- El código se encuentra disponible en internet
- Recalentamiento, en algunos casos resulta conveniente aumentar la temperatura.

Actualmente

ScienceDirect Journals Books Shopping cart Sign in Help

simulated annealing Author name Journal or book title Volume Issue Page Advanced search

Search results: 60,164 results found. Save search alert RSS

Refine filters
Year
☐ 2015 (1,718)
☐ 2014 (4,943)
☐ 2013 (4,398)
☐ 2012 (3,858)
☐ 2011 (3,539)
[View more >>](#)
Publication title
☐ Journal of Molecular Biology (2,478)
☐ European Journal of Operational Research (1,409)
☐ Surface Science (1,294)
☐ Nuclear Instruments and Methods in Physics Rese... (1,214)
☐ Structure (1,043)

Purchase Export

Relevance All access types

☐ **Multi-criteria scheduling of Bag-of-Tasks applications on heterogeneous interlinked clouds with simulated annealing** Original
Research Article
Journal of Systems and Software, Volume 101, March 2015, Pages 1-14
Ioannis A. Moschakis, Helen D. Karatza
[Abstract](#) [Research highlights](#) [Purchase PDF - \\$35.95](#)

☐ **Bayesian system identification of a nonlinear dynamical system using a novel variant of Simulated Annealing** Original Research Article
Mechanical Systems and Signal Processing, Volumes 52-53, February 2015, Pages 133-146
P.L. Green
[Abstract](#) [Research highlights](#) [PDF \(2143 K\)](#) Open Access

☐ **A dual representation simulated annealing algorithm for the bandwidth minimization problem on graphs** Original Research Article
Information Sciences, Volume 303, 10 May 2015, Pages 33-49
Jose Torres-Jimenez, Idelfonso Izquierdo-Marquez, Alberto Garcia-Robledo, Aldo Gonzalez-Gomez, Javier Bernal, Raghu N. Kacker
[Abstract](#) [Research highlights](#) [Purchase PDF - \\$37.95](#)