

模擬退火法基礎

Foundations of Simulated Annealing

授課教師：陳士杰

國立聯合大學 資訊管理學系





■ 授課目標

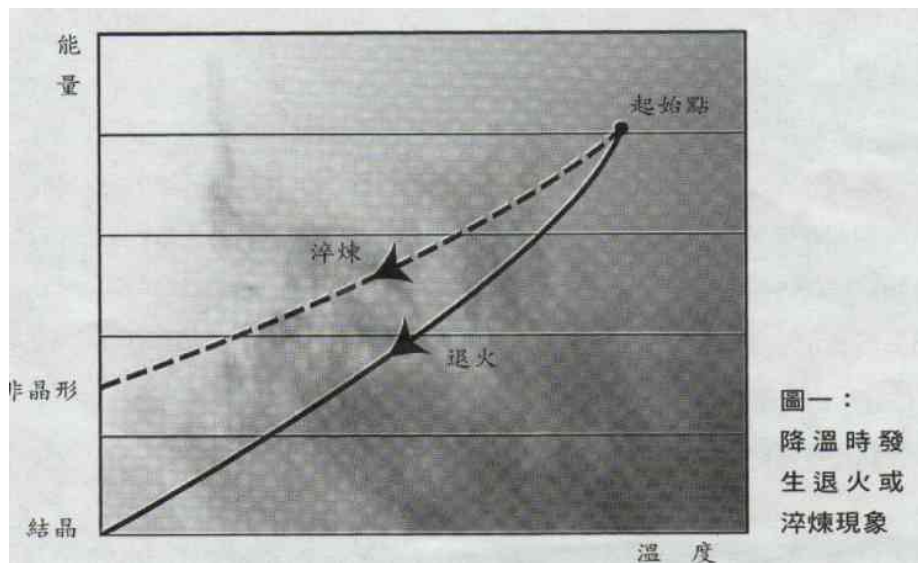
- 簡介
- 發展沿革
- 基本觀念
- 重要名詞
- 操作步驟
- 實際範例
- 基本特性
- 前提與限制



■ 簡介

● 模擬退火法是模擬**冷卻晶體**的過程。

- 在熱力學上，**退火 (Annealing)** 現象指物體逐漸降溫的物理現象，溫度愈低，物體的能量狀態會低；夠低後，液體開始冷凝與結晶，在結晶狀態時，系統的能量狀態最低。
- 如下圖所示，大自然在緩慢降溫 (亦即，退火) 時，可「找到」最低能量狀態：**結晶**。但是，如果過程急就章，快速降溫 (亦稱「淬煉」，Quenching) 時，會導致不是最低態的非晶形。



- 大自然知道慢工出細活：**緩緩降溫**，使得物體分子在每一溫度時，能有充足的時間找到安頓位置，則逐漸地，到最後可得到最低能量狀態時，系統最安穩。

人類在找尋最適解 (optimal solution)，例如系統的最低能量狀態時，可以學學大自然的「智慧」嗎？



發展沿革

- 美國物理學家默察波利斯 (N. Metropolis) 和同仁在 1953 年使用蒙地卡羅模擬 (Monte Carlo simulation) 計算一組分子由一特定高溫逐漸達到冷卻的行為。
- 美國 IBM 公司物理學家科克派特瑞克 (S. Kirkpatrick) 和同仁於 1983 年在《科學》(Science) 上發表了一篇頗具影響力的文章：〈以模擬退火法求最適解〉(Optimization by Simulated Annealing)。
 - 他們發覺其物理系統的能量和一些組合最佳化 (combinatorial optimization) 問題的成本函數相當類似：尋求最低成本即似尋求最低能量。於是，他們發展出以默察波利方法為本的一套演算法 (algorithm)，可用來解決組合問題等的尋求最適解。
 - 科克派特瑞克等人受到默察波利斯等人的啟發而發明了「**模擬退火**」這個名詞，因為它和物體退火過程相類似。系統降溫時，能量也逐漸下降，而同樣意義地，問題的解也「下降」到最小值。此兩者的類似可用下表對照。



表一：自然降溫程序與人工「模擬退火」之相似性比較

自然程序	人工程序
狀態 (組態)	可行解 (feasible solution)
基態 (ground state)	最適解
快速淬煉 (rapid quenching)	局部尋找 (local search)
小心退火 (careful annealing)	模擬退火



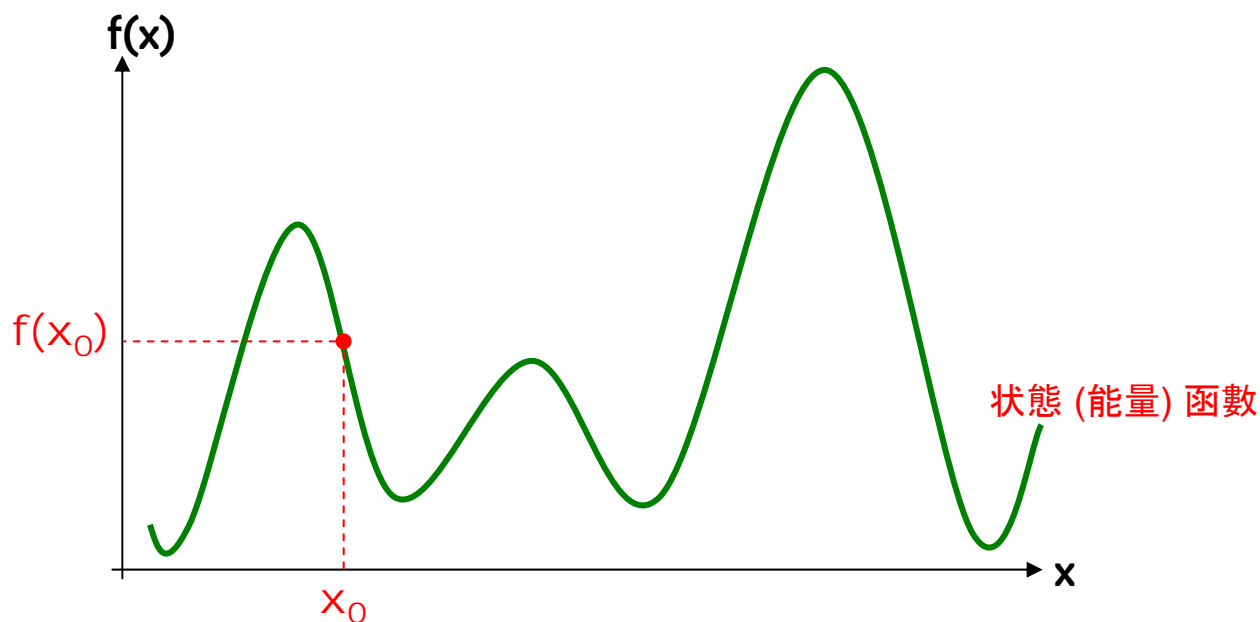
- 幾乎同時，歐洲物理學家卡尼 (V.Carny) 也發表了幾乎相同的發現，但兩者是各自獨立發現的；只是卡尼「運氣不佳」，當時沒什麼人注意到他的大作；或許可以說，《科學》雜誌行銷全球，「曝光度」很高，素負盛名，而卡尼卻在發行數量小的專門學術期刊 (J.Opt.Theory Appl.) 發表的。



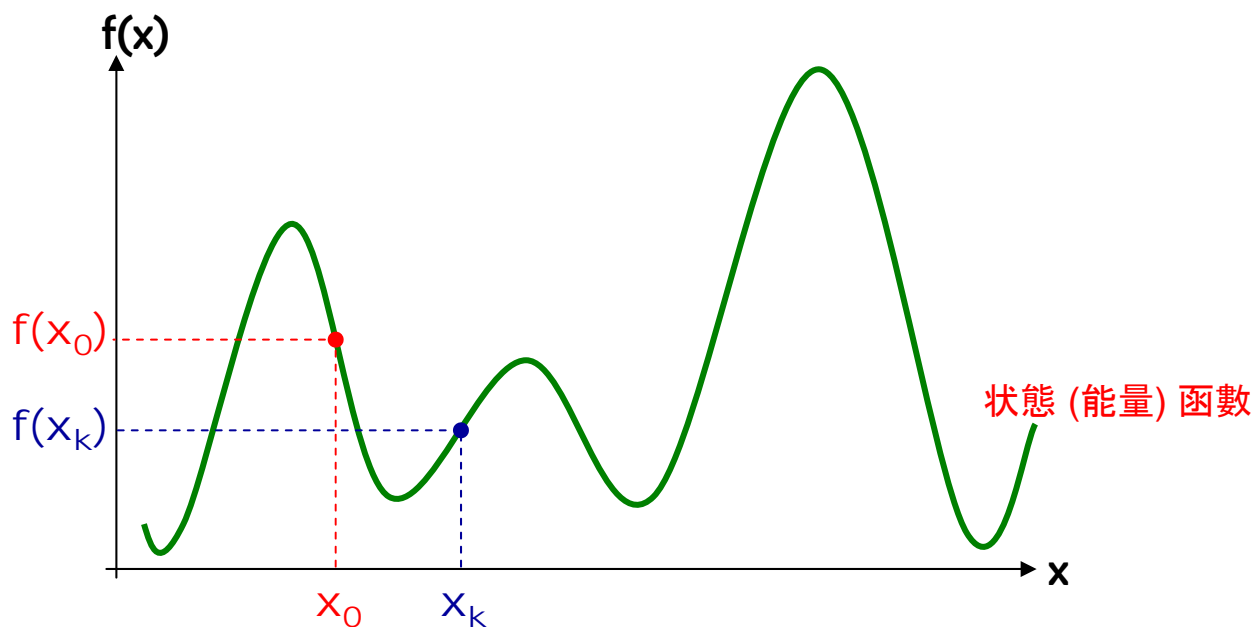
基本觀念

- 假設我們的目標是為找到一個最低能量水準的狀態。不同狀態 x 下的能量水準為 $f(x)$ 。

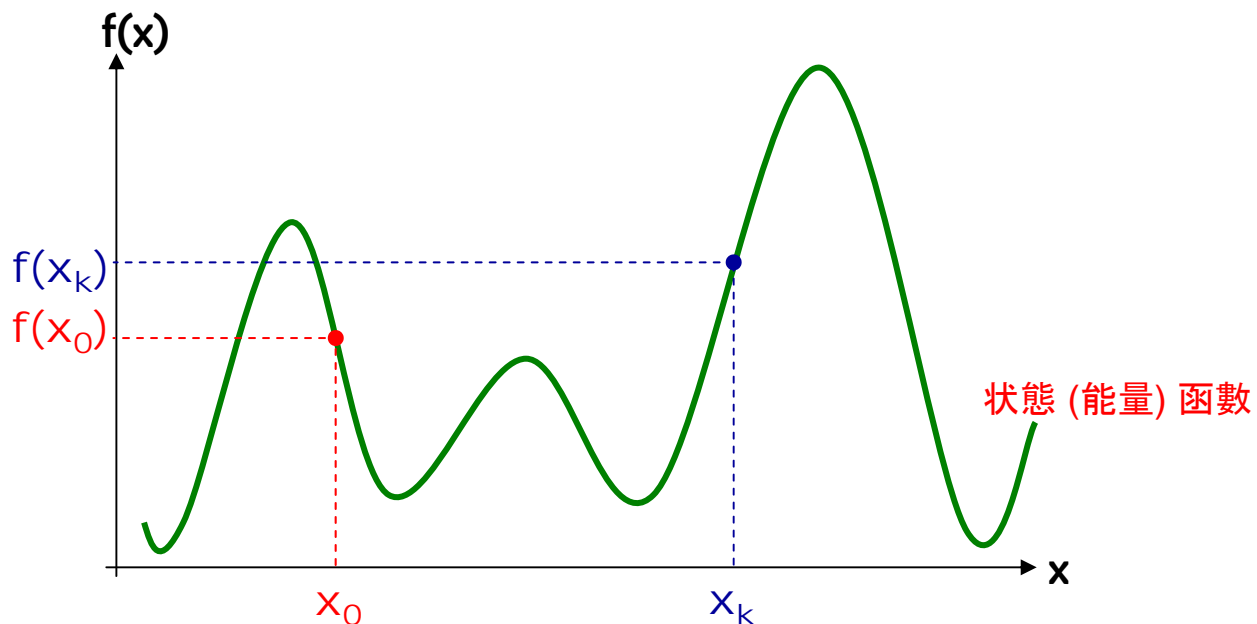
■ 有一個起始的高溫分子狀態 x_0 ，其狀態值為 $f(x_0)$



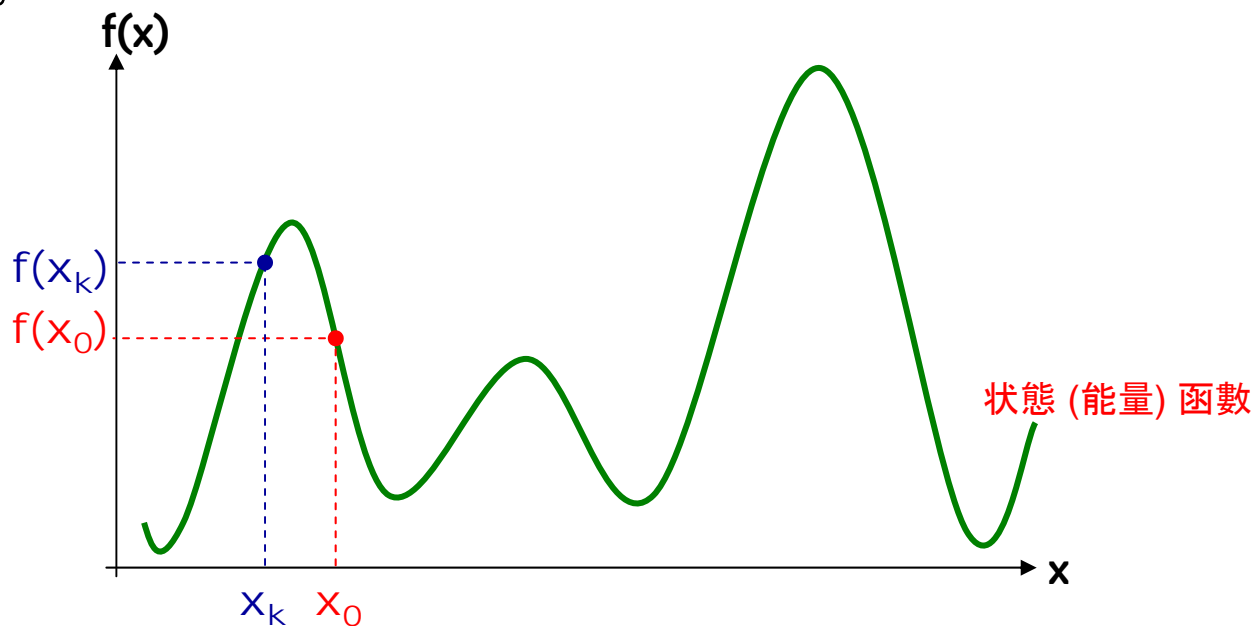
- 對此起始的狀態 x_0 做改變 (擾動、干擾), 以得到一個新的狀態 x_k 。
- 若 $f(x_k) < f(x_0)$, 則表示此新狀態比起始狀態更接近最佳值 (即: 最小值)。故接受此一狀態, 並以此狀態為基礎, 重覆進行干擾, 以尋找下一個更佳的状态。



- 若 $f(x_k) > f(x_0)$ ，則表示此新狀態比起始狀態更遠離最佳值 (即：最小值)。故傳統啟發式演算法將會放棄此一狀態，並重新尋找下一個更佳的状态。

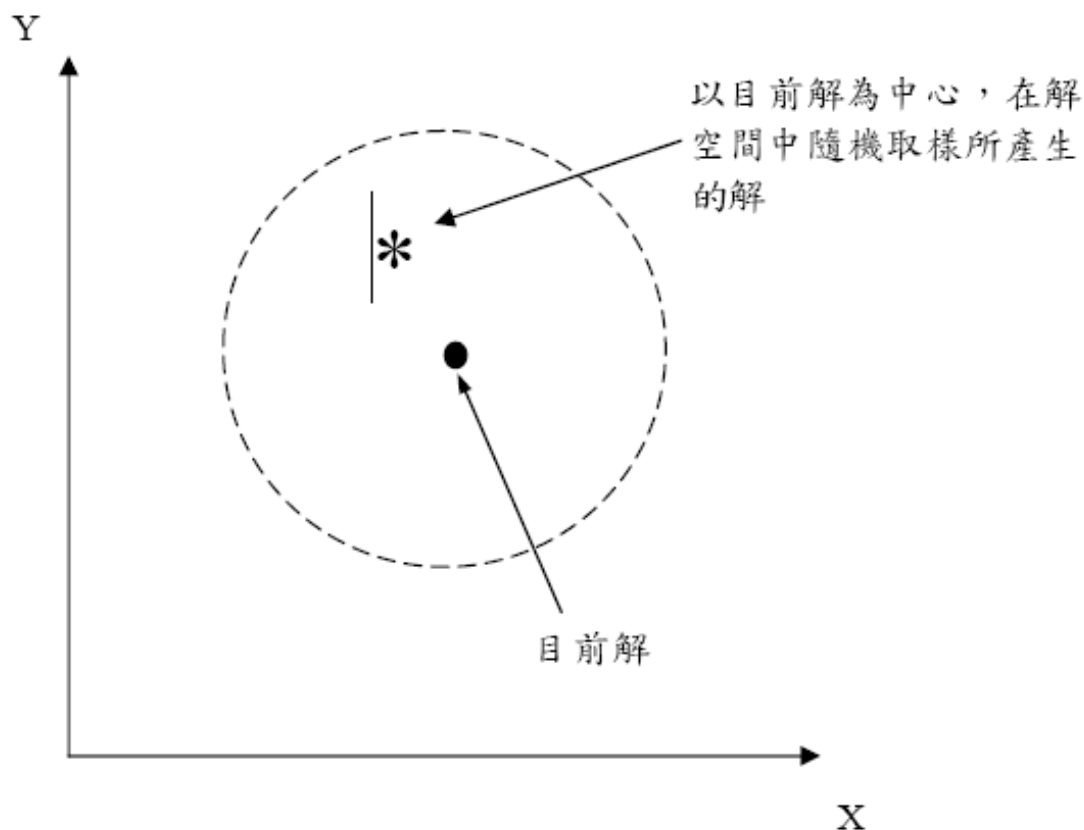


- 然而， $f(x_k) > f(x_0)$ 不一定表示在此一狀態之後無法找到真正的最佳解。
- SA會在一「**設定的機率**」下，接受看似不佳的狀態 x_k ，並以此狀態為基礎，進行下一個尋優動作。此一尋優路線有可能找到最佳解。。



擾動概念

- 擾動的作法就是以目前解為中心，對部分或整個解空間隨機取樣一個解。





■ 重要名詞

- 狀態 (State)
- 能量 (Energy)
- 基礎狀態 (Ground State)
- 波茲曼函數 (Boltzman's Function)
- 控制溫度 (Control Temperature)
- 冷卻時程 (Cooling Schedule)
- 鄰域搜尋法 (Neighborhood Search)
- 停止條件 (Stopping Criterion)





狀態 (STATE)

- 狀態原是指物質在高溫冷卻的過程中，於每一個時點之物質特性及其溫度。
- 應用於求解問題時，則是指尋優過程中所找到的**每一個可行解** (Feasible Solution).
 - 如： x_0 (起始解)、 x_k (某一可行解)





能量 (ENERGY)

- 能量是指物質在高溫冷卻的過程中，於每一個時點之物質當時的物理狀況。
- 應用於求解問題時，則是指尋優過程中所找到的每一個可行解之目標狀態值，即： $f(x)$ 。
 - 如：起始解 x_0 的狀態值為 $f(x_0)$ 、某一可行解 x_k 的狀態值為 $f(x_k)$





基礎狀態 (GROUND STATE)

- 指物質最後冷卻的狀態。
- 應用於求解問題時，則是指尋優過程最後所找到的最佳解。



波茲曼函數 (BOLTZMAN'S FUNCTION)

- SA的最大特色，即是在尋優過程中若發生某一狀態較前一狀態惡化時，SA會以一設定的機率決定是否接受此一狀態，而不是直接放棄。
- 因此：
 - 若機率值太低，可能無法達到跳離局部解之功效
 - 若機率值太高，可能導致SA左右跳動而無法收斂
- 設定機率的計算方式，是依照物質冷卻過程所依循的波茲曼函數加以計算：

$$PR(A) = \min\left\{1, e^{\left(\frac{-\Delta f}{T_k}\right)}\right\}$$





● 其中：

- $PR(A)$ 是接受狀態 x_k 之機率
- Δf 是狀態 x_k 的狀態值 $f(x_k)$ 與前一狀態 x_{k-1} 的狀態值 $f(x_{k-1})$ 之差距，即：

$$\Delta f = f(x_k) - f(x_{k-1})$$

- T_k 是指第 k 個狀態的控制溫度 (Control Temperature)

● 當機率值 $PR(A)$ 計算出來之後，再利用亂數產生器，產生一個介於 0 與 1 之間的亂數 R ，以判斷是否接受該狀態：

- 若 $R > PR(A)$ 則放棄狀態 x_k
- 若 $R \leq PR(A)$ 則接受狀態 x_k

● 因此， $PR(A)$ 之值愈高，狀態 x_k 被接受的機率也愈高





控制溫度 (CONTROL TEMPERATURE)

- 在波茲曼函數中，設有一控制溫度 T_k ，用以決定接受機率的高低
- T_k 值的設計應會隨著尋優過程而降低
- 適當的控制溫度，可使SA能有效地跳離局部解，又能在有限時間內收斂，是SA一項非常重要的控制參數



冷卻時程 (COOLING SCHEDULE)

- 由高溫往低溫行進的一種溫度降低方式 ($T_k \rightarrow T_{k+1}$)
- 基本上，控制溫度 T_k 並非是一個常數，而是一個會隨著尋優次數增加而降低的變數。這是因為：
 - 一般SA的設計會在尋優初期讓接受機率 $PR(A)$ 較高，以便擴大搜尋範圍，增加找到最佳解的機會。但在尋優後期，則會使接受機率 $PR(A)$ 較低，以利於收斂。
 - 因此，接受機率 $PR(A)$ 應該會隨著尋優過程而降低。
- 最簡單且常用的計算方式，是將現在的溫度 T_k 乘上一個介於0與1之間的小數 (α)，或是減去一個固定數值 (σ)，以求得下一時間點的溫度。

$$T_{k+1} = \alpha T_k ; T_{k+1} = T_k - \sigma$$

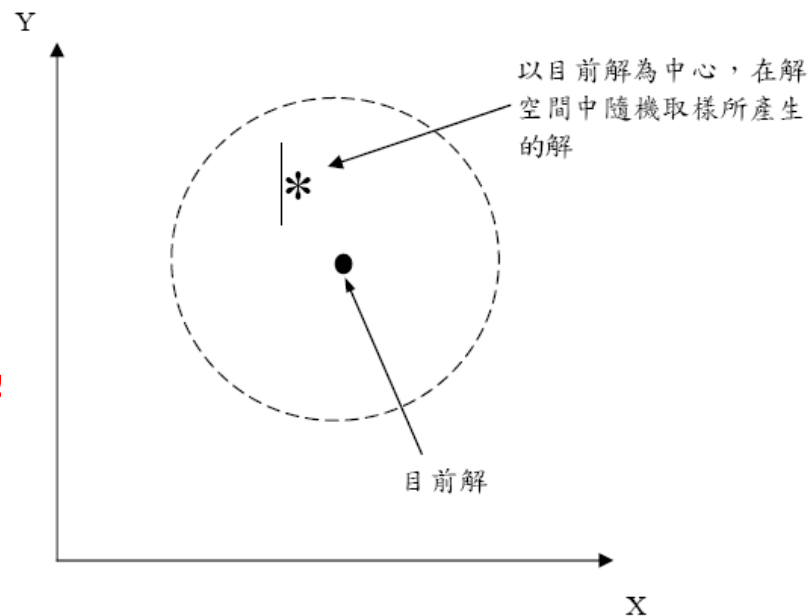
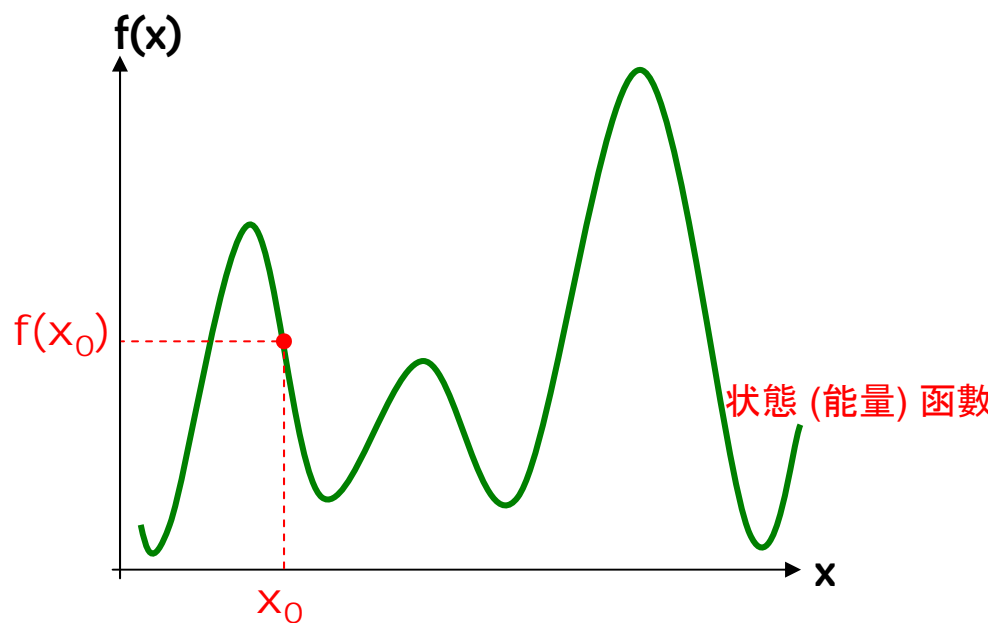


- 也有更複雜的溫度控制方式，即 α 與 σ 的值還會隨著尋優過程而改變。但是何種冷卻時程較佳，並無定論，需視問題特性而定。
- 通常 α 可設定為0.95，至於 σ 則可設為 $(1 - \alpha)T_0$



鄰域搜尋法 (NEIGHBORHOOD SEARCH)

- 應用於求解問題時，是指由起始解之設定方式，以及由某一可行解尋找下一個鄰近區域 (Neighborhood) 可行解的尋優方式。





停止條件 (STOPPING CRITERION)

- 停止條件是用以判斷SA重覆尋優動作是否結束的依據。
- 一般的設定有：
 - 總尋優次數達到一定數量
 - 狀態值持續未獲改善的尋優次數達到一定數量





■ 操作步驟 (以最小化問題為例)

- 設定起始狀態
- 執行鄰域搜尋
- 執行冷卻時程
- 測試停止條件





● 設定起始狀態

- 產生一個起始可行解 x_0 ，其狀態值為 $f(x_0)$
- 設定起始溫度 T_0
- 令：目前最佳狀態 $x_b = x_0$ ，目前最佳狀態值 $f(x_b) = f(x_0)$ ，現有溫度 $T_c = T_0$ ，尋優次數 $k = 0$ ，最大尋優次數 K

● 執行鄰域搜尋

● 執行冷卻時程

● 測試停止條件





- 設定起始狀態

- 執行鄰域搜尋

- 根據鄰域搜尋法與問題特性，由現有狀態 x_k 求解下一狀態 x_{k+1} 。
- 若 $f(x_{k+1}) \leq f(x_k)$ ，則 $x_b = x_{k+1}$ ， $f(x_b) = f(x_{k+1})$
- 若 $f(x_{k+1}) > f(x_k)$ ，則利用波茲曼函數與亂數產生器產生出 $PR(A)$ 與 R
 - 若 $R \leq PR(A)$ ，則令 $x_b = x_{k+1}$ ， $f(x_b) = f(x_{k+1})$
 - 若 $R > PR(A)$ ，則令 $x_b = x_k$ ， $f(x_b) = f(x_k)$
- $k=k+1$

- 執行冷卻時程

- 測試停止條件





- 設定起始狀態
- 執行鄰域搜尋
- 執行冷卻時程
 - 令 $T_{k+1} = \alpha T_k$ 或是 $T_{k+1} = T_k - \sigma$
- 測試停止條件



- 設定起始狀態
- 執行鄰域搜尋
- 執行冷卻時程
- 測試停止條件
 - 判斷尋優次數 k 是否大於最大尋優次數 K ，如果是則停止，最佳狀態及其值為 x_b 及 $f(x_b)$ 。如果為否，則回到第二步。



■ 實際範例

● 網路距離矩陣

節點	1	2	3	4	5	6	7
1	0	19	92	29	49	78	6
2	19	0	21	85	45	16	26
3	92	21	0	24	26	87	47
4	29	85	24	0	76	17	8
5	49	45	26	76	0	90	27
6	78	16	87	17	90	0	55
7	6	26	47	8	27	55	0



Step 1: 設定起始狀態

- 假設有一個起始解為 $x_0 = [1, 2, 3, 4, 5, 6, 7]$, 即: 由節點1依序經2, 3, 4, 5, 6, 7等節點再回至節點1。此路徑總距離為291
- 令 $x_b = x_0$ 、 $f(x_b) = 291$ 、起始尋優次數 $k = 0$ 、最大尋優次數 $K = 100$ 、現有溫度 $T_c = T_0 = 300$



Step 2: 執行鄰近搜尋

- 利用TSP的2-opt的方法進行狀態更新，即隨機任選兩節點互換其順序，以成為一個新的路線。
- 故起始路徑 [1, 2, 3, 4, 5, 6, 7] 之所有可能的鄰近解有 $C_2^7 = 21$ 組，如右表所示。
- 假設隨機選擇節點5與節點2互換，即成為[1, 5, 3, 4, 2, 6, 7]，該路徑的總距離為261。
- 因 $261 < 291$ ，故令 $x_b = [1, 5, 3, 4, 2, 6, 7]$ 、 $f(x_b) = 261$ 。
- 尋優次數 $k = k+1 = 1$

編號	路線	互換節點
1	2134567	1 2
2	3214567	1 3
3	4231567	1 4
4	5234167	1 5
5	6234517	1 6
6	7234561	1 7
7	1324567	2 3
8	1432567	2 4
9	1534267	2 5
10	1634527	2 6
11	1734562	2 7
12	1243567	3 4
13	1254367	3 5
14	1264537	3 6
15	1274563	3 7
16	1235467	4 5
17	1236547	4 6
18	1237564	4 7
19	1234657	5 6
20	1234765	5 7
21	1234576	6 7



Step 3: 執行冷卻時程

- $T_1 = 0.95 \times T_0 = 285$

Step 4: 測試停止條件

- 由於目前 $k = 1$ ，小於最大尋優次數 $K = 100$ ，不符合停止條件，故回至Step 2。



Step 2:執行鄰近搜尋

- 再隨機選擇兩個互換節點，如：節點3與節點6，則新路線成為[1, 5, 6, 4, 2, 3, 7]，該路徑的總距離為315。
- 因 $315 > 261$ ，故利用波茲曼函數計算接受機率：

$$PR(A) = \min\{1, e^{-[(315-261)/285]}\} = 0.8274$$

- 再利用亂數產生器取一介於0至1之間亂數值R，假設 $R = 0.4327$
- 由於 $R < PR(A)$ ，故令 $x_b = [1, 5, 6, 4, 2, 3, 7]$ 、 $f(x_b) = 315$
- 尋優次數 $k = k+1 = 2$



Step 3: 執行冷卻時程

- $T_2 = 0.95 \times T_1 = 271$

Step 4: 測試停止條件

- 由於目前 $k = 2$, 小於最大尋優次數 $K = 100$, 不符合停止條件, 故回至Step 2。

(重覆執行上述步驟, 直到停止條件成立)



基本特性

- 鄰域搜尋與冷卻時程是SA的兩大核心。
 - 鄰域搜尋是由現有狀態推演至下一狀態的方法，即傳統的啟發式解法。
 - 冷卻時程可左右接受機率，使SA具有可跳離局部最佳解的機制。因此，只要任一個傳統啟發式解法再加上冷卻時程的機制，即可發揮SA功能。
- SA利用現有狀態推演下一狀態，與之前的狀態無關。
 - 因此，僅需儲存現有狀態之變數值及目標值，不必另外儲存已搜尋過的可行解，故對記憶體空間的要求不高。不必像GA或ACO需儲存一整個族群的可行解，或是TS需儲存禁忌清單。
- 一次僅探討一個可行解，不似GA或ACO一次搜尋一整個族群的可行解，或像具長期記憶機制的TS，在執行中期可針對多個菁英解進行平行處理。因此，SA在求解大規模問題時，通常需要耗費相當大的電腦執行時間。



■ 前提與限制

- 應用SA時必須先找到一個起始可行解，方能以此可行解為基礎，利用鄰域搜尋法尋找下一個可行解。但SA的設計邏輯內並未包含起始解之求解，此部份仍需利用傳統啟發式解法求之。
- 某一問題的鄰域搜尋方法，未必適合其它問題。
- 冷卻時程是SA的核心之一，不同的冷卻時程會導致不同的尋優績效，但目前並沒有一個設計準則可供依循。
 - 參考相關文獻、Try and error。
- 由於SA是由現有狀態推演出下一狀態，一次僅探討一個可行解。因此在求解大規模問題時，通常需要耗費相當大的電腦執行時間。
- 無法確保可覓得全域最佳解。

