شبیه سازی تبرید

Simulated Annealing

Dr. Mostafa Zandieh
Shahid Beheshti University

منشاء SA

- در سال ۱۹۵۳، متروپولیس[1] و همکارانش یک روش جدید برای ساخت کریستالهای جامد با کیفیت عالی ابداع نمودند.
- خلاصه روش:
  - ابتدا کریستال جامد را تا یک دمای بسیار بالا حرارت می دهند (به طوری که ذرات در حالت مذاب بتوانند در کنار هم آزادانه حرکت کنند).
  - سپس دمای مذاب را به صورت بسیار کند و تدریجی کاهش می دهند(در هر دمایی ذرات فقط به اندازه ای جابجا می شوند که بیشترین پایداری را ایجاد کنند).

1- Metropolis

منشاء SA

- این فرآیند سردسازی تدریجی را آنیل کردن[1] می گویند.
  - آنیل یک فرآیند برای بهبود خواص کیفی کریستال جامد است.
  - سرد سازی بسیار کند و تدریجی موجب آزادسازی انرژی و قرارگیری ذرات در جهت با بیشترین پایداری خواهد شد.
  - در مقابل سردسازی سریع موجب شکنندگی بالا در کریستال جامد می شود.
  - در سال ۱۹۸۳، کریک پاتریک[2] و همکارانش شبیه سازی فرآیند آنیل کردن را برای حل مسائل بهینه سازی ترکیبیاتی پیشنهاد کردند.

1- Annealing 2- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.

شباهت بین مسائل بهینه سازی ترکیبیاتی و ترمودینامیک

بهینه سازی ترکیبیاتی	ترمودینامیک
جواب شدنی	حالت[1]
تابع هدف/ تابع ارزیابی	سطح انرژی
جواب بهینه محلی	حالت نهایی
پارامتر کنترلی T	دما
جستجوی محلی	Rapid Quenching
شبیه سازی تبرید	تبرید دقیق/ آنیل دقیق[2]

1- State 2- Careful Annealing

نامهای معادل برای SA

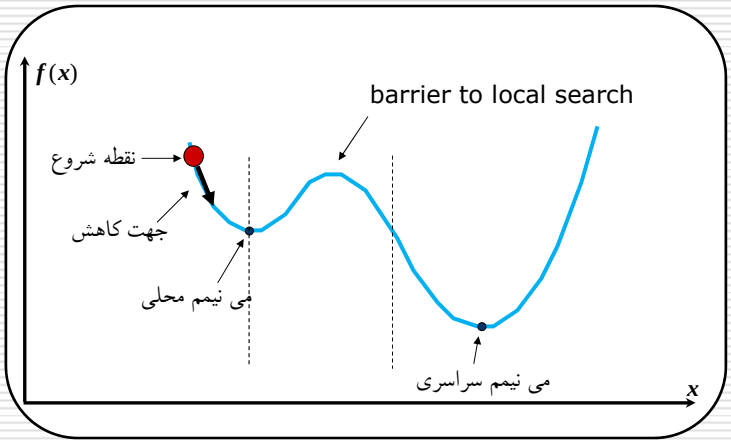
- Mont-Carlo Annealing
- Probabilistic Hill-Climbing
- Probabilistic Cooling
- Probabilistic Exchange Algorithm
- Stochastic Relaxation

روش ابتکاری HC[1]

1. یک جواب شدنی برای مساله ایجاد کنید (به صورت تصادفی).
2. وضعیت جواب فعلی را بر اساس تابع هدف ارزیابی کنید.
3. تمام حرکت های قابل انجام از جواب فعلی را لیست کنید.
4. برای تمام حرکات لیست شده، تابع هدف را محاسبه کنید.
5. بهترین حرکت [2] را (اگر بهبود دهنده باشد) انتخاب و جایگزین جواب فعلی کنید.
6. به قدم ۳ برگردید، مگر این که حرکت بهبود دهنده ای موجود نباشد.

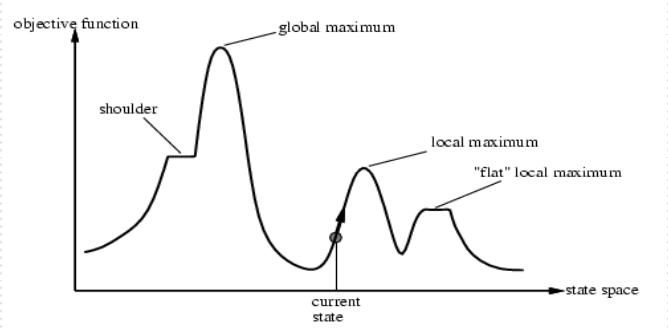
تذکر: این روش ابتکاری، همیشه اوقات به سمت جواب بهتر حرکت می کند.

1- Hill-Climbing 2- Steepest Descent (Ascent)Hill-Climbing



معایب روش ابتکاری HC

- گیر افتادن در دام بهینه ی محلی[1].
- گیر افتادن در دامنه های تخت[2] یک تابع.



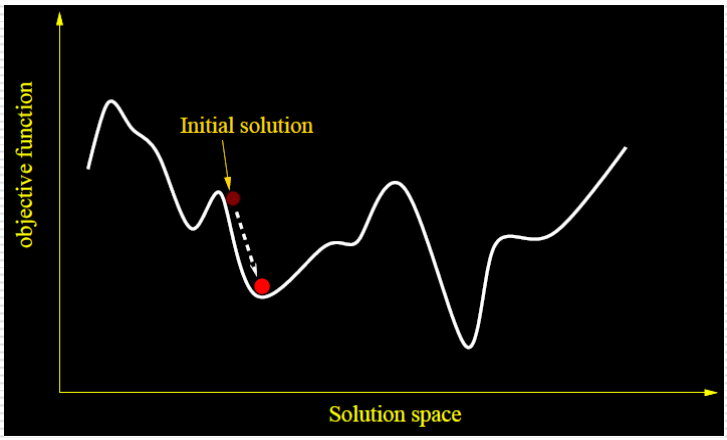
1- Local Optimum 2- Plateu

ایده ی اصلی SA

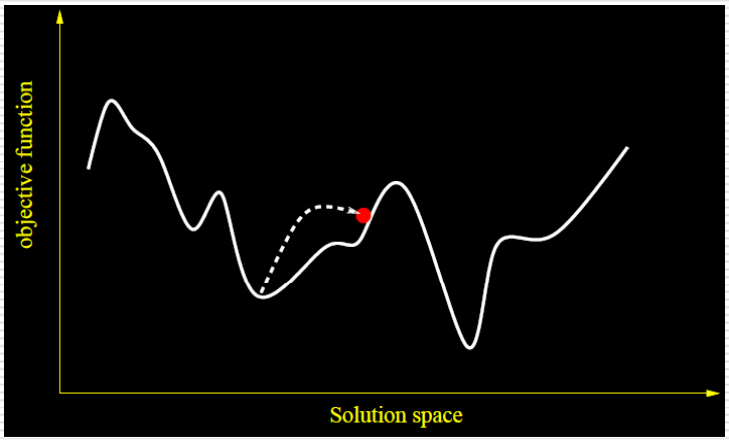
- به جای اینکه همیشه ی اوقات به سمت جواب بهتر حرکت کنیم، تلاش کنیم که در بیشتر اوقات به سمت جواب بهتر حرکت کنیم.
  - برخی اوقات حرکات بد نیز پذیرفته شود، شاید این کار مقدمه ای برای رسیدن به جوابهای بهتر از حد متوسط باشد (تنوع بخشی[1]).
- به منظور فرار از بهینه ی محلی و تضمین همگرایی الگوریتم باید:
  - تعادل مناسبی بین دو مکانیزم تنوع بخشی و تمرکزدهی ایجاد شود.
  - احتمال پذیرش حرکات بد در ابتدای الگوریتم بالا و به تدریج، این احتمال کاهش یابد.
  - در صورت پذیرش حرکات بد، احتمال پذیرش حرکات بدتر در یک مرحله کوچکتر باشد.

1- Diversification

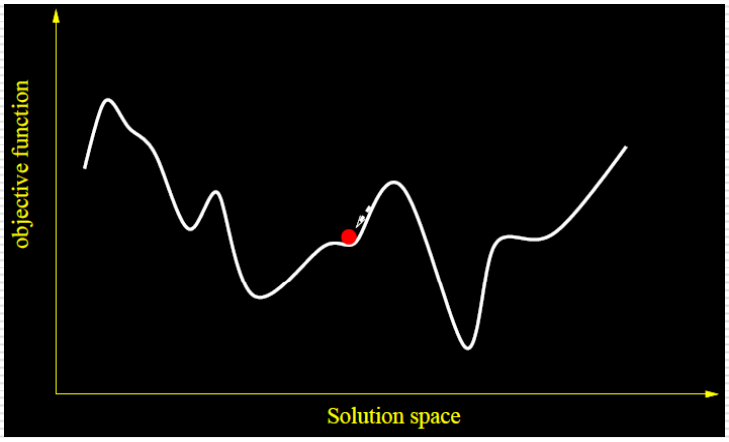
ایده ی اصلی SA



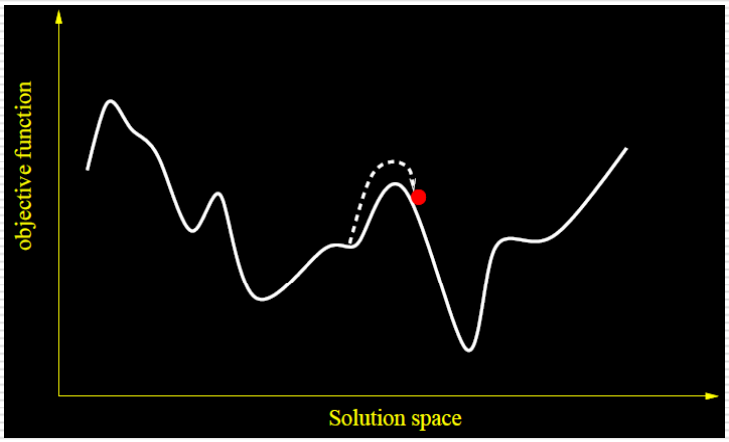
ایده ی اصلی SA



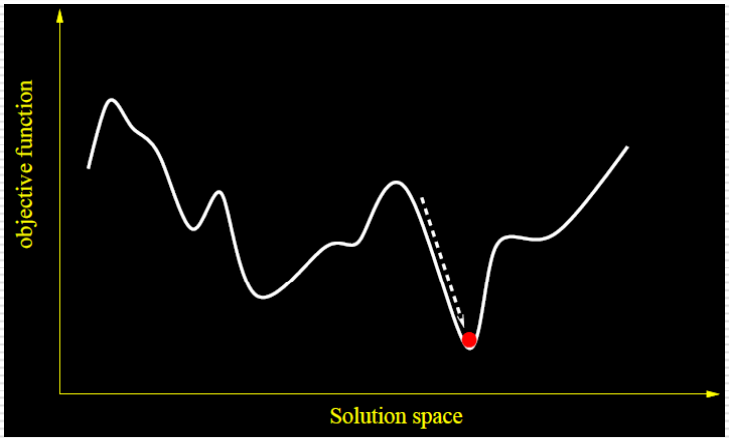
ایده ی اصلی SA



ایده ی اصلی SA



ایده ی اصلی SA



تابع پذیرش بولتزمان[1]

- تابع احتمال بولتزمان

$$p = e^{\frac{-\Delta E}{T}}$$

- احتمال پذیرش دو جواب بد در یک مرحله ی خاص(با یک دمای خاص)

$$(x_c,\ f(x_c)=۲۰۰۰) \rightarrow (x_۱,\ f(x_۱)=۲۱۰۰) \quad \Rightarrow p_۱ = e^{\frac{-\Delta E}{T}} = e^{\frac{-(۲۱۰۰-۲۰۰۰)}{۱۰۰}} = e^{-۱} \simeq ۰.۳۶۸$$

$$(x_c,\ f(x_c)=۲۰۰۰) \rightarrow (x_۲,\ f(x_۲)=۲۲۰۰) \quad \Rightarrow p_۲ = e^{\frac{-\Delta E}{T}} = e^{\frac{-(۲۲۰۰-۲۰۰۰)}{۱۰۰}} = e^{-۲} \simeq ۰.۱۳۵$$

$f(x_i)$	۲۰۵۰	۲۱۰۰	۲۲۰۰	۲۳۰۰	۲۴۰۰	۲۵۰۰	۲۶۰۰
p_i	۰.۶۰۶	۰.۳۶۸	۰.۱۳۵	۰.۰۵۰	۰.۰۱۸	۰.۰۰۷	۰.۰۰۲

1- Boltzmann

تابع پذیرش بولتزمان

- احتمال پذیرش یک جواب بد در دو مرحله ی مختلف از الگوریتم:

$$T_۱=۱۰۰;\ (x_c,\ f(x_c)=۲۰۰۰) \rightarrow (x_۱,\ f(x_۱)=۲۲۰۰) \Rightarrow p_۱ = e^{\frac{-\Delta E}{T}} = e^{\frac{-(۲۲۰۰-۲۰۰۰)}{۱۰۰}} = e^{-۲} \simeq ۰.۱۳۵$$

$$T_۲=۵۰;\ (x_c,\ f(x_c)=۲۰۰۰) \rightarrow (x_۱,\ f(x_۱)=۲۲۰۰) \Rightarrow p_۲ = e^{\frac{-\Delta E}{T}} = e^{\frac{-(۲۲۰۰-۲۰۰۰)}{۵۰}} = e^{-۴} \simeq ۰.۰۱۸$$

T_i	۳۰۰	۲۰۰	۱۵۰	۱۰۰	۷۵	۵۰	۴۰
p_i	۰.۵۱۳	۰.۳۶۸	۰.۲۶۳	۰.۱۳۵	۰.۰۶۹	۰.۰۱۸	۰.۰۰۷

خلاصه الگوریتم SA

- بهینه سازی با یک جواب تصادفی و مقدار بزرگ پارامتر کنترلی **T** آغاز می شود.
  - به عبارتی در ابتدای الگوریتم، درصد زیادی از جوابهای بد پذیرفته می شود.
- در هر دما به اندازه ی کافی تلاش محاسباتی (جستجو) انجام می شود.
  - اگر جواب بهبود یابد مطابق روش HC جواب را قطعا می پذیریم،
  - در غیر اینصورت بر اساس تابع بولتزمان جواب را بصورت احتمالی ممکن است بپذیریم.
- بر اساس یک الگوی خاص، دما کاهش می یابد.
  - مجددا در هر دما به اندازه ی کافی تلاش محاسباتی (جستجو) انجام می شود.
- این فرآیند تا برقراری شرط توقف (معمولا رسیدن به دمای پایانی) انجام می شود.

شبه کد الگوریتم SA

```
Initialization
        Parameters Setting (T0 , Tf , Max_Ite, Cooling_Schedule)
        Generate an Initial_solution (randomly)
        Current_solution = Initial_solution;  Best_solution = Initial_solution;  T= T0
While (T > Tf) do
        For i=1 to Max_Ite do
                Generate  New_solution =Neighbor(Current_solution)
                Evaluate the change in objective function level ΔE
                If ΔE < 0 then
                        Current_solution = New_solution
                        Best_solution = New_solution
                Else
                        Accept  Current_solution = New_solution with probability  p=e^(-ΔE/T)
                End If
        End For
        Add  (Best_solution, Current_solution) to convergence plot(T, i )
        Add  (Best_solution, Current_solution) to convergence plot(T )
        Decrease parameter T according to Cooling_Schedule
End While
Print convergence plot(T, i ) and convergence plot(T )
```

اجزاء الگوریتم SA

- روشی برای نمایش جواب
- روشی برای تولید جواب اولیه
- عملگر حرکت
- تابعی برای ارزیابی جوابها (تابع هدف)
- الگویی برای کاهش دما[1]
- شرط/ شرایط توقف

1- Cooling-Schedule

جواب اولیه[1]

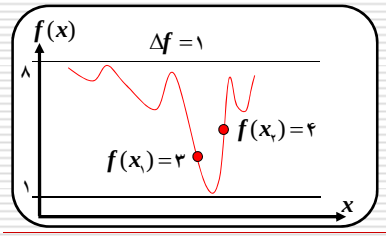
- تصادفی
- مبتنی بر یک روش ابتکاری سازنده
  - روش نزدیک بین[4] در مساله فروشنده ی دوره گرد
  - قواعد توزیع[5] در زمانبندی تولید

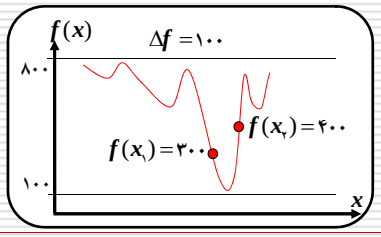
1- Initial Solution 2- Random 3- Constructive 4- Myopic (NBR) 5- Dispatching Rules

تعیین دمای اولیه و دمای نهایی

- دمای اولیه و دمای نهایی به شدت متاثر از مقیاس[1] تغییرات در تابع هدف است.
- با استفاده از تعدادی جواب تصادفی می توان محدوده ی تغییرات تابع هدف را حدس زد.

$$p = e^{\frac{-\Delta E}{T}} \Rightarrow \ln p = \frac{-\Delta E}{T} \Rightarrow T = \frac{-\Delta E}{\ln p}$$





1- Scale

تعیین دمای اولیه و دمای نهایی

- مثال ۱:

$$p = e^{\frac{-\Delta E}{T}} \Rightarrow \ln p = \frac{-\Delta E}{T} \Rightarrow T = \frac{-\Delta E}{\ln p}$$

$$p_{\max} = ۰.۹۹ \; ; \qquad p_{\min} = ۰.۰۱ \; ; \qquad \Delta E \in (۰.۰۱ , ۰.۰۵)$$

$$\Rightarrow T_۱ = \frac{-۰.۰۱}{\ln ۰.۹۹} \simeq ۰.۹۹ ; \quad T_۲ = \frac{-۰.۰۵}{\ln ۰.۹۹} \simeq ۴.۹۷ ; \quad T_۳ = \frac{-۰.۰۱}{\ln ۰.۰۱} \simeq ۰.۰۰۲ ; \quad T_۴ = \frac{-۰.۰۵}{\ln ۰.۰۱} \simeq ۰.۰۱$$

$$\Rightarrow T_o = \max\{T_۱ \simeq ۰.۹۹, \quad T_۲ \simeq ۴.۹۷, \quad T_۳ \simeq ۰.۰۰۲, \quad T_۴ \simeq ۰.۰۱\} + \varepsilon = ۵$$

$$\Rightarrow T_f = \min\{T_۱ \simeq ۰.۹۹, \quad T_۲ \simeq ۴.۹۷, \quad T_۳ \simeq ۰.۰۰۲, \quad T_۴ \simeq ۰.۰۱\} - \varepsilon = ۰.۰۰۱$$

تعیین دمای اولیه و دمای نهایی

- مثال ۲:

$$p = e^{\frac{-\Delta E}{T}} \Rightarrow \ln p = \frac{-\Delta E}{T} \Rightarrow T = \frac{-\Delta E}{\ln p}$$

$$p_{\max} = ۰.۹۹ \; ; \quad p_{\min} = ۰.۰۱ \; ; \quad \Delta E \in (۴۰ , ۵۰)$$

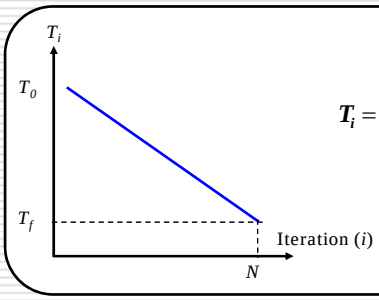
$$\Rightarrow T_۱ = \frac{-۴۰}{\ln ۰.۹۹} \simeq ۳۹۸۰ ; \quad T_۲ = \frac{-۵۰}{\ln ۰.۹۹} \simeq ۴۹۷۵ ; \quad T_۳ = \frac{-۴۰}{\ln ۰.۰۱} \simeq ۸.۸ ; \quad T_۴ = \frac{-۵۰}{\ln ۰.۰۱} \simeq ۱۱$$

$$\Rightarrow T_o = \max\{T_۱ \simeq ۳۹۸۰, \quad T_۲ \simeq ۴۹۷۵, \quad T_۳ \simeq ۸.۸, \quad T_۴ \simeq ۱۱\} + \varepsilon = ۵۰۰۰$$

$$\Rightarrow T_f = \min\{T_۱ \simeq ۳۹۸۰, \quad T_۲ \simeq ۴۹۷۵, \quad T_۳ \simeq ۸.۸, \quad T_۴ \simeq ۱۱\} - \varepsilon = ۸$$

الگوهای سردسازی دما

- الگوی خطی[1]:

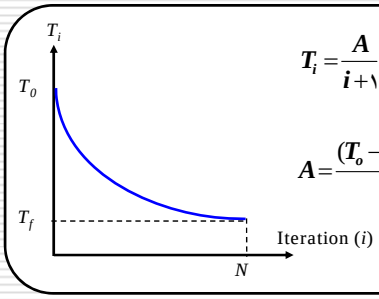


$$T_i = T_o - i\left(\frac{T_o - T_f}{N}\right)$$

1- Linear

الگوهای سردسازی دما

- الگوی نمایی[1]:



$$T_i = \frac{A}{i+1} + B$$

$$A = \frac{(T_o - T_f)(N+1)}{N}; \quad B = T_o - A$$

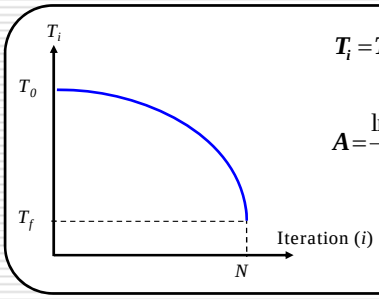
$$T_i = \alpha T_{i-1}$$

$$0 < \alpha < 1$$

1- Exponential

الگوهای سردسازی دما

- الگوی لگاریتمی[1]:



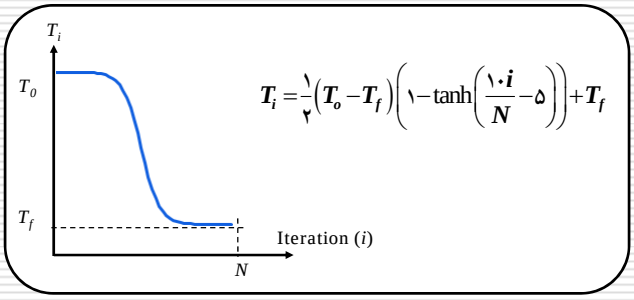
$$T_i = T_o - i^A$$

$$A = \frac{\ln(T_o - T_f)}{\ln(N)}$$

1- Logarithmic

الگوهای سردسازی دما

- الگوی هیپربولیک[1]:



$$T_i = \frac{۱}{۲}\left(T_o - T_f\right)\left(۱ - \tanh\left(\frac{۱۰ \cdot i}{N} - ۵\right)\right) + T_f$$

1- Hyperbolic

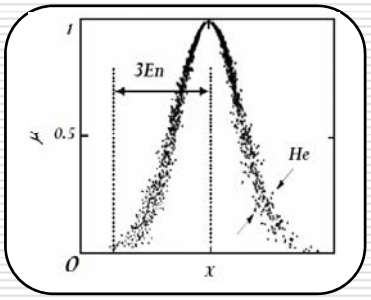
برخی نکات تکمیلی در مورد SA

- الگوریتم شبیه سازی تبرید، یک الگوریتم بهبود دهنده ی مبتنی بر نقطه[1] است.
- تعداد پارامترهای الگوریتم شبیه سازی تبرید، نسبت به سایر الگوریتم های فراابتکاری کمتر است.
- نسخه ی پیشرفته ای از الگوریتم شبیه سازی تبرید بنام الگوریتم شبیه سازی تبرید مبتنی بر تئوری ابر[2] (**CSA**) وجود دارد که دارای دو خصوصیت مهم است:
  - دمای هر مرحله متغیری تصادفی است که در هر مرحله میانگینش کاهش می یابد.
  - میانگین دمای هر مرحله نیز متغیری تصادفی است (منشاء ایجاد ابر).
  - واریانس این متغیر تصادفی به تدریج کاهش می یابد.

1- Point-Based 2- Cloud-Based Simulated Annealing

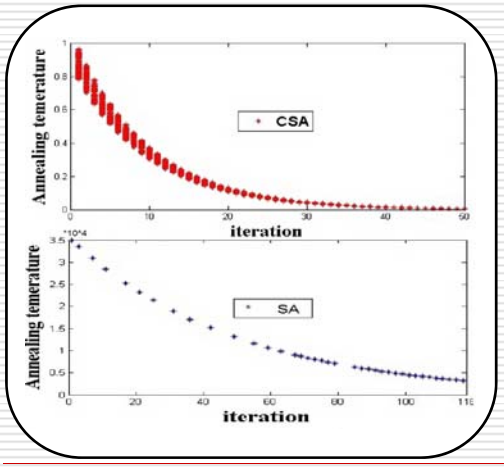
برخی نکات تکمیلی در مورد SA

- ابر مجموعه ای از اعداد تصادفی است که از قوانین توزیع نرمال پیروی می کند.
- ابر با سه مشخصه زیر تعریف می شود.
  - Ex : مرکز ابر (امید ریاضی)
  - En :محدوده ی پوشش ابر (آنتروپی)
  - He :درجه ی پراکندگی قطرات ابر نسبت به یکدیگر (سوپر آنتروپی)



برخی نکات تکمیلی در مورد SA

- مقایسه ی رویه ی کاهش دما در SA و CSA



برخی نکات تکمیلی در مورد SA

- در هر مرحله از حلقه ی خارجی میانگین دما به صورت نمایی کاهش می یابد.

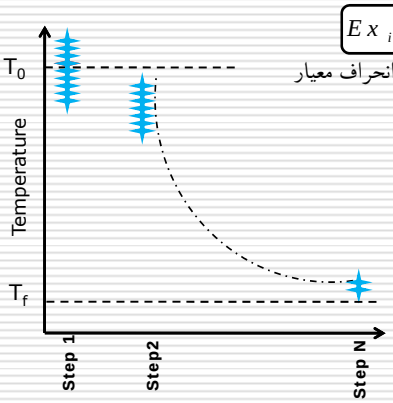
$$Ex_i = \lambda^i . Ex_0 ; \quad 0 < \lambda < 1$$

- به منظور تشکیل ابر، هر بار در حلقه ی داخلی انحراف معیار از توزیع نرمال تولید می شود.

$$En'_j \sim N(\mu_\sigma = En,\ \sigma_\sigma = He)$$

- در حلقه ی داخلی، دما از توزیع نرمال تولید می شود.

$$T_{ji} \sim N\left(\mu_{T_i} = Ex_i,\ \sigma_{T_i} = En'_j\right)$$



References

[1] Herault, L. 2000. Rescaled simulated annealing- Accelerating convergence of SA by rescaling the state energy. Journal of Heuristics. Vol. 6, No. 1, pp. 111 – 134.

[2] Laguna, M. 2000. Tuning strategies in constrained SA for non-linear global optimization. International Journal of Artificial Intelligent Tools.

[3] Laarhoven, P.J.M., Arts, E. 1987. Simulated annealing: Theory and applications. Journal of Heuristics. John Wiley & Sons, Inc.

[4] Arts, E., Korst, J. 1990. Simulated annealing and Boltzman machine. John Wiley & Sons, Inc.

[5] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. 1983. “Optimization by Simulated Annealing. Science, Vol. 220, No. 4598, pp. 671-680.