Otimização de Redes

Simulated Annealing & Tabu Search

Prof. Lucas S. Batista

lusoba@ufmg.br

www.ppgee.ufmg.br/∼lusoba

Universidade Federal de Minas Gerais
Escola de Engenharia

Sumário

Problema de Otimização Mono-objetivo

- Formulação geral:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}, \ \boldsymbol{x} \in \mathcal{F}$$

$$\mathcal{X} = \{\boldsymbol{x} = (x_1, x_2, \ldots, x_n), \ x_i \in \mathcal{D}_i\}$$

$$\mathcal{F} = \begin{cases} g_i(\boldsymbol{x}) \leq 0; & i = 1, \ldots, p \\ h_j(\boldsymbol{x}) = 0; & j = 1, \ldots, q \\ \boldsymbol{x} \in \mathcal{X} \end{cases}$$

Sumário

Introdução

- *Simulated Annealing* (SA) foi proposto por três pesquisadores da IBM em 1982 (S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, 1983).

- Na ocasião estavam tentando resolver um problema de disposição ótima de componentes eletrônicos em uma placa de circuitos.

Introdução

- SA é uma metaheurística de busca local usada para tratar problemas de otimização discreta e, em menor escalar, contínua.
- Sua principal característica é a habilidade de escapar de mínimos locais, através da aceitação de “movimentos” de piora da qualidade da solução.
- Tornou-se muito popular na década de 90 devido à fácil implementação e propriedades de convergência atrativas.

História e Motivação

- SA é inspirado no processo de "recozimento" físico de sólidos:
  - Um sólido é aquecido e, em seguida, resfriado em estágios, lentamente, até atingir a configuração cristalina mais regular possível (i.e., estado de menor energia), a qual é livre de defeitos.
- SA estabelece uma conexão entre esse tipo de comportamento termodinâmico e o processo de busca por mínimos globais de problemas de otimização discreta.

Funcionamento do Algoritmo

- Em cada iteração do SA, duas soluções candidatas são comparadas (a atual e a modificada):
  - soluções melhoradas são sempre aceitas;
  - uma parcela das soluções inferiores é aceita, com o intuito de escapar de ótimos locais.
- A probabilidade de aceitação de soluções inferiores depende do parâmetro “temperatura”, o qual é usualmente não-crescente ao longo das iterações.
- À medida que o parâmetro “temperatura” tende a zero, movimentos de piora ocorrem com menor frequência e o método converge para um ótimo local, que pode ou não ser global.

Funcionamento do Algoritmo

- Assuma as seguintes definições[1]:
  - Ω: espaço de soluções (conjunto de todas as soluções possíveis);
  - $f : \Omega \to \mathbb{R}$: uma função objetivo definida em Ω;
  - $\mathcal{N}(\boldsymbol{x})$: função de vizinhança da solução $\boldsymbol{x} \in \Omega$;
  - $\boldsymbol{x}^*$: mínimo global, i.e., $\boldsymbol{x}^* \in \Omega$ tal que $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \forall\, \boldsymbol{x} \in \Omega$.

[1] A menos que seja dito o contrário, será considerado o SA para problemas de otimização discreta.

Funcionamento do Algoritmo

- Evolução geral do SA:
  - Começa a busca a partir de uma solução inicial $\boldsymbol{x} \in \Omega$;
  - Uma nova solução candidata $\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x})$ é gerada;
  - Emprega-se o critério de aceitação de Metropolis (1953), que modela como um sistema termodinâmico move-se do estado corrente, $\boldsymbol{x} \in \Omega$, para um novo estado, $\boldsymbol{x}' \in \Omega$, de menor energia.

Funcionamento do Algoritmo

- A probabilidade de aceitação da solução candidata, $\boldsymbol{x}'$, como a alternativa corrente é dada por:

$$P\{\boldsymbol{x}'\} = \begin{cases} exp[-(f(\boldsymbol{x}') - f(\boldsymbol{x}))/t_k] & se \;\; f(\boldsymbol{x}') - f(\boldsymbol{x}) > 0 \\ 1 & se \;\; f(\boldsymbol{x}') - f(\boldsymbol{x}) \leq 0 \end{cases}$$

em que t_k é o parâmetro temperatura no estágio k, tal que:

$$t_k > 0 \;\; \forall \, k \;\; \text{e} \;\; \lim_{k \to \infty} t_k = 0 \; .$$

- Note que em altas temperaturas, $exp[-(f(\boldsymbol{x}') - f(\boldsymbol{x}))/t_k] \to 1$, e em baixas, $exp[-(f(\boldsymbol{x}') - f(\boldsymbol{x}))/t_k] \to 0$.

Funcionamento do Algoritmo

- A regra de redução da temperatura possui a seguinte forma:
  - Regra geométrica (amplamente aceita, muito simples)

$$t_{k+1} = \alpha t_k, \quad \alpha \text{ é uma constante menor que 1.}$$

  - Regra adaptativa (mais efetiva)

$$t_{k+1} = \alpha(t_k) t_k$$

$$t_{k+1} = \left(1 - t_k \frac{\Delta(t_k)}{\sigma^2(t_k)}\right) t_k, \quad t_{k+1} = \min\left(D_0, \frac{E_k}{\overline{E}_k}\right) t_k, \quad \text{etc.}$$

em que $\sigma^2(t_k)$ é o desvio padrão das soluções aceitas no estágio k, $\Delta(t_k)$ depende da regra adaptativa definida, $D_0 \in [0.5, 0.9]$, E_k é o menor $f(\cdot)$ aceito durante o estágio k e $\overline{E}_k$ é o valor médio de todas as soluções aceitas no o estágio k.

Pseudocódigo do Algoritmo

Algoritmo 1: Simulated Annealing

Defina um contator $k = 0$;
Defina uma temperatura inicial $t_k \geq 0$;
Defina T_k (função que controla a variação da temperatura);
Defina M_k (no. de iterações executadas na temperatura t_k);
Selecione uma solução inicial $\boldsymbol{x} \in \Omega$;
while *critério de parada não alcançado* **do**
    Defina o contator $m = 0$;
    while $m \leq M_k$ **do**
        Gere uma solução $\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x})$;
        Calcule $\Delta E = f(\boldsymbol{x}') - f(\boldsymbol{x})$;
        if $\Delta E \leq 0$ **then**
            $\boldsymbol{x} \leftarrow \boldsymbol{x}'$;
        else
            $\boldsymbol{x} \leftarrow \boldsymbol{x}'$ com probabilidade $exp(-\Delta E / t_k)$;
        $m \leftarrow m + 1$;
    $t_{k+1} \leftarrow T_k(t_k)$;
    $k \leftarrow k + 1$;

Considerações Práticas

- Algumas escolhas são fortemente dependentes do problema, e influenciam diretamente a eficiência do SA:
  - função objetivo;
  - mapeamento das soluções candidatas;
  - função de vizinhança;
  - "tamanho" da vizinhança[2].

[2] Alguns autores sugerem a implementação do SA com busca em vizinhança variável. Resultados numéricos sugerem um aumento do desempenho do método.

Considerações Práticas

- Definição da temperatura inicial t_0:
  - realize 100 perturbações em $\boldsymbol{x}_0$, e obtenha o valor médio $\overline{\Delta E}$;
  - escolha uma taxa de aceitação inicial τ_0, e.g., $\tau_0 = 0.5$ ou $\tau_0 = 0.2$;
  - deduza t_0 a partir da relação $\exp(-\Delta E / t_0) = \tau_0$.
- Critério para alteração da temperatura:
  - $12n$ perturbações aceitas, ou $100n$ perturbações testadas.
- Regra de redução da temperatura:
  - $t_{k+1} = 0.9 t_k$
- Critério de parada do método:
  - 03 estágios sucessivos de temperatura sem melhora.

Extensão para Problemas Contínuos

- A maioria das aplicações com SA são para problemas discretos, entretanto, existem algumas para tratar problemas contínuos.
- A principal dificuldade relaciona-se com a modelagem da “discretização” do espaço de busca.
- Inúmeros trabalhos discutem as propriedades de convergência do SA aplicado a problemas contínuos de otimização global.

Extensão para Problemas Contínuos

Simulated Annealing para problemas contínuos:

1. Geração de novas soluções:
   $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$, em que $\boldsymbol{\delta} = \sigma(\boldsymbol{x}^{ub} - \boldsymbol{x}^{lb})\mathcal{D}(0,1)$, $\mathcal{D}(0,1) = 2rand(n,1) - 1$

2. As n variáveis são modificadas em grupos de $p \cong n/3$:
   $\boldsymbol{x}' = \boldsymbol{x} + H\boldsymbol{\delta}$, em que H é uma matriz diagonal com 1s nas posições a serem modificadas.

3. Regra de atualização de σ:
   se $A_k > 0.2$, $\sigma = 2\sigma$;
   se $A_k < 0.05$, $\sigma = 0.5\sigma$;
   em que A_k é a taxa de aceitação no estágio k.

4. O desvio padrão inicial é $\sigma = 0.25$, sendo atualizado ao fim de cada estágio de temperatura.

Vantagens e Desvantagens do Método

- Vantagens:
    - Geralmente encontra uma solução de boa qualidade;
    - Sua aplicação e adaptação é muito flexível;
    - Possui fácil implementação.
- Desvantagens:
    - Alto número de parâmetros de controle;
    - O ajuste de parâmetros é dependente do problema;
    - O custo computacional pode ser alto em algumas aplicações.

Sumário

Introdução

- *Tabu Search* (TS) foi proposta por Fred Glover (1986) para a solução de problemas de otimização combinatória.
- Assim como o SA, TS é considerada uma metaheurística, i.e., uma estratégia geral para guiar e controlar heurísticas subordinadas especializadas.
- Tornou-se muito popular na década de 90 principalmente por:
    - apresentar grande efetividade,
    - determinar soluções muito próximas do ótimo global, e
    - lidar com problemas de grande escala.

Introdução

- O princípio básico da TS consiste na aplicação de busca local ("local search" - LS) e na aceitação de movimentos de piora da qualidade da solução.

- Um mecanismo de "memória"[3] (chamado *tabu list*) é usado para evitar revisitação de soluções conhecidas.

[3] Conceito chave relacionado à teoria de inteligência artificial.

Espaço de Busca e Estrutura de Vizinhança

- Assuma as seguintes definições[4]:
  - Ω: espaço de soluções (conjunto de todas as soluções possíveis);
  - $f : \Omega \rightarrow \mathbb{R}$: uma função objetivo definida em Ω;
  - $\mathcal{N}(\boldsymbol{x})$: função de vizinhança da solução $\boldsymbol{x} \in \Omega$;
  - $\boldsymbol{x}^*$: mínimo global, i.e., $\boldsymbol{x}^* \in \Omega$ tal que $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \forall\, \boldsymbol{x} \in \Omega$.

[4] A menos que seja dito o contrário, será considerado o TS para problemas de otimização discreta.

Espaço de Busca e Estrutura de Vizinhança

- O domínio de busca pode ser mapeado de diferentes formas para um mesmo problema.
    - Uma escolha mal sucedida dessa representação pode comprometer muito o desempenho do método.
    - Por exemplo, em muitos casos pode ser desejável/necessário pesquisar em regiões infactíveis.
- De forma similar, em geral existem inúmeras estruturas de vizinhança possíveis para um mesmo mapeamento do domínio de busca.
- A definição do espaço de busca e da estrutura de vizinhança representa o passo mais crítico no projeto de qualquer TS.

Tabus

- *Tabus* são usados para evitar operações cíclicas, i.e., a revisitação de soluções.
- Essa “inteligência” é implementada por meio de listas tabus que armazenam *movimentos tabus*.
- Esse mecanismo permite ao TS escapar de ótimos locais e, simultaneamente, a explorar regiões ainda não visitadas.

Tabus

- Múltiplas listas tabus podem ser utilizadas simultaneamente, e em alguns casos são aconselhadas, e.g., quando diferentes tipos de movimentos são empregados para pesquisar a vizinhança.
- Listas tabus de tamanho fixo frequentemente não previnem revisitações:
  - Recomenda-se variar o comprimento das listas durante a busca;
  - Outra opção é a geração aleatória da "duração" tabu de cada movimento.

Critérios de Aspiração

- *Tabus* são, algumas vezes, muitos fortes!
  - Eles podem proibir movimentos atrativos, mesmo quando não há risco de revisitação, ou podem levar à estagnação do método.
- São necessárias então técnicas que permitam revogar os tabus! Essas estratégias são chamadas *critérios de aspiração*.
  - Em geral, um movimento é aceito (mesmo sendo tabu) se ele resulta em uma solução melhor do que a melhor alternativa conhecida.

Pseudocódigo do Algoritmo

- $\boldsymbol{x}$, a solução corrente;
- $\boldsymbol{x}^*$, a melhor solução conhecida;
- f^*, o valor de $f(\boldsymbol{x}^*)$;
- $\mathcal{N}(\boldsymbol{x})$, a vizinhança de $\boldsymbol{x}$;
- $\tilde{\mathcal{N}}(\boldsymbol{x})$, o subconjunto admissível de $\mathcal{N}(\boldsymbol{x})$[5];
- T, a lista tabu.

Algoritmo 2: Tabu Search

```
Selecione uma solução inicial x_0 ∈ Ω;
Defina x ← x_0, x* ← x_0, f* ← f(x_0), T ← ∅;
while critério de parada não alcançado do
    Determine x = arg min_{x' ∈ Ñ(x)} f(x');
    if f(x) < f* then x* ← x, f* ← f(x) ;
    Atualize a lista tabu T (delete entradas antigas se necessário);
```

[5] Conjunto não-tabu ou aceito por aspiração.

Busca Local

- A busca local, $\boldsymbol{x} = \arg\min_{\boldsymbol{x}' \in \tilde{\mathcal{N}}(\boldsymbol{x})} f(\boldsymbol{x}')$, pode ser implementada de diferentes formas:
  - *best improvement*;
  - *first improvement*;
  - TS probabilístico e listas candidatas.

Busca Local

- Visando minimizar o custo computacional, é frequente a presença de um TS probabilístico e listas candidatas.
  - No TS probabilístico, uma amostra aleatória $\mathcal{N}'(\boldsymbol{x}) \subset \mathcal{N}(\boldsymbol{x})$ substitui o conjunto completo $\mathcal{N}(\boldsymbol{x})$.
  - Essa aleatoriedade na busca minimiza problemas de revisitação;
  - As listas tabus podem ser menores;
  - Soluções "excelentes" podem ser perdidas!
  - Listas candidatas "contornam" a limitação anterior, a partir da geração de um subconjunto $\mathcal{N}'(\boldsymbol{x})$ mais promissor.

Critérios de Parada do Algoritmo

- Os critérios de parada mais comuns no TS são:
  - máximo número de iterações;
  - máximo custo computacional;
  - número específico de iterações sem melhora;
  - determinação de um limiar preestabelecido para $f(\cdot)$.

Elementos Adicionais

- Intensificação:
  - busca local intensa, aplicada de tempos em tempos, usando uma lista candidata, uma estrutura de vizinhança variável, ou um sub-conjunto de vizinhança mais amplo (no caso do TS probabilístico).
- Diversificação:
  - busca exploratória por regiões não investigadas;
  - *restart diversification* – promove a inserção, na solução atual, de alguns movimentos raramente usados até a iteração corrente, e reinicia a busca a partir dessa alternativa;
  - *continuous diversification* – emprega "memória de frequência" durante a evolução do algoritmo para polarizar a utilização de movimentos pouco explorados.

Elementos Adicionais

- Presença de soluções inviáveis:
  - a eliminação de todas as soluções inviáveis pode restringir muito o espaço de busca, e levar a soluções finais ruins;
  - técnicas de relaxação das restrições ampliam o domínio de busca que, geralmente, poderá ser explorado com uma estrutura de vizinhança mais simples;
  - essas técnicas são facilmente implementadas por meio da adição de funções penalidade ponderadas à função objetivo.

Elementos Adicionais

- Modelos de aproximação local:
  - a avaliação de $f(\cdot)$ de alguns problemas é proibitiva;
  - visando minimizar os custos, pode-se avaliar soluções vizinhas usando uma função objetivo aproximada, $\tilde{f}(\cdot)$, de baixo custo;
  - após a exploração dessa vizinhança, e a identificação de um subconjunto promissor de soluções, estas são avaliadas no objetivo original $f(\cdot)$ para se determinar a melhor alternativa.

Literatura Especializada

M. Gendreau, J.-Y. Potvin (eds.), Handbook of Metaheuristics, Springer, 2nd ed., 2010.

J. Dréo, P. Siarry, A. Pétrowski, E. Taillard, Metaheuristics for Hard Optimization: Methods and Case Studies, Springer, 2006.

Início