

Otimização do plano de geração de energia hidrelétrica através de métodos de busca local

Bruno Nocera Zanette

Universidade Federal do Paraná

bnzanette@inf.ufpr.br

Setembro, 2016

Sumário

- 1 **Introdução**
 - Objetivo
 - Informações sobre o SIN
 - Geração de energia hidrelétrica
- 2 **Problema do planejamento**
 - Problema
 - Simulação
- 3 **Revisão bibliográfica**
 - Algoritmos de busca clássicos
 - Método de Monte Carlo
- 4 **Proposta**
 - Estado atual do trabalho
 - Resultados preliminares
 - Propostas
 - Cronograma de atividades

Objetivo

Reduzir o uso de outras fontes mais caras e poluentes, mantendo a produção de energia hidrelétrica sempre elevada.

Como?

Otimizando um plano mensal de geração de energia hidrelétrica de um período de 5 anos.

Dificuldades

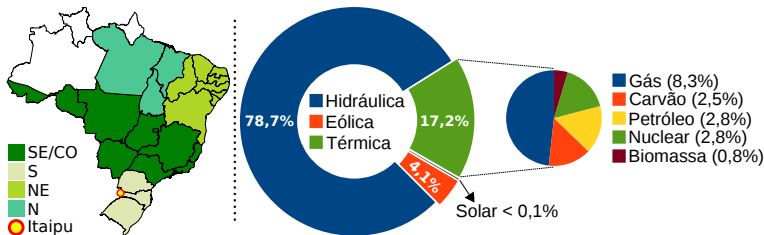
- O sistema é composto de equações não-lineares
- Restrições que só podem ser validadas após a execução das ações
- No total são mais de 12000 ações a serem otimizadas
- A variação das chuvas impede o reaproveitamento de ações
- Cada uma dessas usinas possui um potencial energético diferente

Metodologia

- Implementar um otimizador de planos de geração de energia hidrelétrica com base em técnicas de busca local clássicas capazes de buscar planos melhores baseados numa função fitness.

Informações sobre o SIN

- Sistema Interligado Nacional (SIN) é controlado pelo Operador Nacional do Sistema Elétrico (ONS)
- Dividido em 5 subsistemas: SE/CO, S, NE, N e Itaipu
- Utiliza diversas fontes diferentes de energia em conjunto para garantir o fornecimento ininterrupto de energia.



Problema

- O plano precisa ser factível
- As restrições afetam as variáveis do sistema de diferentes modos:
 - Diretamente: Quantidade de água vertida e turbinada
 - Indiretamente: Nível do reservatório, que depende de várias outras variáveis para ser calculado.
- Ou seja, tanto a validação do plano quanto o cálculo da quantidade de energia gerada envolvem funções não-lineares cujas variáveis só são instanciadas durante a simulação do sistema hidrelétrico.
- A princípio, não há outra forma de abordar esse problema sem usar um simulador.

Componentes da simulação

- Cálculo do volume do reservatório
- Restrições
- Cálculo da energia gerada
- Qualidade do plano
- Rede de transmissão

- Nas fórmulas que serão exibidas:
 - u = Usina hidrelétrica
 - p = Período (mês)

Cálculos da simulação

Cálculo do volume do reservatório

$$Vol[u] = (Vol[u] + Ch[u, p]) - (V[u, p] + T[u, p]) + \sum_{m=1}^{UMon} (V[m, p] + T[m, p]) \quad (1)$$

- Vol = Volume do reservatório.
- Ch = Quantidade de chuva prevista para o período.
- V = Quantidade de água vertida.
- T = Quantidade de água turbinada.

Cálculos da simulação

Restrições

$$\begin{aligned} \min_V[u] &\leq V[u, p] \\ 0 &\leq T[u, p] \leq \max_T[u] \\ \min_Vol[u] &\leq Vol[u] \leq \max_Vol[u] \end{aligned} \quad (2)$$

- \min_V = Quantidade mínima de água que deve ser vertida.
- \max_T = Quantidade máxima de água que pode ser turbinada.
- \min_Vol = Volume mínimo do reservatório.
- \max_Vol = Volume máximo do reservatório.

Cálculos da simulação

Cálculo da energia gerada

$$E = AI * P * T \quad (3)$$

- AI = Desperdício de água durante a queda devido a ineficiências do sistema.
- P = Produtividade da usina hidrelétrica.
- T = Quantidade de água turbinada.

Qualidade do plano

- Função que calcula o quão perto de um objetivo pré-definido o resultado da simulação desse plano está
- Valor resultante define o quão bom é um plano, sem que seja necessário armazenar ou processar mais nenhuma outra informação sobre o mesmo
- Neste trabalho o objetivo definido é: Reduzir ao máximo a diferença absoluta entre a quantidade de energia elétrica gerada e demandada

Rede de transmissão

- Como não é possível, por questões estruturais e geográficas, que todos os subsistemas consigam suprir a própria demanda, é necessário que haja uma rede de transmissão de energia, para que se possa transferir a energia gerada por um subsistema para outro
- Além disso, alguns subsistemas podem não ter demanda, tendo como único propósito gerar e distribuir energia hidrelétrica para outros subsistemas.
 - Como é o caso do subsistema Itaipu no SIN.

Resumo

- É necessário ter um simulador de planos capaz de verificar o cumprimento das restrições e de calcular a quantidade de energia gerada por cada usina hidrelétrica
- A qualidade do plano é calculada por uma função dadas as quantidades de energia elétrica produzida e demandada por cada subsistema.
- Para buscar por planos melhores serão usadas técnicas de busca local clássicas de inteligência artificial

Subida de encosta e Têmpera simulada

Subida de encosta

- Iteração que continuamente se move na direção do valor mais alto, e termina ao chegar num ponto onde nenhum de seus vizinhos possuem valores maiores que o valor corrente

Têmpera simulada

- Alia aleatoriedade ao algoritmo subida de encosta
- Baseado no processo metalúrgico de temperar, ou enrijecer, um metal ou vidro que é obtido esquentando o material até uma alta temperatura e depois esfriando-o gradualmente, fazendo com que o mesmo atinja um estado cristalino de baixa energia [Russell e Norvig, 1995]

Busca em feixe local e algoritmos genéticos

Busca em feixe local

- Armazena mais de um estado ao mesmo tempo.
- k indivíduos aleatórios são gerados. Todos os sucessores desses indivíduos são gerados e avaliados. Os k melhores são selecionados e assim o processo é reiniciado.

Algoritmos genéticos

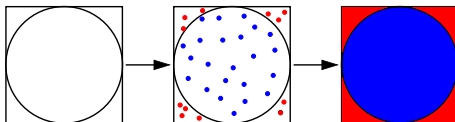
- Se inspiram na reprodução sexuada, diferente do algoritmo de busca em feixe local. Os novos indivíduos são gerados a partir da combinação de dois indivíduos.
- Essa combinação é feita em 3 (três) etapas principais: seleção, cruzamento e mutação.

Método de Monte Carlo

- Processos estocásticos, nos quais suas conclusões são obtidas a partir da observação de eventos aleatórios e podem ser usados para resolver tanto problemas determinísticos quanto probabilísticos [Hammersley, 2013].
- Não deve ser utilizada exclusivamente como uma caixa-preta [Andrieu et al., 2003]. Ao incorporar um conhecimento prévio do domínio do problema ao design da solução é possível obter resultados mais significativos.
- A precisão dos resultados depende diretamente da quantidade de eventos aleatórios observados. Quanto mais eventos forem observados maior será a precisão.

Exemplos de aplicação

- Probabilístico: Calcular a probabilidade de cair um certo número num lançamento de um dado
- Determinístico: Calcular o valor de π , com base apenas nas equações das áreas do quadrado e circunferência



Capacidade de realizar buscas locais

- São capazes de realizar buscas. As observações também podem ser vistas como uma espécie de busca aleatória.
- É ineficiente pois no pior caso pode acabar fazendo mais comparações do que o próprio tamanho do espaço de busca.

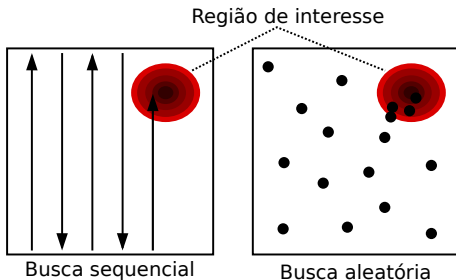


Figura: Diferentes métodos de busca de região

Algoritmos implementados

- **Simulador do sistema hidrelétrico**
 - Simula todas as ações, verifica o cumprimento das restrições e calcula a quantidade de energia elétrica produzida por cada usina
- **Gerador de planos**
 - Modifica um plano original, acrescentando às ações uma fração positiva ou negativa das mesmas
- **Função fitness**
 - Calcula a qualidade do plano dadas as quantidades de energia elétrica produzida e demandada por cada subsistema
- **Buscador de planos**
 - Simula e verifica a qualidade de uma certa quantidade de planos gerados pelo gerador de planos, e retorna o melhor
- **Otimizador**
 - Controla os parâmetros de execução do gerador de planos e define as condições de parada do algoritmo

Simulador do sistema hidrelétrico

- O simulador foi implementado com base no modelo de despacho hidrotérmico sugerido pelo projeto PHOENIX [Bessa et al., 2013]
- As usinas foram modeladas usando um grafo direcionado em forma de árvore
- As séries de chuvas usadas são sintéticas, geradas com base no modelo hidrológico CARMA, descrita em [Hipel e McLeod, 1994] e citada em [Bessa et al., 2013]
- Cada usina possui uma série de chuva
- Para a busca foi implementado um algoritmo de busca híbrido, composto pelos algoritmos método de Monte Carlo (buscador) e Têmpera simulada (otimizador)

Simulador do sistema hidrelétrico

Algoritmo 1 Algoritmo do simulador

```
1: for  $p \leftarrow 1$  to 60 do
2:   for all  $u \leftarrow \text{ordenacaoTopologica}(\text{conjuntoUsinas})$  do
3:     if  $V[u, p] < \text{min\_V}[u]$  or  $T[u, p] > \text{max\_T}[u]$  then
4:        $\text{planoValido} \leftarrow \text{false}$ ;
5:       return erro
6:     end if
7:      $\text{usinasMontante} \leftarrow \text{selecionaUsinasMontante}(u, \text{conjuntoUsinas})$ 
8:      $\text{Vol}[u] \leftarrow \text{calculaVolume}(u, \text{usinasMontante})$ 
9:     if  $\text{Vol}[u] < 0.0$  or  $\text{Vol}[u] > \text{max\_Vol}[u]$  then
10:       $\text{planoValido} \leftarrow \text{false}$ 
11:      return erro
12:    end if
13:     $E[u, p] \leftarrow \text{calculaEnergiaGerada}[u, p]$ 
14:  end for
15: end for
16:  $\text{planoValido} \leftarrow \text{true}$ 
17: return sucesso
```

Simulador do sistema hidrelétrico

- Foram implementados dois métodos que geram planos em tempo de execução com base nos volumes dos reservatórios:
 - **Guloso:** consome toda água disponível nos reservatórios, mantendo-os sempre no nível mínimo
 - **Conservador:** consome o mínimo de água possível afim de manter os reservatórios sempre cheios
- Comprovar o funcionamento do simulador como um todo
- Servir como ponto de partida para a busca de planos melhores, no caso de não haver um plano inicial

Gerador de planos

Algoritmo 2 Gerador de planos

```
1: for all  $u \leftarrow \text{selecionaUsinas}(\text{conjuntoUsinas}, \text{qtdUsinas})$  do
2:   for  $p \leftarrow 1$  to 60 do
3:      $\text{sinal} \leftarrow ((\text{rand}()\%100) < 50) ? 1.0 : -1.0$ 
4:      $\text{perc} \leftarrow (\text{rand}() \bmod \text{txVariacao})/100.0$ 
5:      $T[u, p] \leftarrow T[u, p] * (\text{sinal} * \text{perc})$ 
6:
7:      $\text{sinal} \leftarrow ((\text{rand}()\%100) < 50) ? 1.0 : -1.0$ 
8:      $\text{perc} \leftarrow (\text{rand}() \bmod \text{txVariacao})/100.0$ 
9:      $V[u, p] \leftarrow V[u, p] * (\text{sinal} * \text{perc})$ 
10:   end for
11: end for
```

Função fitness

- A princípio, foram implementadas duas funções diferentes:
 - Uma calcula o somatório das diferenças entre E_{SS} e a energia demandada por um subsistema (D_{SS}) em cada período:

$$fitness = \sum_{p=1}^{Periodos} \sum_{s=1}^{Subsistemas} abs(E_{SS}[s, p] - D_{SS}[s, p]) \quad (4)$$

- A outra calcula a diferença entre os somatórios de E_{SS} e D_{SS} em todos os períodos:

$$fitness = \sum_{p=1}^{Periodos} abs\left(\sum_{s=1}^{Subsistemas} E_{SS}[s, p] - \sum_{s=1}^{Subsistemas} D_{SS}(s, p)\right) \quad (5)$$

Função fitness

- Diferença entre as duas equações:
 - A primeira estabelece um tratamento mais igualitário e independente aos subsistemas
 - A segunda os trata como se fossem um único grande sistema
- A consequência disso é que a penalização por um subsistema produzir mais energia elétrica do que o demandado pelo mesmo é maior na equação 4 do na equação 5

Função fitness

- No mundo ideal a equação 4 seria a mais adequada. Mas como isso não é possível, a equação 5 não somente é válida como também produz resultados mais próximos da realidade.
- Por isso foi decidido que o resultado final da função fitness seria uma média ponderada entre os resultados dos dois cálculos.
- Porém, esse cálculo não é aplicável caso algum subsistema não tenha demanda alguma...

$$\begin{aligned}
 \text{fitness} = & (p1 * \sum_{p=1}^{\text{Periodos}} \sum_{s=1}^{\text{Subsistemas}} \text{abs}(E_{ss}[s, p] - D_{ss}[s, p])) + \\
 & (p2 * \sum_{p=1}^{\text{Periodos}} \text{abs}(\sum_{s=1}^{\text{Subsistemas}} E_{ss}[s, p] - \sum_{s=1}^{\text{Subsistemas}} D_{ss}(s, p)))
 \end{aligned}
 \tag{6}$$

Buscador de planos

Algoritmo 3 Buscador de planos

```
1: melhorPlano ← planoOriginal
2: fitnessMelhorPlano ← calculaFitness(planoOriginal)
3: for i ← 0 to maxIteracoes do
4:   novoPlano ← geraPlano(melhorPlano)
5:   simula(novoPlano)
6:   if ehValido(novoPlano) then
7:     fitnessNovoPlano ← calculaFitness(novoPlano)
8:     if fitnessNovoPlano < fitnessMelhorPlano then
9:       melhorPlano ← novoPlano
10:      fitnessMelhorPlano ← fitnessNovoPlano
11:    end if
12:  end if
13: end for
14: return melhorPlano
```

Otimizador

Algoritmo 4 Otimizador de planos

```
1: simula(planoOriginal)
2: if not ehValido(planoOriginal) then
3:   return erro // O plano não é valido
4: end if
5: fitnessMelhorPlano  $\leftarrow$  calculaFitness(planoOriginal)
6: do
7:   novoMelhorPlano  $\leftarrow$  buscadorDePlanos(planoOriginal)
8:   fitnessNovoMelhorPlano  $\leftarrow$  calculaFitness(novoMelhorPlano)
9:   if fitnessNovoMelhorPlano  $\leq$  fitnessMelhorPlano then
10:    decrementoFitness  $\leftarrow$  fitnessMelhorPlano - fitnessNovoMelhorPlano
11:    fitnessMelhorPlano  $\leftarrow$  fitnessNovoMelhorPlano
12:   else
13:    txVariacao  $\leftarrow$  txVariacao * decrementoTxVariacao
14:   end if
15: while txVariacao  $\geq$  txVariacaoMin and decrementoFitness  $\geq$  decrementoFitnessMin
16: return novoMelhorPlano
```

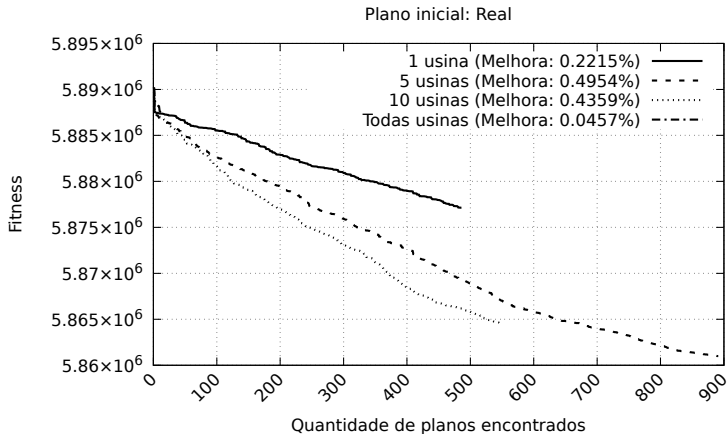


Figura: Evolução do fitness a partir de um plano inicial real.

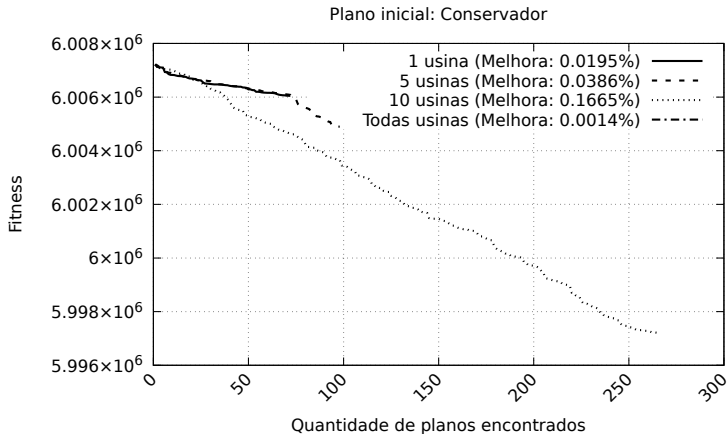


Figura: Evolução do fitness a partir de um plano inicial conservador.

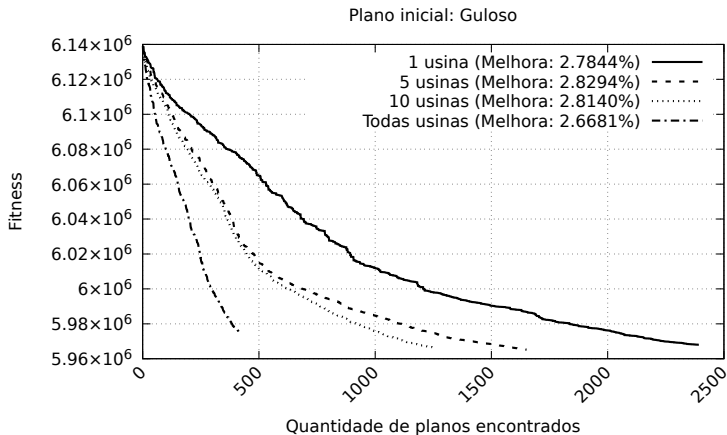


Figura: Evolução do fitness a partir de um plano inicial guloso.

Implementação da rede de transmissão

- Componente essencial para que a simulação seja considerada realista.
- Porém, ainda não foi implementado.
- Caso seja decidido por implementar a rede de transmissão completa:
 - Aumenta consideravelmente o tamanho do grafo e o custo computacional de cada simulação.
 - Exige a criação de um novo plano, o que resulta em novas restrições.
 - Será necessário a implementação de um outro buscador.
- Caso contrário:
 - Necessidade de mudança no cálculo do Fitness
 - Implementar uma verificação da necessidade de mais energia elétrica por cada subsistema

Aprimoramento dos métodos de busca

- O método de Monte Carlo é uma abordagem simplista e minimamente eficiente para o problema proposto.
- Uma ideia seria utilizar métodos de aprendizado computacional para tentar prever o comportamento das ações a medida que os planos são aprimorados, com base nos dados gerados pelas buscas feitas com base no método de Monte Carlo.
- Outra é a criação e utilização de mais de um tipo de busca, fazendo uso dos algoritmos genéticos, por exemplo.
- Armazenar mais de um plano por iteração, mesmo que o plano não seja melhor que o melhor plano corrente, para que o algoritmo possa explorar novas áreas.

Redução do custo computacional

- O alto custo computacional para encontrar soluções é o principal contratempo dos métodos de Monte Carlo.
 - Porém, de acordo com [Doucet et al., 2001], esses métodos são muito paralelizáveis e aplicáveis em diversas situações.
 - No caso desse trabalho, cada simulação é totalmente independente da outra, o que permite que sejam executadas paralelamente.
- Porém, paralelizar a busca em si não é trivial.
 - Uma das questões que precisam ser abordadas é a concorrência entre vários planos e a quantidade de mensagens necessárias para implementar isso.
 - Métodos como o do Quadro-Negro (Blackboard) e Times assíncronos podem ser a solução.

Redução do custo computacional

Atividade	2016				2017	
	09	10	11	12	01	02
Estudar rede de transmissão	X	X				
Implementação da rede de transmissão		X				
Aprimorar os métodos de busca			X			
Reduzir custo computacional			X	X		
Escrita do texto				X	X	
Defesa						X

Tabela: Cronograma de atividades

Springer Science & Business Media.



Hipel, K. W. e McLeod, A. I. (1994).

Time series modelling of water resources and environmental systems, volume 45.

Elsevier.



Russell, S. e Norvig, P. (1995).

A modern approach.

Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 25:27.