

Algoritmo de Otimização por Enxame de Partículas Paralelo para Minimização de Perdas de Potência Ativa em Sistemas Elétricos de Potência

Igor M. Araújo¹, Vincent W. A. Tadaiesky¹, Iago A. Cavalcante¹, Ádamo L. Santana¹

¹Laboratório de Inteligência Computacional e Pesquisa Operacional
Universidade Federal do Pará (UFPA) – Belém – PA – Brasil

{igoraraujo, vincent, adamo}@ufpa.br, iagoangelimc@gmail.com

Abstract. *With the increase in electricity demand and, consequently, the complexity of operation and control of electric power systems in recent decades, studies for reducing active power loss in these systems have followed. Using the IEEE models of 14, 30 and 57 bus, this study aims to evaluate the applicability of GPU computing (Graphic Process Unit) specifically CUDA (Compute Unified Device Architecture), in this area. Performing a critical analysis of bio-inspired strategies used in parallel architecture in order to minimize the loss of active power in electric power systems. It was observed from the results that the use of GPU computing shows promise when the mesh size is increased.*

Resumo. *Com o aumento da procura de eletricidade e, conseqüentemente, a complexidade da operação e controle de sistemas de energia elétrica nas últimas décadas, estudos de redução da perda de potência ativa nestes sistemas têm seguido. Assim, usando os modelos IEEE 14, 30 e 57 barras, este estudo tem como objetivo avaliar a aplicabilidade de computação em GPU (Unidade de Processo Gráfico), mas especificamente de CUDA (Compute Unified Device Architecture), nesta área. Realizando uma análise crítica da utilização de estratégias bioinspirado em arquitetura paralela, a fim de minimizar a perda de potência ativa em sistemas elétricos de potência. Foi observado a partir dos resultados que o uso de computação GPU promissora quando o tamanho da malha é aumentado.*

1. Introdução

A partir do grande avanço tecnológico e crescimento populacional, o consumo de energia elétrica tem crescido consideravelmente. Para que os Sistemas Elétricos de Potência (SEP) suportem essa crescente demanda, unidades de geração são construídas, linhas de transmissão são criadas e redes de distribuição são projetadas. Essa eclosão torna cada vez mais difícil a manutenção de tais sistemas, assim como o controle da qualidade da energia transmitida. Além disso, para suprir esse crescimento do consumo rapidamente, os sistemas atualmente existentes têm executado sob condições estressantes [De and Goswami 2014]. Portanto, com o intuito de aliviar essa condição dos sistemas, busca-se reduzir as perdas de potencia ativa nas linhas de transmissão. Contudo, dentre os tão diversos estudos realizados nos SEP, destaca-se o de fluxo de potência, que possibilita a verificação da situação atual da rede, bem como de dados importantes, como a quantidade de potência ativa perdida na transmissão, por exemplo.

A partir do estudo de fluxo de potência é possível ajustar os parâmetros de funcionamento da rede, a fim de minimizar a perda de potência ativa ou adequar a rede a condições de segurança preestabelecidas. Para isso, realizam-se ajustes em equipamentos de controle, tais como reguladores automáticos de tensão (AVRs - Automatic Voltage Regulator) e compensador estáticos de potência reativa (ShRs – Shunt Reactor). Com base neste objetivo, e de posse das especificações de dependência dos parâmetros dos equipamentos e as condições de funcionamento da rede, tal tarefa pode ser modelada como um problema de otimização, para os quais metaheurísticas como algoritmos genéticos (GA – Genetic Algorithm) e otimização por enxame de partículas (PSO – Particle Swarm Optimization) podem ser aplicadas. Diante disso, o presente trabalho tem como objetivo encontrar a melhor configuração dos equipamentos de controle de um sistema elétrico de potência a fim de reduzir a perda de potência ativa, a partir da utilização do PSO executado na plataforma CUDA.

Este artigo encontra-se organizado como segue: a Seção 2 apresenta alguns estudos relacionados ao tema; na Seção 3 são apresentadas três estratégias de paralelismo em GPGPU; as Seções 4 e 5 apresentam os resultados e conclusões, respectivamente.

2. Trabalhos Correlatos

Existem vários estudos voltados para a otimização de sistemas elétricos de potência. No estudo de [Yoshida et al. 2000] utiliza-se PSO em sistemas elétricos de potência de tal forma a analisar a sensibilidade dos parâmetros do método de enxames de partículas para avaliar a melhor maneira deste minimizar as perdas de potência ativa existentes. Além disso, compara o PSO com RTS (*Reactive Tabu Search*), outro algoritmo de otimização, demonstrando uma maior eficiência para o primeiro.

A partir de [Zimmerman et al. 2011], pode-se entender diferentes métodos computacionais existentes para a solução do problema do fluxo de carga, pois em seu trabalho desenvolve um pacote simulador de sistemas de potência *open-source* baseado em Matlab que apresenta diversos procedimentos para a análise de sistemas elétricos de potência. [Guo et al. 2012] faz um comparativo a partir de diferentes métodos de cálculo de fluxo de potência executados em GPU, fator este determinante na escolha do melhor método apresentado a ser aplicado neste trabalho, o Newton-Raphson.

No trabalho de [Tadaiesky et al. 2013] é demonstrado que nem sempre a paralelização em GPU traz benefícios dependendo do problema e/ou das estratégias de implementação, sua execução em CPU pode ter maior ganho. A partir desse estudo, chegou-se à conclusão de que seria vantajoso explorar diferentes estratégias, não só de apenas executar em GPU, mas também de buscar alternativas de operação em conjunto de CPU e GPU para realizar o objetivo deste trabalho.

3. Estratégias de Paralelização

Para este trabalho, a paralelização foi realizada de três maneiras, sendo que em cada uma tentou-se explorar ao máximo uma característica da arquitetura da plataforma CUDA, quantidade de núcleos, tipo de operações por núcleo e compartilhamento de informação entre as *threads*. Um maior detalhamento dos conceitos e funcionalidades de PSO podem ser encontrados em [Yoshida et al. 2000]. Essas estratégias, denominadas PSOP1, PSOP2 e PSOP3, são apresentadas a seguir.

3.1. PSOP1

Tendo conhecimento que a GPU possui maiores quantidades de núcleos de processamento, cada *thread* tornou-se uma partícula, realizando suas operações de inicialização, movimentação e avaliação independentes das outras. Já a atualização do melhor global não pode ser paralelizada.

3.2. PSOP2

A GPU possui mais núcleos que, comparados à CPU, são simples. Por esse motivo, torna-se mais benéfica sendo utilizada em operações de menor custo operacional. Visto isso, nesta segunda estratégia, a utilização da GPU para operações mais simples como inicialização, movimentação e atualização, já que são execuções de fórmulas e buscas.

Esta estratégia é executada da maneira semelhante ao PSOP1, porém, a etapa de avaliação, por ser mais complexa, contendo o cálculo da matriz de admitância e execução do Newton-Raphson (NR), foi movida para execução na CPU com parte sendo calculada em GPU. Identificou-se que a maior parte do cálculo da avaliação está na operação da matriz Jacobiana, presente no método de NR, devido ser a maior matriz e ser executada iterativamente. Após a movimentação das partículas serem realizadas, os dados das partículas são copiados para o CPU, onde são realizadas avaliações de cada partícula, sequencialmente. Ao término da avaliação, os resultados são copiados para a GPU, para atualização e nova movimentação.

A Arquitetura CUDA nos permite a utilização de até 3 dimensões de *threads* e, para a paralelização dos cálculos da Jacobiana, foi optada pela utilização de duas dimensões. Desta forma, cada *thread* se torna responsável pelo cálculo de um elemento da matriz.

3.3. PSOP3

A fim de evitar que ocorra um gargalo, devido ao fluxo de execução da estratégia anterior ocorrer em GPU depois em CPU e então voltar a GPU, volta-se a executar a avaliação em GPU, porém, desta vez a avaliação de cada partícula não será mais independente, mas sequencial e todas as *threads* disponíveis irão auxiliar nos cálculos; em outras palavras, todas as partículas ajudam a avaliar as demais. Basicamente, esta estratégia é uma união do PSOP1 com o PSOP2.

4. Testes e Resultados

Todos os testes foram realizados em uma mesma máquina com as seguintes configurações: Processador Intel Xeon E5-2620 v2 com 6 núcleos e 15 MB de Cache, 16 GB de Memória RAM, Placa de Vídeo NVIDIA Tesla K20 com 2496 núcleos CUDA e 5 GB de Memória, Sistema Operacional CentOS 6.5 e versão do CUDA Toolkit 6.5 instalada.

4.1. Conjunto de dados

Para os testes, foram utilizados datasets dos modelos do IEEE de 14, 30 e 57 barras, do arquivo de caso de testes em Sistemas de Potência, [Christie 2015]. No modelo de 14 barras fora considerado a presença de ShRs nas barras 9 e 14.

4.2. Parâmetros do PSO

Conforme o resultado dos testes de sensibilidade dos parâmetros do PSO para sistemas elétricos de potência [Yoshida et al. 2000], foi identificada a melhor configuração dos parâmetros do PSO, que são: Social e Cognitivo igual a 2; inércia igual a Equação 1, cujo W_{max} e W_{min} são 0,9 e 0,4, respectivamente e $Iter$ refere-se a iteração atual. Também foram realizados testes variando os parâmetros base do algoritmo de PSO, a saber, considerando um tamanho de população com 10 e 100 partículas, com um critério de parada de no máximo 300 iterações. As dimensões das partículas são os AVR's presentes em todos os geradores e compensadores síncronos com exceção da barra de referência, utilizando valores contínuos entre 0,9 a 1,1 p.u., e também a quantidade de ShR's, utilizando valores discretos entre 0 a 3; onde cada ShR possui um valor de 0,06 p.u.

$$W = W_{max} - \frac{W_{max} - W_{min}}{Iter_{max}} * Iter \quad (1)$$

4.3. Análise das Estratégias

As três estratégias foram executadas utilizando o modelo de 14 Barras. A priori, analisou-se apenas o tempo para a execução até que o critério de parada fosse satisfeito, como pode ser visto na Tabela 1. Os três convergiram para o mesmo ponto porém tiveram um tempo de execução bem diferentes 50s para PSOP1, 2s para o PSOP2 e finalmente 68s para o PSOP3. Percebe-se que onde a avaliação foi toda executada em GPU houve grande consumo de tempo devido a complexidade das operações. O PSOP1, foi prejudicado pelo *delay* de acesso a Memória Global e, já o PSOP3, devido a vários *threads* cooperarem para resolver o mesmo método, foram necessárias muitas sincronizações das *threads*, fato este que influenciou na performance da paralelização.

Tabela 1. Tempo de execução das estratégias para o modelo IEEE de 14 Barras

	PSOP1	PSOP2	PSOP3
Tempo (s)	50	2	68

4.4. Paralelo x Sequencial

Utilizando a estratégia que obteve melhor resultado (PSOP2), foi realizada uma comparação com um PSO não paralelizado, executando de forma sequencial para os modelos de 14, 30 e 57 barras. Realizou-se uma análise mais detalhada, executando 30 vezes para cada teste e coletando a mínima e média da perda de potência ativa, a quantidade de execuções que convergiram para o valor mínimo, a média da iteração de convergência com valor mínimo e a média do tempo necessário para convergir das execuções que obtiveram o valor mínimo. A Tabela 2 exibe os resultados para 10 e 100 partículas.

5. Conclusões

Analisando os resultados, podemos concluir que a execução em GPU apresenta um pequeno *delay* para começar a executar, já que no teste de 14 barras e com 10 partículas a execução em paralelo retardou para ser concluída, porém a iteração do resultado encontrado, foi menor que o sequencial. Esse atraso se dá pela alocação da memória da GPU.

Tabela 2. Resultados para 10 e 100 partículas executado 30 vezes.

	14 Barras		30 Barras		57 Barras	
	Paralelo	Sequencial	Paralelo	Sequencial	Paralelo	Sequencial
10 Partículas						
Média (p.u)	0.132865	0.132744	0.174571	0.175366	0.283650	0.285318
Mínimo (p.u)	0.132284	0.132284	0.174571	0.174571	0.278145	0.278145
Tempo (s)	1.031	0.216	3.156	2.483	17.866	20.587
Iteração	100	101	137	135	129	143
Exec. Conver.	15	17	30	27	10	7
100 Partículas						
Média (p.u)	0.132310	0.132299	0.174571	0.174571	0.278523	0.278397
Mínimo (p.u)	0.132284	0.132284	0.174571	0.174571	0.278145	0.278145
Tempo (s)	3.407	1.356	19.173	18.771	120.514	132.129
Iteração	59	62	98	101	89	91
Exec. Conver.	29	29	30	30	27	28

Contudo, a execução de forma paralelizada mostrou-se promissora para a grande quantidade de barras, compensando o tempo de alocação de memória com a execução dos cálculos em GPU. Outra observação realizada é que o aumento de número de partículas reduz a interação de convergência mas não necessariamente o tempo, já que a maior demora está na avaliação de cada partícula e que com 100 partículas quase todas as 30 execuções convergiram para o valor mínimo.

Agradecimentos

Esta pesquisa fora fomentada em parte pelas agências CAPES, CNPq e FAPESPA. Agradecimentos também à NVIDIA Corporation, pela doação da Tesla K20 GPU, utilizada nesta pesquisa.

Referências

- Christie, R. D. (2015). Power Systems Test Case Archive, University of Washington, Electrical Engineering. online: <http://www.ee.washington.edu/research/pstca/>.
- De, M. and Goswami, S. (2014). Optimal reactive power procurement with voltage stability consideration in deregulated power system. *Power Systems, IEEE Transactions on*, 29(5):2078–2086.
- Guo, C., Jiang, B., Yuan, H., Yang, Z., Wang, L., and Ren, S. (2012). Performance comparisons of parallel power flow solvers on gpu system. In *RTCSA*, pages 232–239. IEEE Computer Society.
- Tadaiesky, V. W. A., Santana, A. L., Dias, L. d. J. C., Oliveira, I. d. I. d., Jacob Junior, A. F. L., and Lobato, F. M. F. (2013). Runtime performance evaluation of gpu and cpu using a genetic algorithm based on neighborhood model. In *The 2013 International Conference on Parallel and Distributed Processing Techniques and Applications*.
- Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., and Nakanishi, Y. (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *Power Systems, IEEE Transactions on*, 15(4):1232–1239.
- Zimmerman, R. D., Murillo-Sánchez, C. E., and Thomas, R. J. (2011). MATPOWER: Steady-State operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.*, 26(1):12–19.