

Implementação Paralela do Método da Otimização Extrema Generalizada

Roberto Luiz Galski, Fernando Manuel Ramos, Fabiano Luís de Sousa, Airam Jônatas Preto e Stephan Stephany

Instituto Nacional de Pesquisas Espaciais (INPE)

galski@ccs.inpe.br, fernando@lac.inpe.br, fabiano@dem.inpe.br, airam@lac.inpe.br, stephan@lac.inpe.br

Abstract

This paper presents a parallel implementation of the Generalized Extremal Optimization - GEO algorithm, and analyses its parallel performance, when applied to a test function commonly used to evaluate optimization algorithms, the Rastringin function. The proposed parallel version, denominated GEOPar-1, is part of an ongoing doctorate research. It was written in Fortran with calls to the Message Passing Interface (MPI) communication library. In order to test the performance of this parallel version, it was included in the test function a simulated processing load, in order to reproduce the execution time of practical applications. The performance results were obtained using a distributed memory parallel machine composed by six dual-processor nodes. Performance results are presented for different number of processors and configurations that include several simulated loads and number of bits to code the test function variables.

Resumo

Este artigo apresenta uma implementação paralela do algoritmo Otimização Extrema Generalizada (*Generalized Extremal Optimization* - GEO), e analisa seu desempenho computacional quando aplicada à uma função teste comumente utilizada para avaliar algoritmos de otimização, a função de Rastringin. A versão paralela implementada, denominada GEOPar-1, é parte de uma pesquisa de doutorado, tendo sido escrita em Fortran com chamadas à biblioteca de comunicação por troca de mensagens MPI (Message Passing Interface). A fim de testar o desempenho dessa versão paralela, foi incluída na função teste uma carga de processamento simulada, de maneira a reproduzir o tempo de execução de aplicações práticas. Os resultados de desempenho foram obtidos utilizando uma máquina paralela de memória distribuída composta de 6 nós biprocessadores. São apresentadas curvas de desempenho em função do número de processadores para várias configurações da função teste, as quais incluem cargas simuladas diversas, bem como diferentes números de bits para a codificação das variáveis no GEO.

1. Introdução

O algoritmo GEO é uma meta-heurística de otimização global (Sousa & Ramos, 2002; Sousa et al., 2003a, 2003b, 2004) desenvolvida recentemente como uma proposta de generalização do algoritmo τ -EO (Boettcher & Percus, 2001). Assim como os métodos Recozimento Simulado - RS (Kirkpatrick et al., 1983) e os Algoritmos Genéticos -

AG (Goldberg, 1989), o GEO é um algoritmo estocástico inspirado em processos observados na natureza.

O GEO foi concebido para resolver problemas de otimização complexos, onde pouco ou nada se conhece a respeito do espaço de busca da função a ser otimizada. Pode ser aplicado a praticamente qualquer problema de otimização contínua ou combinatória, não convexo ou disjunto, com ou sem restrições, podendo, inclusive, ser aplicado a problemas com uma miscelânea de variáveis reais, inteiras ou discretas. É um algoritmo de fácil implementação que não requer o conhecimento de derivadas da função objetivo. Não requer tampouco o fornecimento de uma solução candidata inicial situada no espaço viável. Esta universalidade de aplicação, aliada ao fato de possuir apenas um parâmetro de ajuste (τ), torna o GEO atraente para diversas aplicações.

Em Galski et al. (2003), uma versão aperfeiçoada do GEO foi exposta e testada, obtendo bons resultados. Este artigo apresenta o GEOPar-1, a versão paralela desta última. A função teste utiliza a função de Rastringin, que foi selecionada dentre funções teste não lineares utilizadas em trabalhos anteriores sobre o GEO, tendo, as mesmas, sido utilizadas para avaliação de AG's (Potter & De Jong, 1994) e também para avaliação do GEO original (Sousa & Ramos, 2002).

A motivação principal para o desenvolvimento do GEOPar-1 é a de estender seu uso a aplicações práticas de projeto ótimo, onde o custo computacional de cada avaliação da função objetivo é expressivo, o que torna a otimização indesejável numa máquina monoprocessada face ao alto número de avaliações exigido em métodos estocásticos como o GEO. Dentro deste contexto, o objetivo deste artigo é expor as potencialidades do GEOPar-1, almejando-se antever, através de testes simulados, qual será seu desempenho na execução em paralelo, quando aplicado a problemas reais de otimização da ciência e da engenharia.

2. A Versão Paralela GEOPar-1

No GEO, as variáveis de projeto são codificadas em uma sequência de bits que é similar à um cromossomo binário em um Algoritmo Genético canônico, como exemplificado na Figura 1. A solução candidata inicial do GEO, gerada aleatoriamente, fornece uma sequência de bits, associados às variáveis de projeto X_i . O conjunto destas variáveis é codificado por L bits. Figura 2 apresenta um fluxograma para o GEO. Como pode observar-se, a cada iteração do GEO é alterado um único bit da sequência, gerando-se uma nova solução candidata. Para isso, avalia-se através da função objetivo, a influência da alteração de cada um dos L bits da

seqüência individualmente. Em seguida, ordenam-se os bits de acordo com um índice de adaptação, baseado na avaliação correspondente da função objetivo $F(\mathbf{X})$. Essa ordenação serve para escolher o bit a ser alterado nessa iteração, segundo um critério probabilístico.

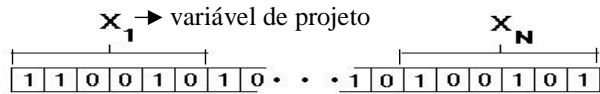


Figura 1 – N variáveis de projeto codificadas em uma seqüência binária com 6 bits por variável ($N = L/6$).

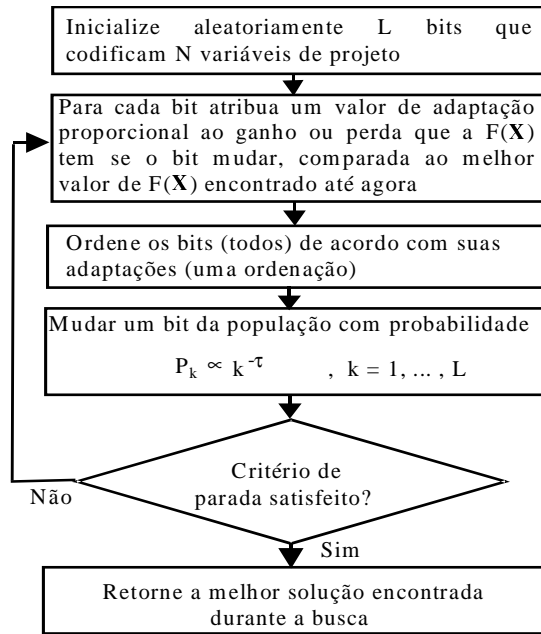


Figura 2 - Fluxograma do GEO
Fonte: Adaptado de Sousa et al., 2003a

Na versão paralela, o programa foi escrito em Fortran e a paralelização foi obtida com chamadas à biblioteca de comunicação por troca de mensagens MPI (Message Passing Interface), para execução numa máquina paralela de memória distribuída, um *cluster*. Explora-se o paralelismo inerente ao algoritmo, dividindo-se as avaliações da seqüência alterada de um único bit entre os processadores seguindo um esquema mestre-escravo. O processador mestre envia aos demais processadores as seqüências a serem avaliadas e, ao receber os resultados, executa a ordenação e a escolha do bit a ser alterado para a próxima iteração. Durante a execução, todos os processadores verificam o critério de parada estabelecido. Neste trabalho, o critério adotado foi o número máximo de avaliações da função objetivo.

Eventualmente, a divisão do domínio entre processadores pode levar a um pequeno desbalanceamento de carga, pois o número de bits L pode não ser múltiplo do número de processadores. Uma consideração importante é que o esquema de paralelização adotado reproduz exatamente o mesmo processo evolutivo do GEO seqüencial, para qualquer número de avaliações de $F(\mathbf{X})$.

Portanto, exceto pelo desempenho temporal, as análises e comparações de desempenho já efetuadas para o GEO valem para o GEOPar-1. Essa foi a razão de se utilizar, neste trabalho, uma função teste baseada apenas na função de Rastrigin, uma vez que objetiva-se a análise do desempenho do algoritmo paralelo. Esta função tem dimensionalidade ajustável, é não linear e multimodal.

Define-se granularidade de um algoritmo paralelo como a razão média entre o tempo de processamento e o de comunicação. Uma paralelização eficiente apresenta alta granularidade, i.e., granularidade grossa, além de apresentar um bom balanceamento de carga entre processadores. No caso de máquinas paralelas de memória distribuída, constituídas por nós monoprocessados ou multiprocessados, a rede de interconexão dos nós tem grande influência no tempo de comunicação. Os parâmetros relevantes são a latência e a largura de banda da rede. A latência da rede (LR) é o tempo gasto para se enviar uma mensagem de tamanho nulo entre dois nós, enquanto que a largura de banda (LB) é a taxa de transmissão da rede. O *cluster* utilizado neste trabalho tem rede padrão Gigabit Ethernet, com taxa nominal de 1 Gbit/s. O tempo gasto para o envio de uma mensagem de tamanho M entre dois nós pode ser calculado como:

$$\text{tempo de envio} = LR + M/LB \quad (1)$$

Assim, tempos de envio elevados podem, conforme o programa em execução, deixar processadores ociosos, à espera de uma comunicação ser completada, ocasionando baixa granularidade, ou seja, perda de desempenho.

3. Função Teste com Tempo de Avaliação Escalonável

Para realizar a avaliação de desempenho proposta, optou-se por modificar o tempo de avaliação da função de Rastrigin, capacitando a mesma a emular cargas de processamento de problemas reais. Foram definidos 3 patamares T_i , $i=\{1,2,3\}$, de tempo de execução, conforme a Tabela 1, com base em exemplos abordados em trabalhos anteriores do GEO (Galski et al., 2004; Sousa et al., 2004; Sousa et al., 2003a). Simula-se, assim, do ponto de vista da carga de processamento, 3 classes de problemas de otimização. Cabe ressaltar que T_1 foi definido mais para testar os limites de GEOPar-1 (pela diminuição da granularidade) do que para simular uma aplicação custosa.

A função objetivo $F(\mathbf{X})$ é dada pela função de Rastrigin, aqui designada FR , a qual é definida por:

$$FR = F(\mathbf{X}) = 3,0N + \sum_{i=1}^N (X_i^2 - 3,0 \cos(2\pi X_i)) \quad (2)$$

$$-5,12 \leq X_i \leq 5,12 \quad N \in \{1,2,3,\dots\}$$

Esta função é escalonável dimensionalmente, por meio de N , o número de variáveis. A solução ótima (mínimo) é $FR=0$ para $\mathbf{X}=\mathbf{0}$. Com relação ao tamanho L da seqüência

de bits, foram definidos 3 níveis de codificação, conforme a Tabela 2. Simula-se, assim, do ponto de vista da complexidade, a qual depende do tamanho da sequência L de bits, 3 níveis deste problema de otimização. A complexidade não está unicamente relacionada a N, mas é diretamente proporcional a este no presente caso. Em todos os níveis, cada variável de projeto utiliza 16 bits (logo, $N = L/16$). Os 9 casos de teste escolhidos estão definidos na Tabela 3, por meio das combinações possíveis {T,L}.

Tabela 1 - Patamares de tempo para a FR

Patamar	Tempo de execução, T (s)
T1	0,001
T2	0,1
T3	1,0

Tabela 2 - Níveis de codificação (L) e variáveis de projeto (N)

Nível	Número de bits da sequência (L)	N
L1	16	1
L2	64	4
L3	640	40

Tabela 3 – Casos de teste adotados (combinações {T,L})

	L1	L2	L3
T1	{0,001; 16}	{0,001; 64}	{0,001; 640}
T2	{0,1; 16}	{0,1; 64}	{0,1; 640}
T3	{1,0; 16}	{1,0; 64}	{1,0; 640}

4. Análise de Desempenho

A fim de testar o desempenho do GEOPar-1, é necessário saber avaliar a influência dos diversos parâmetros do problema. Estes parâmetros foram escolhidos para os diversos casos de teste conforme as tabelas ilustradas na seção anterior.

A cada iteração do GEOPar-1, considerando NP processadores e L múltiplo de NP, cada processador efetua (L/NP) avaliações de $F(X)$ e retorna uma mensagem com o resultado para o nó mestre. Portanto, o tamanho desta mensagem é diretamente proporcional à razão (L/NP) .

Conseqüentemente, L influi, na Equação (1), no termo associado à largura de banda. O tempo T de avaliação de FR, por sua vez, pode, em função do valor de LR, ocasionar ociosidade de processamento, pois se o intervalo entre troca de mensagens for menor do que LR, haverá tempo perdido por todos os nós simplesmente aguardando na fila para enviar sua mensagem. Além disso, como a razão L/NP está associada ao número de avaliações de FR realizadas por cada processador, L também pode ocasionar ociosidade de processamento devido à latência LR.

As combinações {T,L} ilustradas na Tabela 3 foram testadas com o GEOPar-1, sendo os tempos de execução apresentados nas Figuras 3 a 11, a seguir, para NP variando entre 1 e 12. Para os mesmos casos, a versão sequencial original do GEO foi executada num nó do *cluster*, para servir de referência no cálculo da eficiência ϵ . O número de avaliações limite utilizado foi 2000, servindo de critério de parada. Para cada caso, três execuções

independentes foram feitas utilizando sementes diferentes na geração dos números aleatórios. Em seguida, os tempos médios para cada semente foram calculados e utilizados para gerar as curvas em escala logarítmica (Figuras 3 a 11). O valor limite de 2000 avaliações foi escolhido de maneira a garantir um mínimo de 3 iterações do algoritmo, conforme exposto na Seção 2, quando utilizado o nível de codificação L3, em que cada iteração requer 640 avaliações. Quando os níveis L2 e L1 são utilizados, o número mínimo de iterações completas sobe para 30 e 120, respectivamente.

O critério de parada adotado, que impõe um limite no número de avaliações, é adequado, pois visa manter um custo computacional equivalente para os diferentes níveis de codificação em cada patamar T_i .

O *speed-up* $S(NP)$ de uma execução paralela é dado pela razão entre o tempo de execução da versão sequencial e o tempo de execução da versão paralela com NP processadores:

$$S(NP) = t_{\text{sequencial}}/t_{\text{paralelo}} \quad (3)$$

A eficiência de uma execução paralela é dada por:

$$\epsilon(NP) = S(NP)/NP \quad (4)$$

Observando-se as Figuras 3 a 11, pode-se notar que: (i) na classe de problemas T1 (Figs. 3 a 5) há queda de eficiência com o aumento de NP, sendo esta queda mais significativa para valores menores de L; (ii) nas classes T2 (Figs. 6 a 8) e T3 (Figs. 9 a 11) há queda de eficiência com o aumento de NP para L1 e, em menor proporção, também para L2. As Figuras 12, 13 e 14, apresentam as eficiências obtidas pelo GEOPar-1 para T1, T2 e T3, respectivamente.

A análise dessas figuras mostra que, para T2 e T3, as eficiências são similares, apesar do tempo de processamento maior de T3. Isto significa que, nestes casos, a granularidade é satisfatória, i.e., o tempo de processamento é grande o suficiente em relação ao tempo de comunicação, resultando em boa eficiência, a menos do problema de desbalanceamento de carga, que tende a ser maior para valores menores de L. Observa-se nitidamente os picos de eficiência nas Figuras 13 e 14 quando o número de processadores é múltiplo de L (razão L/NP inteira) e portanto há balanceamento de carga. No caso T1, ilustrado na Figura 12, o tempo de avaliação é muito pequeno, implicando numa granularidade mais fina que penaliza a eficiência para maior número de processadores, uma vez que isso implica obviamente em mais comunicação. Em função disso, o efeito de desbalanceamento de carga observado em T2 e T3 fica visivelmente atenuado.

Quando a razão L/NP é não inteira, a parte inteira e o resto são dados por:

$$D = \text{INT}(L/NP) \text{ e } R = \text{MOD}(L/NP) \quad (5)$$

Isso implica que R processadores executam $(D + 1)$ avaliações e os demais, D avaliações. Uma vez que o tempo de avaliação é uniforme, o desbalanceamento de

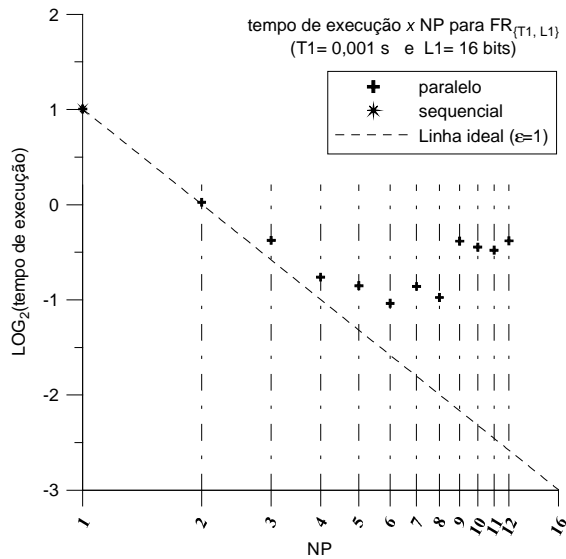


Figura 3 - Tempos de execução para {T1; L1}

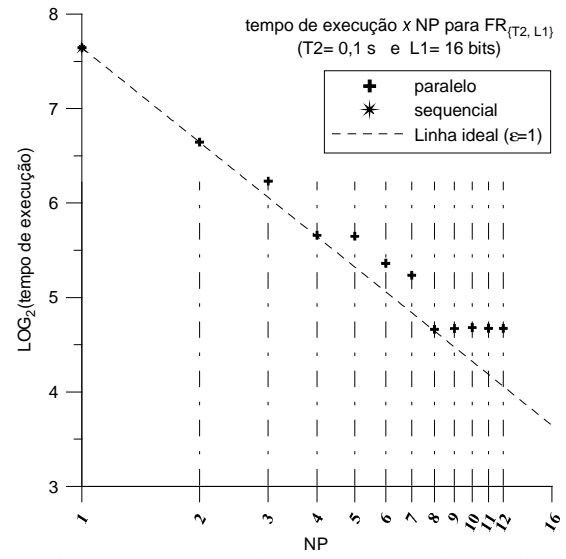


Figura 6 - Tempos de execução para {T2; L1}

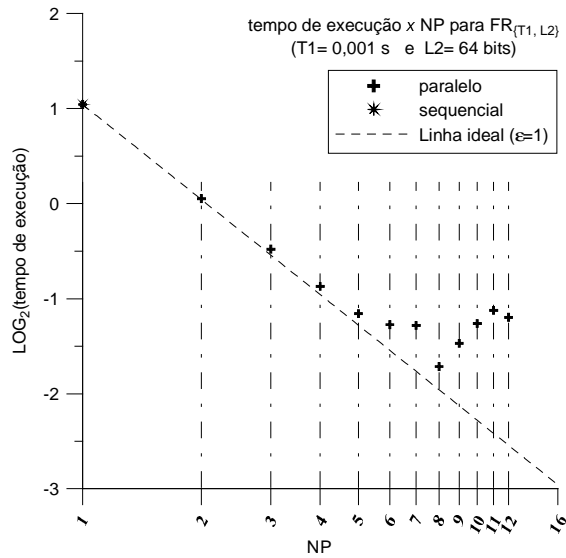


Figura 4 - Tempos de execução para {T1; L2}

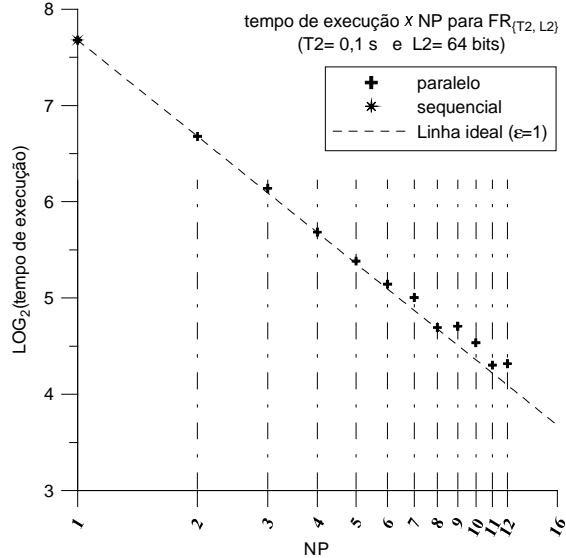


Figura 7 - Tempos de execução para {T2; L2}

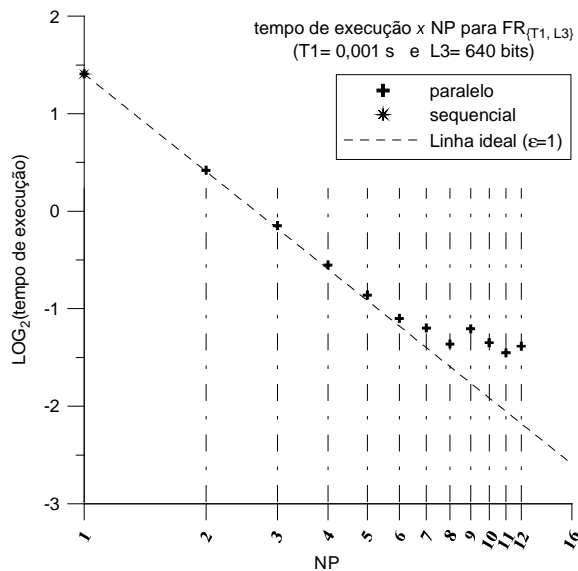


Figura 5 - Tempos de execução para {T1; L3}

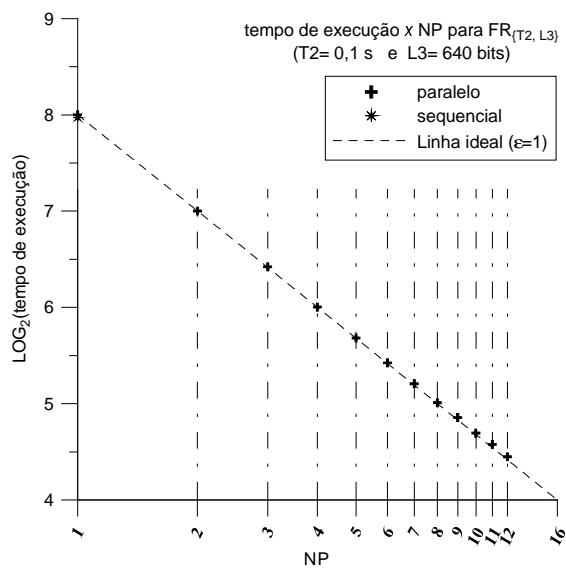


Figura 8 - Tempos de execução para {T2; L3}

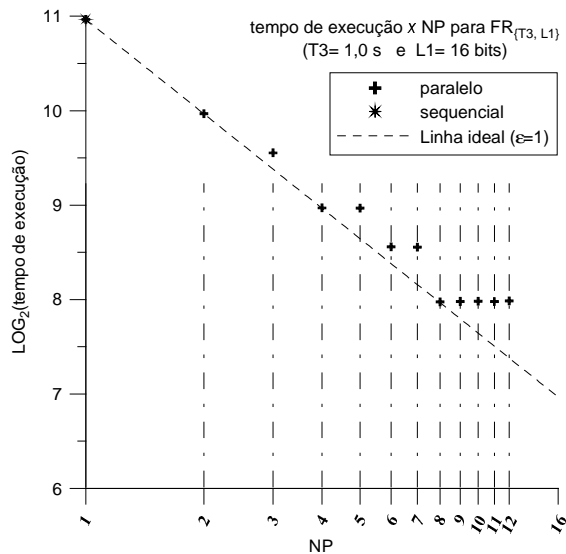


Figura 9 - Tempos de execução para {T3; L1}

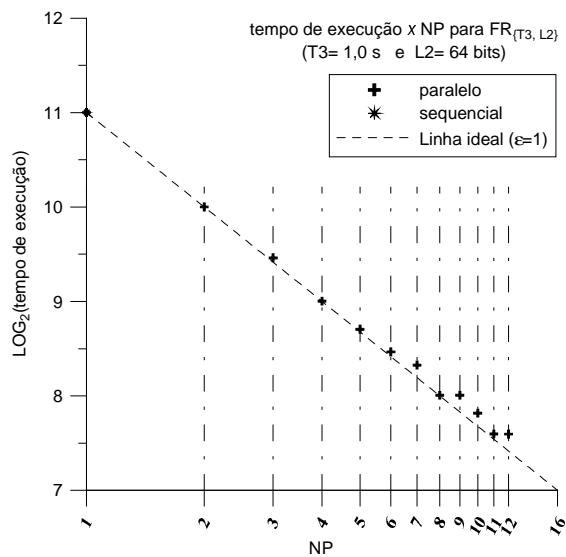


Figura 10 - Tempos de execução para {T3; L2}

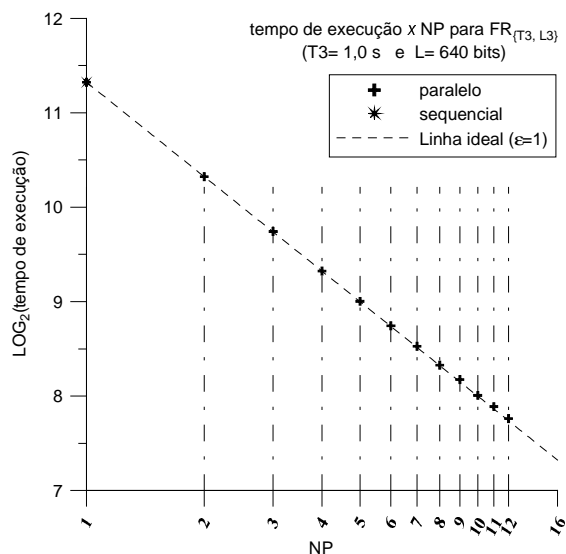


Figura 11 - Tempos de execução para {T3; L3}

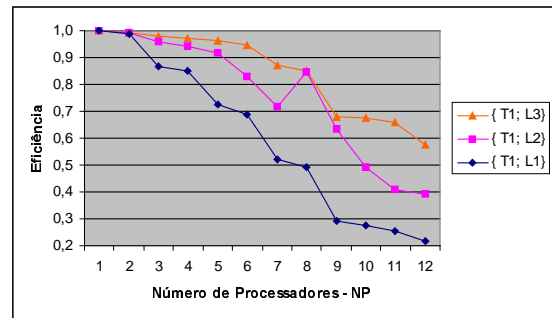


Figura 12 - Eficiência com T1

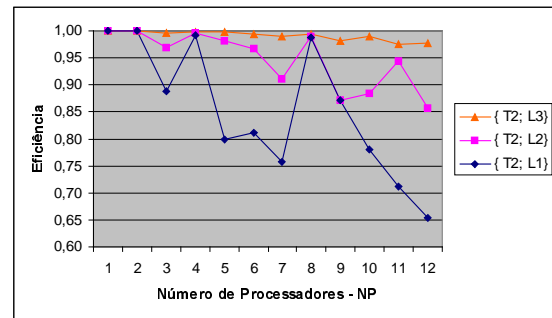


Figura 13 - Eficiência com T2

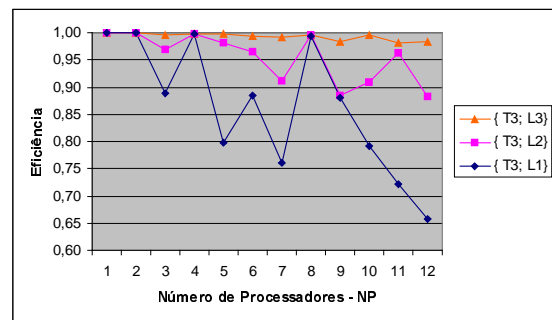


Figura 14 - Eficiência com T3

carga pode ser quantificado: há $(NP-R)$ processadores ociosos durante 1 avaliação. Uma eficiência ótima pressupõe balanceamento de carga entre processadores. Pode-se demonstrar que, desprezando custos de comunicação, a eficiência máxima teórica considerando o desbalanceamento acima exposto é dada por:

$$\epsilon_{\max} = L/(NP.(D+1)) \quad (R \neq 0) \quad (6)$$

A Tabela 4, a seguir, compara os valores máximos teóricos de eficiência, obtidos da Equação (6), aos valores reais observados nos testes. Se, por um lado, a eficiência real tende a ser inferior à máxima teórica acima devido à comunicação, por outro lado, há otimização do acesso à memória cache, uma vez que, com o aumento de NP, a quantidade de memória exigida por processador tende a ser menor do que no algoritmo sequencial. Isto diminui o tempo de processamento por processador, podendo-se obter valores de eficiência maiores que o da Equação (6), conforme observado nos valores destacados na Tabela 4.

Tabela 4 – Valor teórico máximo e observado para ϵ

L	NP	ϵ (máx.)	ϵ (ocorrido) com {T1; T2; T3}
L1 (16 bits)	3	0,8889	{0,8674; 0,8875; 0,8881}
	5	0,8000	{0,7239; 0,7979; 0,7987}
	6	0,8889	{0,6857; 0,8110; 0,8846}
	7	0,7619	{0,5199; 0,7585; 0,7599}
	9	0,8889	{0,2906; 0,8722; 0,8807}
	10	0,8000	{0,2733; 0,7800; 0,7917}
	11	0,7273	{0,2541; 0,7126; 0,7209}
	12	0,6667	{0,2174; 0,6538; 0,6581}
L2 (64 bits)	3	0,9697	{0,9572; 0,9686; 0,9688}
	5	0,9846	{0,9183; 0,9819; 0,9810}
	6	0,9697	{0,8293; 0,9659; 0,9656}
	7	0,9143	{0,7149; 0,9114; 0,9118}
	9	0,8889	{0,6340; 0,8715; 0,8848}
	10	0,9143	{0,4936; 0,8835; 0,9084}
	11	0,9697	{0,4077; 0,9434; 0,9619}
	12	0,8889	{0,3934; 0,8561; 0,8825}
L3 (640 bits)	3	0,9969	{0,9801; 0,9959; 0,9961}
	5	0,9922	{0,9630; 0,9979 ; 0,9983 }
	6	0,9969	{0,9475; 0,9943; 0,9948}
	7	0,9938	{0,8693; 0,9907; 0,9913}
	9	0,9877	{0,6797; 0,9823; 0,9845}
	10	0,9846	{0,6751; 0,9897 ; 0,9959 }
	11	0,9861	{0,6596; 0,9756; 0,9820}
	12	0,9877	{0,5767; 0,9775; 0,9829}

Como esperado, as combinações {T1,x} e {x,L1}, demarcaram o limite inferior de desempenho do GEOPar-1. Ressalta-se, mais uma vez, que essas combinações não são representativas de problemas reais complexos e custosos. Analisando-se as combinações que são representativas ({T2,L2}, {T2,L3}, {T3,L2}, {T3,L3}), tem-se $\epsilon > 0,85$ para qualquer NP, sendo que com balanceamento de carga, tem-se $\epsilon > 0,95$.

5. Conclusões

Neste artigo, a versão paralelizada GEOPar-1 do algoritmo estocástico de otimização global GEO foi apresentada e testada num *cluster* composto por 6 nós biprocessadores interligados por uma rede Gigabit Ethernet. Foi utilizada uma função teste com dimensionalidade e tempo de avaliação configuráveis, de forma a simular cargas de processamento típicas de aplicações complexas. Os testes de desempenho efetuados comprovaram que o GEOPar-1 tem um grande potencial como otimizador paralelo, obtendo eficiência acima de 0,85 em todos os testes que procuraram simular aplicações complexas, computacionalmente custosas. Este fato, somado à universalidade de aplicação do GEO, característica mantida inalterada no GEOPar-1, aumenta ainda mais a aplicabilidade do algoritmo em problemas reais de otimização, principalmente naqueles em que o custo de processamento da avaliação da função objetivo seja elevado.

6. Referências

- Boettcher, S.; Percus, A. "Optimization With Extremal Dynamics". *Physical Review Letters*, v. 86, pp. 5211-5214, 2001.
- Cardoso, H.P.; Muraoka, I.; Bastos, J.L.F.; Bambace, L.A.W.; Oliveira Filho, O.B.; Leite, R.M.G. *Software de Análise Térmica PCTER: Manual do Usuário*, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil, 1990.
- Galski, R. L.; Sousa, F. L.; Ramos, F. M.; Muraoka, I. "Spacecraft Thermal Design with the Generalized Extremal Optimization Algorithm". Anais do Inverse Problems, Design and Optimization Symposium (cd-rom), Rio de Janeiro, Março, 2004.
- Galski, R. L.; Sousa, F. L.; Ramos, F. M.; Muraoka, I. "Um Estudo para o Aprimoramento do Método da Otimização Extrema Generalizada". Terceiro Workshop dos cursos de Computação Aplicada do INPE (III WORCAP), São José dos Campos, Novembro, 2003.
- Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing, New York, 1989.
- Kirkpatrick, S.; Gellat Jr., C.D.; Vecchi, M.P. "Optimizing by simulated annealing". *Science*, v. 220, n. 4598, pp. 671-680, 1983.
- Potter, M.; De Jong, K.A., "A Cooperative Coevolutionary Approach to Function Optimization", Anais do *Third Parallel Problem Solving From Nature*, Springer-Verlag, pp. 249-257, 1994.
- Sousa, F. L.; Ramos, F. M., "Function Optimization Using Extremal Dynamics", Anais do *4th International Conference on Inverse Problems in Engineering*, Rio de Janeiro, Brasil, 2002.
- Sousa, F.L., Ramos, F.M., P. Paglione, R.M. Girardi, "New Stochastic Algorithm for Design Optimization", *AIAA Journal*, Vol. 41, Number 9, pp. 1808-1818, 2003.
- Sousa, F.L. Vlassov, V. e Ramos, F.M. "Generalized Extremal Optimization for Solving Complex Optimal Design Problems". *Lecture Notes on Computer Science*, 2723:375-376, 2003.
- Sousa, F. L.; Ramos, F. M.; Galski, R. L.; Muraoka, I. "Generalized Extremal Optimization: A New Meta-heuristic Inspired by a Model of Natural Evolution". *Recent Developments in Biologically Inspired Computing*, De Castro, L. N. & Von Zuben, F. J. (editores), Idea Group Inc., pp. 41-60, 2004.