Paralelização do Algoritmo de Otimização por Enxame de Partículas para Combinação de Descritores de Imagem

Handrey E. Galon[1], **Roberto S. U. Rosso Jr.**[1], **Rafael S. Parpinelli**[1]

[1]Programa de Pós-Graduação em Computação Aplicada – PPGCA
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

handrey.galon@edu.udesc.br, roberto.rosso@udesc.br, rafael.parpinelli@udesc.br

Resumo. *A Otimização por Enxame de Partículas é um algoritmo de otimização para problemas complexos de domínio contínuo. Apresenta-se neste trabalho a versão deste algoritmo implementado de forma paralela utilizando o pacote multiprocessing do Python para realizar a combinação de descritores de imagem que, quando comparado com sua versão sequencial, apresenta um speedup de médio de 1,5x.*

A Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*) proposta por [Kennedy and Eberhart 1995] é um algoritmo de otimização para problemas complexos de domínio contínuo. Este algoritmo é um dos mais utilizados nos que se encaixam na categoria de *Swarm Intelligence*, porém, para problemas de grande porte, o elevado número de avaliações da função *fitness* restringe a sua aplicação. Por outro lado, a computação paralela é uma alternativa atraente para sua utilização, devido ao fato de ser facilmente paralelizado. Neste trabalho, são desenvolvidas uma versão serial e outra paralela de diferentes abordagens do PSO. O algoritmo paralelo é implementado utilizando o pacote *multiprocessing* do Python, que suporta a criação de processos usando uma API semelhante ao módulo de *threading*. O pacote oferece simultaneidade local e remota usando subprocessos em vez de *threads*. Devido a isso, o módulo de *multiprocessing* permite que o programador aproveite completamente os vários núcleos de uma determinada máquina. Dessa forma, este trabalho apresenta os resultados obtidos através da implementação das seguintes abordagens do algoritmo PSO: (1) PSO em sua versão canônica; (2) PSOw, proposto por [Shi and Eberhart 1998], que é uma versão do PSO com a adição do parâmetro de peso de inércia. Nesta abordagem foram utilizadas uma versão com peso de inércia fixo durante as iterações (PSOw *Offline*), e com peso de inércia adaptativo (PSOw *Online*); e, por último (3) PSOx, proposto por [Clerc and Kennedy 2002], que possui o parâmetro referente a um fator de constrição do enxame de partículas.

O objetivo do trabalho é realizar a combinação de descritores de imagem, atribuindo pesos para cada um, visando melhorar a taxa de classificação das imagens da base ETH-80, que conta com 3.280 imagens distribuídas uniformemente entre oito classes. A Equação 1 apresenta como será feita a combinação dos descritores.

$$\delta_D(I) = \sum_{1}^{n} w_i \cdot d_i(I) \quad 0 \leq w_i \leq 1 \tag{1}$$

Onde δ_D resulta na medida de similaridade entre as imagens da base, w_i é o peso atribuído à cada descritor e $d_i(I)$ são os próprios descritores. Quanto menor o valor de

w_i, menor será a influência do descritor na classificação das imagens, e quanto maior o valor de w_i, maior será sua influência. Os descritores de imagem utilizados neste trabalho são: ACC (*Auto Color Correlation*), BIC (*border/interior pixel classification*), JAC (*Joint auto-correlogram*), SID (*Invariant Steerable Pyramid Decomposition*), HTD (*Homogeneous Texture Descriptor*), SMS (*Steerable Mean Standard-deviation*), Fourier, MSF (*MultiScale Fractal*) e TSDIZ (*Tensor Scale Descriptor with Influence Zones*), os quais possuem uma descrição mais detalhada em [da Silva 2011].

O resultado de cada descritor de imagem é uma matriz com dimensões de 3.280×3.280 que representa o vetor de características de cada imagem da base. A paralelização ocorre na etapa de realizar o cálculo da matriz resultante de um descritor multiplicando pelo respectivo peso atribuído pelo PSO. Esses cálculos são utilizados para obter o valor do *fitness* das partículas, que é a própria taxa de acerto de classificação das imagens.

A execução dos algoritmos foi realizada 12 vezes para cada abordagem do PSO em uma máquina com processador Intel Core i7-4771, 3,50 GHz com 16 GB de memória RAM e com o sistema operacional Linux Ubuntu 17.10 (64 bits). A taxa de acerto na classificação das imagens da base, após a combinação dos descritores, utilizando as abordagens citadas do PSO foram de 92% em média, apresentando uma melhora quando comparado ao método de combinação de descritores utilizando a técnica de voto majoritário, que apresentou uma taxa de acerto de 89% em média. A Tabela 1 apresenta os tempos médios de processamento obtidos no processo de otimização. Verifica-se um speedup não muito expressivo dadas as características do problema e tamanho da base de dados.

Tabela 1. Resultados das abordagens testadas.

Abordagem	**Sequencial**	**Paralelo**	***Speedup***
PSO Canônico	15h49min (±0,11)	10h22min (±0,08)	1,52x
PSOw *Offline*	16h07min (±0,24)	10h38min (±0,09)	1,51x
PSOw *Online*	16h09min (±0,14)	10h43min (±0,08)	1,50x
PSOx	16h32min (±0,33)	10h42min (±0,10)	1,45x

Trabalhos futuros envolvem testar outros métodos de paralelização, como por exemplo, MPI4Py (*Message Passing Interface for Python*), visando melhorar o *speedup*.

Referências

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

da Silva, A. T. (2011). *Recuperação de imagens por conteúdo baseada em realimentação de relevância e classificador por floresta de caminhos ótimos*. PhD thesis, Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948.

Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation Proceedings*, pages 69–73.