

PARALELIZAÇÃO DA BIBLIOTECA PYROSETTA PARA PREDIÇÃO DE ESTRUTURA DE PROTEÍNA COM EVOLUÇÃO DIFERENCIAL

Gabriel Dominico¹, Pedro Henrique Narloch², Rafael Stubs Parpinelli³

¹ Acadêmico(a) do Curso de BCC - CCT - bolsista PROBIC/UDESC

² Acadêmico do Curso de PPGCA – CCT

³ Orientador, Departamento de Ciência da Computação - CCT – rafael.parpinelli@udesc.br

Palavras-chave: Algoritmos Bio-inspirados. Proteínas. Funções de Energia.

Algoritmos bio-inspirados vem sendo cada mais utilizados para a resolução de problemas complexos onde abordagens tradicionais (seja por falta de poder de processamento ou a alta demanda de tempo) não conseguem resolver, sendo necessário a utilização das técnicas inspiradas na biologia. Um destes problemas é a predição de estruturas de proteínas, onde a alta complexidade se relaciona com a grande quantidade de sequências que um aminoácido pode assumir. Uma proteína pode ser definida como uma macromolécula que desempenha diferentes tipos de funções para a sobrevivência do organismo em que se encontra presente.

O objetivo do trabalho realizado foi a diminuição do tempo de processamento destas funções de energia. A possibilidade analisada para a redução deste tempo foi realizar a implementação de forma paralela para o cálculo das funções de energia.

Para a realização dos experimentos, foi utilizado a abordagem da Evolução Diferencial, o qual se encontra presente dentro da classe de algoritmos bio-inspirados. Para a parte do cálculo da diversidade foi utilizado a fórmula abaixo:

$$\frac{\sum_{i=0}^N \ln \left(1 + \forall j \in [i+1, N] \frac{1}{D} \sqrt{\left(\sum_{k=1}^D (x_{i,k} - x_{j,k})^2 \right)} \right)}{NMDF}$$

onde, D representa a quantidade de dimensões, N o tamanho da população e x a possível solução do problema. A variável NMDF fica responsável pela normalização da diversidade, assumindo o maior valor encontrado. Esta fica entre os valores 0 (menor) e 1 (maior).

Na Fig 1. apresenta-se um simples fluxo de como a Evolução Diferencial acontece. O quadrado em vermelho representa a parte em que foi aplicado o paralelismo, ou seja, cada indivíduo é analisado de forma paralela para obter o seu valor de acordo com a função de energia.

No trabalho foi verificado o impacto da diversidade nos resultados encontrados e como conciliar as técnicas de intensificação e diversificação no algoritmo.

Para a função ROSETTA foi utilizado a linguagem Python, o qual tem uma biblioteca chamada de PyRosetta que já implementa essa função de energia. A paralelização foi feita utilizando da biblioteca Joblib, porém não possuiu um impacto positivo no tempo de

processamento (visto na Tab. 1). Uma dificuldade ao paralelizar a função de avaliação de energia foi a própria biblioteca do PyRosetta, onde apresentava conflitos com os processos criados pela



Fig. 1 Fluxo do algoritmo implementado

biblioteca.

Para a realização dos testes foi utilizado a proteína 1CRN, a qual possui 41 aminoácidos. Foi utilizado $F=0.5$ e $CR=1$ com um total de 50 iterações sendo executado 5 vezes. Foi utilizado 50 indivíduos na população e cada indivíduo tendo 46 ângulos para a otimização. Todos os testes foram realizados em um hardware CPU @ i7-4770k com 6GB de RAM e sistema operacional Ubuntu 14.04.

Como pode ser analisado, a Tab. 1 traz os resultados referentes a função de energia Rosetta. A principal diferença se encontra em relação ao tempo de processamento, onde quando mais *threads*, maior o tempo encontrado. Este aumento no tempo é atribuído a demora para a criação/destruição destas *threads*, causando com que não seja essa a melhor abordagem para realizar a paralelização. Verifica-se também que a média e desvio padrão tem uma diferença, o que pode ser atribuído ao algoritmo implementado, o qual é populacional e probabilístico, não tendo possuindo um resultado “estático”.

Número de threads	Média \pm Desvio Padrão	Diversidade	Tempo (s)
1	830.171342 \pm 14.4090	0.085855	25.3332
8	1307.135596 \pm 15.2076	0.105838	34.4861
16	943.020594 \pm 14.9050	0.108636	79.6638

Tab. 1 Tabela de resultados para a função Rosetta.

Um possível trabalho futuro para melhorar o desempenho da função Rosetta seria a implementação da paralelização utilizando alguma outra biblioteca e verificar o impacto entre diferentes bibliotecas ou, em um caso mais extremo, realizar a implementação em outra linguagem com maior desempenho.