

Y	T	A	N	F	P	I	I	K	Y	W	G	Q	Z
V	V	Y	I	P	C	M	R	Q	I	I	E	S	R
K	W	X	J	J	F	A	G	M	S	S	U	T	I
T	W	F	U	G	A	S	E	S	N	I	V	J	B
E	Á	F	R	I	C	A	J	S	M	H	S	E	C
S	M	E	S	F	N	H	P	F	Z	U	M	B	I
C	H	D	O	B	K	Z	D	D	H	C	Q	Z	C
R	I	W	D	H	C	A	S	T	I	G	O	S	J
A	G	T	Q	I	S	A	B	E	L	S	X	O	M
V	L	I	B	E	R	D	A	D	E	R	O	D	D
O	P	A	L	M	A	R	E	S	H	Z	H	K	N
S	L	S	E	N	Z	A	L	A	S	V	W	J	T
A	E	Q	U	I	L	O	M	B	O	S	K	I	H
S	I	K	P	N	V	V	J	U	S	T	I	Ç	A

ÁFRICA

CASTIGOS

ESCRAVOS

FUGAS

ISABEL

JUSTIÇA

LEI

LIBERDADE

PALMARES

QUILOMBOS

SENZALAS

ZUMBI

noapalavras.com



Gerador de Caça-palavras

Lucas Mateus Fernandes

Palavras-chave: Genético, Caça-palavras, Permutação

Introdução

O problema do caça-palavras consiste em colocar o máximo de palavras em uma matriz ‘N’x’M’ de modo que as palavras não estejam sobrecrevendo outras palavras.

Quanto mais palavras existirem mais complexo se torna o processo.

O trabalho aqui expresso usa como técnica para resolução algoritmo genético que usa técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação.

Metodologia

O problema combinatório foi reduzido a um problema de permutação onde a ordem das palavras é importante para a resolução do problema.

Sempre que um individuo é criado é reposicionada todas as palavras e é calculado o fitness ou seja o quão adaptado a solução é.

Formula do Fitness:

$$v_{Sentido} \times \left(\left(\sum_{i=1}^n (tPi \times 0.1) \right) + n \right)$$

vSentido = Variabilidade do Sentido
n= total de palavras inseridas
tPi = Tamanho da Palavra atual

Formula Variabilidade do Sentido:

$$var\ 1 = (tD - tH)$$
$$var\ 2 = (tD - tH)$$
$$var\ 3 = (tH - tV)$$

$$mVar > 0.25 \Rightarrow v_{Sentido} = 1 - mVar$$
$$mVar \leq 0.25 \Rightarrow v_{Sentido} = 1$$

tD = Total de Palavras na diagonal
tH = Total de Palavras na horizontal
tV = Total de palavras na vertical
mVar = max(var1,var2,var3) dividido pelo total de palavras

O critério de parada do algoritmo é a evolução de ‘n’ gerações.

Ao exceder a uma idade ‘x’ os indivíduos são descartados exceto o melhor individuo que persiste entre as gerações.

Para selecionar os pais foi utilizado o método da roleta a modo de evitar uma elitização precoce dos indivíduos.

Operador de cross over utilizado foi o ‘Order Crossover (OX)’ no qual evita que uma operação provoque a repetição de um gene no mesmo individuo e mantém uma ordem relativada dos genes.

Resultados e Discussões

Devido a forma como as palavras são inseridas na matriz, a palavra ocupa um retângulo, tal representação limitou a criação dos caça-palavras porem reduziu o custo para verificar a colisão entre as palavras.

Quanto mais palavras for entregue como parâmetro de entrada mais demorado será a execução do algoritmos pois influencia diretamente na quantidade de genes em cada indivíduo e consequentemente reflete no calculo de colisão entre os genes.

A cada criação de um individuo existe um padrão para a inserção das palavras ou seja, a próxima palavra só pode ser inserida no inicio ou no final da diagonal secundaria que corta o retângulo que representa a palavra, isto gera um padrão para resolução dos caça-palavras.

Conclusões

Mesmo que a solução gerada seja sub-ótima o custo para criação acaba se tornando um pouco alto devido os cálculos feitos para factibilidade do indivíduo.

As respostas geradas seguem um padrão fácil de se detectar durante a resolução do caça palavra.

Foi observado que a quantidade de indivíduos inicial item um valor alto no processamento da solução, e talvez um critério diferente para permanência do individuo na população reduziria o custo total porem poderia chegar a um ótima local muito rápido.

O critério de parada do algoritmo pode ser melhorado de modo que após ‘m’ gerações sem melhora no fitness o algoritmo pode encerrar, porem um ‘m’ muito pequeno poderia gerar um resposta sub-ótima muito rápido porem muito limitada.

Referências Bibliográficas

DE ANDRADE, Mariana Silva Faleiro et al. Um algoritmo evolutivo híbrido aplicado a solução do problema de corte bidimensional guilhotinado. XLI SBPO-Simpósio Brasileiro de Pesquisa Operacional, 2009.

POZO, Aurora et al. Computação evolutiva. Universidade Federal do Paraná, 61p.(Grupo de Pesquisas em Computação Evolutiva, Departamento de Informática-Universidade Federal do Paraná), 2005.

CDSID. META-HEURÍSTICA SIMULATED ANNEALING APLICADA AO PROBLEMA DE CORTE BIDIMENSIONAL NÃO-GUILHOTINADO. Disponível em: <http://cdsid.org.br/sbpo2015/wp-content/uploads/2015/08/143004.pdf>. Acesso em: 1 jun. 2019.

