Technical Report
Genetic Programming for V-Measure Maximization

Alysson Ribeiro da Silva
Computer Science Graduate Program
Federal University of Minas Gerais
Belo Horizonte, Brazil

Abstract—A programação genética pode ser utilizada para gerar conteúdo, ou uma função, que pode vir a ser utilizada com diversas finalidades. Esta proposta contempla a implementação Programação Genética, algoritmo de clusterização k-means, e o algoritmo para cálculo da média entre entropias *V-Measure*. Foram realizadas análises de parâmetro, com relação a métodos de inicialização da população, tamanho da população, taxa de mutação e cruzamento, onde resultados mostram a capacidade da proposta de convergir, manter ou extinguir a diversidade de acordo com alterações propostas. Além disso, também foram avaliadas duas bases de dados, uma referente a detecção de câncer de mama e outra referente a classificação de vidros. Os resultados indicam a viabilidade da técnica para aproximação de funções que descrevem tendências.

Index Terms—Genetic Programming, Evolutionary Computing,

INTRODUÇÃO

A programação genética [1] pode ser utilizada para se gerar conteúdo, ou uma função, que pode vir a ser utilizada com diversas finalidades. Neste trabalho, a programação genética é utilizada para se encontrar uma métrica de distância capaz de maximizar a medida *V-Measure* que é uma média entre entropias para classificação em diferentes contextos.

Esta proposta contempla a implementação dos seguintes algoritmos para solucionar problemas no contexto da Programação Genética: Programação Genética e afins, algoritmo de clusterização k-means, e o algoritmo para cálculo da média entre entropias (*V-Measure*) [2]. As mesmas foram codificadas utilizando a linguagem *C++*. Essa escolha foi feita pois inicialmente uma implementação preliminar utilizando a linguagem *Python* se mostrou ineficiente. Não foram utilizadas bibliotecas adicionais para a implementação dos algoritmos mencionados.

Os testes foram realizados utilizando duas bases de dados, uma referente a detecção de câncer de mama e outra referente a classificação de tipos de vidros. A qualidade da solução foi avaliada em termos da média e desvio padrão do melhor e pior indivíduos ao longo das gerações, média e desvio padrão do melhor e pior indivíduo. Além disso, também foram analisadas a influência dos parâmetros durante a execução do algoritmo proposto. Os resultados obtidos não foram satisfatórios para alguns casos. Já em analises particulares o algoritmo mostra-se capaz de convergir a uma solução sub-ótima. Análises e discussões são apresentadas ao fim do documento.

1 IMPLEMENTAÇÃO

Nesta seção são apresentadas as implementações propostas para o algoritmo de programação genética baseado em árvores. Primeiramente é apresentado o método de seleção. Em seguida, são apresentados os métodos de cruzamento e mutação, respectivamente. E por fim, é apresentado o algoritmo de clusterização k-means e a métrica de média entre entropias implementados.

1.1 Representação de um indivíduo

Os indivíduos são representado por árvores binárias, onde a raiz de cada sub-árvore é sempre uma função e terminais podem ser variáveis ou constantes. Essa representação foi escolhida por sua facilidade de implementação e entendimento. Além disso, também foi considerada a conveniência de efetuar a troca de sub-árvores com a utilização de apontadores

O nodo de cada indivíduo, além de armazenar referências para seu lado esquerdo e direito, também contém variáveis auxiliares que permitem facilitar as operações de cruzamento e mutação. Dentre as variáveis utilizadas estão a profundidade máxima, a profundidade atual, uma referência para o pai, e também uma variável de controle para saber a qual dos lados do pai o filho pertence, direita ou esquerda.

Várias funções foram implementadas para auxiliar na manipulação de um indivíduo, como

as funções de inicialização *Grow*, *Full*. Além disso, dentre outras funções, também foram criados métodos auxiliares para retornar listas contendo todos os nodos da árvore do indivíduo, profundidade dos nodos, funções para expandir, transformar (mutação), e cruzar. Na implementação proposta existe um método que permite podar a árvore do indivíduo ao custo de uma mutação forçada, no entanto esse método foi abandonado ao implementar um cruzamento mais seletivo.

1.1.1 Funções de Inicialização

Foram implementadas as três funções de inicialização, Grow, Full, e Ramped half-and-half. O método implementado *Grow* permite a inicialização de um indivíduo de forma irregular, onde começa-se pela raiz e decide-se se seus filhos serão transformados em terminais ou funções com probabilidade $1/2$. Caso um filho seja transformado em um terminal, então a inicialização para aquele ramo da árvore é finalizada. Caso contrário, se um filho for transformado em uma função, então a o método de inicialização repete o processo para seus dois filhos. O processo se repete até que se obtenha a altura máxima permitida ou até que não tenha mais funções para serem inicializadas.

Em contrapartida, o método de inicialização *Full* implementado visa criar uma árvore balanceada. O mesmo começa a ser executado nos filhos da raiz do indivíduo, onde cada filho pode se tornar uma função ou terminal. Os mesmos são transformados em terminais quando a altura máxima da árvore é alcançada e são transformados em funções caso ainda não se tenha obtido a mesma. Dessa forma, funções são criadas sempre que a profundidade atual p_i for menor que a profundidade máxima. Já os terminais são inicializados apenas quando a profundidade p_i do nodo for igual a profundidade máxima.

Por fim, o método Ramped half-and-half foi implementado utilizando-se a combinação do método *Grow* e `Full` de forma a tentar melhorar a diversidade da população inicial.

1.2 Seleção

A seleção é uma importante etapa para garantir o sucesso do algoritmo, uma vez que a mesma permitirá selecionar soluções promissoras. O método utilizado é a seleção por torneio, onde são selecionados K indivíduos aleatoriamente e o que possuir a melhor *fitness* ganha o torneio. Para efetuar um cruzamento, a implementação proposta faz duas chamadas ao método de seleção, uma primeira vez para selecionar um indivíduo A e posteriormente para selecionar um indivíduo B.

Uma vez em posse dos indivíduos A e B, os mesmos são copiados e enviados para o cruzamento. O método de seleção, assim como o cruzamento é executado N vezes até que se tenha a quantidade necessária de elementos para que possa ser possível gerar uma nova população.

1.3 Cruzamento

Uma vez que deseja-se evitar *bloats*, o cruzamento proposto troca elementos de sub-árvores que possuem a mesma profundidade. Para isso, são recebidos da seleção dois indivíduos A e B. Posteriormente, é definido qual dos dois indivíduos possui o menor tamanho. Caso seja o indivíduo A, então é selecionado um nó A_i de sua árvore. É importante notar que a profundidade máxima de A_i é menor que a profundidade máxima de A_r e menor ou igual a profundidade máxima de qualquer nodo B_i pertencente a B, onde A_r é a raiz de A.

Para selecionar um nodo de B a ser trocado, a profundidade máxima p_i de cada nodo de B é calculada. Dessa forma, apenas nodos de B, com profundidade máxima p_i igual a profundidade do nodo selecionado em A são considerados para o cruzamento. Com essa proposta garante-se que a árvore nunca irá ultrapassar um tamanho máximo definido e que o contexto das árvores não será drasticamente modificado por eventuais podas de *bloats*. O elemento de profundidade máxima p_i é selecionado com probabilidade $1/|P_i|$, onde P_i é o conjunto que contém todos os elementos de profundidade máxima p_i. É importante notar que uma vez que B tem profundidade máxima maior que A, portanto qualquer nodo de A caberá em B mesmo considerando sub-árvores.

1.4 Mutação

A mutação proposta baseia-se na mutação de um ponto. Esse modelo foi escolhido para evitar grandes perdas de contexto dentro de um indivíduo. O nodo da árvore do indivíduo que é selecionado para mutação pode ser uma função ou um terminal, caso seja uma função o mesmo só pode ser transformado em outra função. Em contrapartida, caso o mesmo seja um terminal, o mesmo só pode ser transformado em outro terminal. Nesta proposta, uma mutação só ocorre quando um número aleatório $p_m \in [0, 1]$ é menor que a probabilidade total, ou seja, quando $p_m < prob_m$, onde $prob_m \in [0, 1]$ é a probabilidade de mutação.

O tipo de nodo a ser selecionado para mutação é obtido com probabilidade igual a 0.5, caso o tipo terminal seja escolhido, então um terminal é selecionado para ser transformado com probabilidade $1/t_t$, onde t_t é o total de terminais no indivíduo. Em contrapartida, caso o tipo função seja escolhido, uma função é selecionada para ser transformada com probabilidade $1/t_f$, onde t_f é o total de funções.

1.5 K-Means

O algoritmo K-Means também foi implementado nesta proposta. O mesmo tem a finalidade de categorizar amostras em diferentes *clusters* e é definido da seguinte forma. Seja $C = c_1, ...c_k$, onde $k = |C|$, o conjunto de *clusters* e $S = s_1, ...s_n$, onde $s_i \in S$ é um vetor de números reais, deseja-se associar $s_i \in S$ a um $c_j \in C$, de forma que $dist(s_i, c_j) = \min dist(s_i, c_h) \forall c_h \in C$. Ou seja, deseja-se categorizar s_i em um cluster c_j, onde a distância entre os dois é mínima. O algoritmo funciona em duas etapas, onde a primeira é responsável por atribuir um s_i a um c_j e outra é responsável por atualizar os centroides para cada $c_j \in C$.

Para permitir a implementação do mesmo foram utilizadas estruturas auxiliares para armazenamento da base de dados e também para o armazenamento de cada amostra. Uma vez que o objetivo do trabalho é a obtenção de uma função capaz de categorizar corretamente uma amostra, o mesmo é utilizado dentro dos indivíduos para auxiliar no cálculo da função de *fitness* utilizada.

1.6 V-Measure

O `V-Measure`, a média harmônica entre a homogeneidade e completude, também foi implementada. A mesma permite verificar o quão boa está uma predição sem a necessidade de conhecer o conjunto de samples a serem avaliados. Ela é definida como descrito pela Equação 1,

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \tag{1}$$

onde, h é a homogeneidade, c a completude, e β é uma variável que permite dar mais prioridade um ou outro. A completude por sua vez é definida como descrito pela Equação 3,

$$c = \begin{cases} 1 & \text{se } G(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{caso contrário} \end{cases} \tag{2}$$

onde, $H(K)$ representa a entropia entre os *clusters*. Por outro lado, a homogeneidade é definida como descrito pela Equação 3

$$h = \begin{cases} 1 & \text{se } G(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{caso contrário} \end{cases} \tag{3}$$

onde, $H(C)$ representa a entropia entre as classes.

A implementação feita foi validada com dados da posposta original apresentada por Rosenberg e Hirschberg [2] e também em comparação com a biblioteca sklearn disponível para a linguagem *Python*.

1.7 Avaliação

A avaliação de um indivíduo é feita de acordo com os centroides iniciais para o *dataset* e *V-Measure*. Em contrapartida, para efetuar a avaliação de um indivíduo é utilizada uma tabela de símbolos auxiliar.

Essa tabela é direcionada para dentro do algoritmo do k-means, para a função de distância ilustrada no Algoritmo 4, e ela permite armazenar o valor de cada terminal. Uma avaliação recursiva dos nodos da árvore do indivíduo atual é feita e permite calcular a distância entre os dois samples. Durante o percorrimento da árvore, caso um terminal seja alcançado, então o valor do mesmo é buscado na tabela de símbolos e associado ao devido elemento. Ao finalizar o cálculo da distância, a mesma é normalizada para apenas valores positivos e evitar que valores negativos sejam considerados como bons. Ao final do processo, a tabela de símbolos é esvaziada para se efetuar cálculos com novos valores.

EXPERIMENTOS

Os experimentos foram divididos em duas etapas. A primeira etapa refere-se a uma análise do impacto dos parâmetros e a segunda é referente a qualidade da solução obtida durante treinamento e testes do algoritmo proposto.

1.8 Análise de Parâmetros

A análise de parâmetros foi feita utilizando apenas a base de teste referente a detecção de câncer de mama. Foram analisados o impacto das três inicializações (Grow, Full, e Ramped) na média do melhor indivíduo durante as gerações, assim como os parâmetros de mutação e cruzamento. Para efetuar os testes o tamanho da população foi fixado em 200, o tamanho máximo do indivíduo em 7, quantidade de gerações em 30, e seleção por torneio com $k = 10$. Para os testes de inicialização da população, o cruzamento e mutação foram fixados em 0.8 e 0.1, respectivamente. A quantidade de iterações para o k-Means foi fixada em 10 para evitar gargalos. Os operadores utilizados para esses testes foram **+**, **-**, *****, e **/**.

Como pode ser observado no Gráfico 1, não há melhoria significativa na média de execução para cada geração entre os três tipos de inicialização. Observa-se um máximo próximo a 0.3 para o método Grow, e um empate entre o Full e o Ramped próximo de um V-Measure igual a 0.2.

De acordo com as análises, não foi observado ganho significativo em termos de ótimo local utilizando meios diferentes de inicialização para os parâmetros utilizados nos testes. Além disso, a taxa de mutação e cruzamento tem uma tendência em fazer o algoritmo convergir ou não para um ótimo local mais rapido e ficar estagnado.

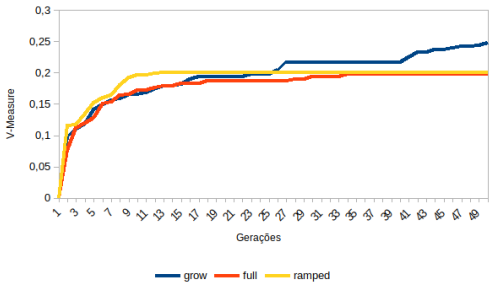

Gráfico 1: Média do melhor indivíduo para os métodos de inicialização Full, Grow, e Ramped.

Diferentemente, o tamanho da população afeta drasticamente a qualidade das soluções obtidas. Como ilustrado pelo Gráfico 2, uma população de apenas 30 indivíduos prejudica drasticamente a qualidade da solução. Porém, também pode-se observar que populações de 100 e 200 indivíduos proporcionam mais diversidade e uma evolução mais distante de ótimos locais.

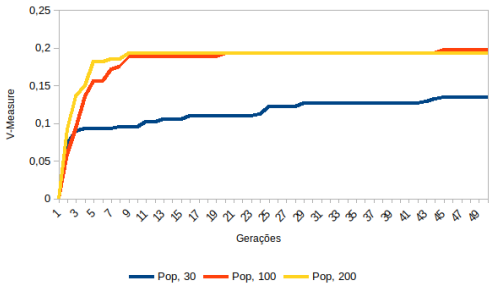

Gráfico 2: Média do melhor indivíduo para diferentes tamanhos de população.

Já com relação a probabilidade de mutação e cruzamento, como ilustrado no Gráfico 3, onde ambas estão foram avaliadas em 0.6 e 0.4, respectivamente, vemos um grande aumento da diversidade. Isso pode ser observado através de uma diferença considerável entra a média e o melhor indivíduo. Dessa forma, uma alta taxa pode acarretar em estagnação ou em tornar o algoritmo em uma abordagem aleatória.

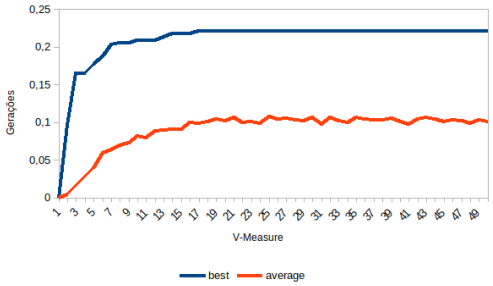

Gráfico 3: Média do melhor indivíduo e fitness médio com probabilidade de cruzamento em 0.6 e probabilidade de mutação em 0.4.

Em contrapartida, como ilustrado no Gráfico 4, uma alta taxa de cruzamento e baixa mutação reduzem drasticamente a diversidade. Dessa forma pode haver a tendência de se encontrar em um ótimo local bom mais facilidade.

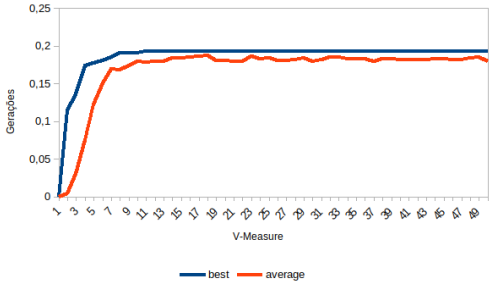

Gráfico 4: Média do melhor indivíduo e fitness médio com probabilidade de cruzamento em 1.0 e probabilidade de mutação em 0.01.

1.9 Avaliação da Melhor Solução Encontrada

A avaliação foi divididos em duas etapas, uma para o treinamento e outra para a avaliação. Duas bases de dados foram utilizadas, uma referente a câncer de mama e outra a tipos de vidros. Ambos os treinos e testes foram executados com os parâmetros descritos na Tabela 1.

Os parâmetros da 1 foram escolhidos pois demonstram pouca influencia para a quantidade de gerações utilizada nos testes. Em especial a taxa de mutação e cruzamento fixas em 0.8 e 0.1 aparentam ter um equilíbrio entre a média e o melhor indivíduo. O k-Means foi fixado em 5 para evitar gargalos, e também porque uma quantidade exacerbada de iterações não melhora a qualidade da solução final para a implementação proposta. O torneio foi fixado em apenas 10 indivíduos, porque dessa forma tem-se uma melhor diversidade nos indivíduos selecionados entre um torneio e outro. Além disso, o Elitismo foi utilizado em todas as iterações.

Para os experimentos descritos na Seção Treino e Teste, cada indivíduo contem as funções **+**, **-**, *****, e **/**. Além disso, constantes $c_i \in [0, 10]$ também foram adicionadas as árvores de forma aleatória durante dua inicialização. Cada base de dados contém 9 atributos, onde as bases de treinamento e teste sobre câncer possuem 91 e 23 elementos, respectivamente. Já as bases de treinamento e teste sobre vidros possuem 170 e 42 amostras, respectivamente.

Durante o treino, a convergência média do algoritmo foi analisada para os três tipos de inicializações de indivíduos, *Grow*, *Full*, e *Ramped half-and-half*. Os parâmetros que demonstraram melhores resultados foram utilizados para efetuar os testes. Em contrapartida, os testes foram executados apenas utilizando o melhor indivíduo obtido durante os treinos, onde a média e

Base	Pop.	Class.	Clus.	Tour.	Gen.	Ind.	K-means	Cross.	Mut.
Câncer	200	2	2	10	100	7	5	0.8	0.1
Vidro	200	7	7	10	30	7	5	0.8	0.1

Tabela 1: Parâmetros gerais.

Base	Max	Min	Média	Desvio	Var	Med Abaixo Med	Med Acima Med
Vidro	0.6	0.03	0.28	0.18	0.03	0,12	0,46
Câncer	0.34	0.01	0.16	0.14	0.02	0.01	0,31

Tabela 2: Estatísticas sobre o melhor indivíduo, sua média, desvio padrão, variância, e médias acima e abaixo da média, quando avaliados nas bases de teste.

o desvio padrão da *V-Measure* foram utilizados para avaliação.

Para coletar estatísticas sobre o comportamento do algoritmo em cada base de dados, é feita uma média de 4 execuções do algoritmo para uma determinada geração. Ou seja, o ponto 1 de uma seção de coleta é a média da primeira geração de 4 execuções. Dessa forma, pode-se observar o comportamento da média e também do desvio padrão para o melhor fitness e fitness médio ao longo das gerações.

Ambos o treinamento e avaliação foram efetuados utilizando o método de inicialização *Grow* isso pois o mesmo se mostrou mais eficiente por proporcionar indivíduos com menos nodos e mais variados.

1.10 Treino - Base de dados para detecção de câncer

O Gráfico 5 mostra o desvio padrão e a média do melhor indivíduo para a base de treinamento com relação a detecção de câncer de mama. Pode-se observar no mesmo que foi obtido um máximo de cerca de 0.2 para o V-Measure enquanto observa-se a queda do desvio padrão. Isso pode ocorrer, pois ao longo das gerações o desvio padrão do melhor indivíduo tende a se estabilizar por consequência do elitismo.

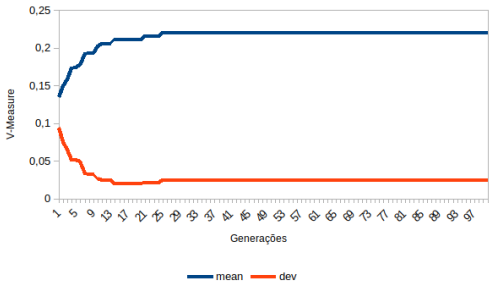

Gráfico 5: Média e desvio padrão do melhor indivíduo para todas as gerações utilizando a base de dados referente a detecção de câncer.

Já com relação ao fitness médio, como demonstrado no Gráfico 6, tem-se uma média não tão próxima do melhor indivíduo, de 0.12 e um desvio padrão em 0.02. Ao longo das gerações não se observa perturbações e também não há a existência de convergência da média. Isso pode indicar que alguns indivíduos não contribuem tanto para a solução final e estão gerando filhos cujo fitness é menor que o fitness médio dos pais.

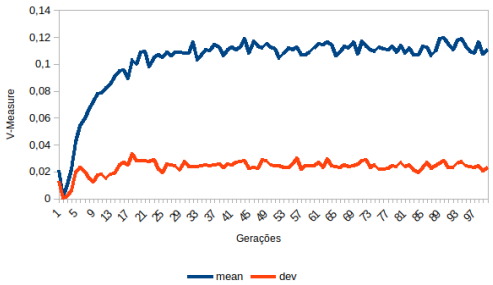

Gráfico 6: Média e desvio padrão do fitness médio para todas as gerações utilizando a base de dados referente a detecção de câncer.

O melhor indivíduo e sua média, ilustrados no Gráfico 7, mostram indícios de que houve uma estagnação da solução. Vários fatores podem ter contribuído para isso, desde de falta de representatividade de operadores (suficiência), até uma inicialização inicial da população desvantajosa.

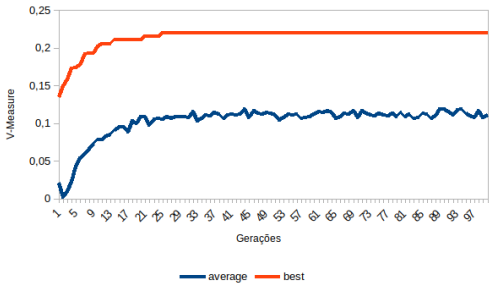

Gráfico 7: Média do melhor fitness e fitness médio para todas as gerações utilizando a base de dados referente a detecção de câncer.

1.11 Treino - Base de dados para detecção de tipos de vidro

O treinamento da base para classificação de tipos de vidros mostra comportamento semelhante ao observado na base sobre detecção de câncer de mama. Como mostra o Gráfico 8, que ilustra a média do melhor indivíduo e também a média

do desvio padrão, obteve-se um fitness máximo próximo de 0.6 e um desvio padrão mínimo de 0.3. Esse comportamento pode indicar a estabilização do melhor indivíduo ao longo das gerações para as diferentes execuções do algoritmo.

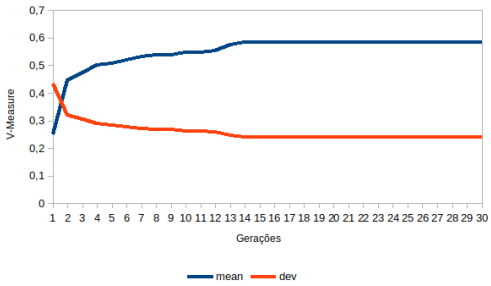

Gráfico 8: Média e desvio padrão do melhor indivíduo para todas as gerações utilizando a base de dados referente a classificação de vidros.

Diferentemente, o Gráfico 9 ilustra o desvio padrão da média e a média para cada geração. Pode-se observar que a média está longe do melhor indivíduo e ainda assim o algoritmo apresenta um desvio padrão baixo. Dessa forma, esse comportamento levanta indícios de que o algoritmo não irá conseguir convergir a um ótimo global pois está preso devido a estabilização do fitness médio proporcionando uma alta diversidade.

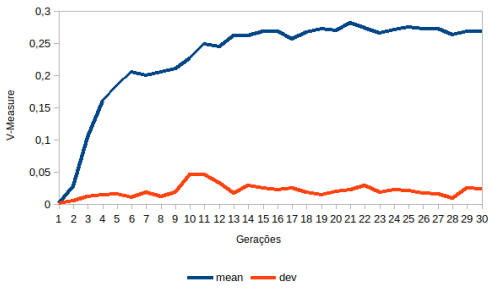

Gráfico 9: Média e desvio padrão do fitness médio para todas as gerações utilizando a base de dados referente a classificação de vidros.

Já no caso geral, como mostra o Gráfico 10, a solução encontrada para as 4 execuções apresenta uma diversidade muito alta, onde o máximo está proximo de 0.6 e a média próxima de 0.2 para o V-Measure.

1.12 Avaliação para Ambas as Bases de Dados

Os testes foram feitos sem conhecimento prévio da base de teste, ou seja os algoritmos não foram treinados com as mesmas. Pode observar-se na Tabela 2, que o algoritmo proposto obteve um fitness máximo na detecção de câncer de mama de 0.34, um mínimo de 0.01, com desvio padrão em torno de 0.14. Dessa forma, há indícios de que

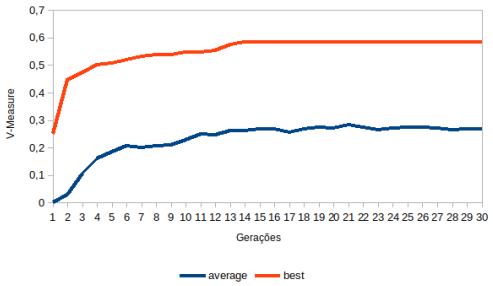

Gráfico 10: Média do melhor fitness e fitness médio para todas as gerações utilizando a base de dados referente a classificação de vidros.

o máximo local encontrado pode ser um outlier, que pode ser melhor estimado como $0.16^{+}_{-}0.14$, englobado pela média devido a um alto desvio.

Já com relação a classificação de vidros, foi averiguado um máximo próximo a 0.6, mínimo em 0.03, média em 0.28, e desvio em 0.18. Uma vez que houve também um desvio considerável, é possível inferir que o máximo de V-Measure encontrado pode ser considerado um outlier, onde uma estimativa local mais favorável se aproxima de um V-Measure igual a $0.28^{+}_{-}0.18$.

Conclusões

Neste trabalho foram implementados algoritmos para solucionar problemas no contexto da Programação Genética, algoritmo de clusterização k-means, e métrica V-Measure para cálcula de média entre entropias. Foram realizadas análises de parâmetro, com relação a métodos de inicialização da população, tamanho da população, taxa de mutação e cruzamento, onde resultados mostram a capacidade da proposta de convergir, manter ou extinguir a diversidade de acordo com alterações propostas. Além disso, também foram avaliadas duas bases de dados, uma referente a detecção de câncer de mama e outra referente a classificação de vidros.

Com relação a base de dados para detecção de mama foi observado um treinamento com convergência para ótimos locais e estagnação do algoritmo. A média se mantém longe do ótimo local encontrado para os parâmetros utilizados, o que sugere reduzir um pouco mais a diversidade para tentar melhorar a convergência do algoritmo. Além disso, também foi observado que parâmetros como o tamanho da população e inicialização da mesma podem afetar drasticamente a qualidade da solução obtida. Nos testes cegos, foi observado um máximo de $0.16^{+}_{-}0.14$ o que pode indicar a presença de muitos outliers.

Com relação aos testes referentes a base de dados para classificação de vidro, foi observado convergência para ótimos locais e também alta diversidade. Em todos os casos os melhores indivíduos ficaram acima da média e o desvio padrão

se mostrou alto. Nos testes cegos para a classificação de vidro, foi observado um máximo de $0.28^{+}_{-}0.18$. Em contrapartida, também foi obtida uma solução próxima de 0.6 para o V-Measure.

REFERENCES

[1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[2] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.