PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Análise de Algoritmos para o Ajuste de Parâmetros da SVM - Uma Aplicação na Predição de Função de Proteínas*

Marcos Felipe Martins Silva[1]
Cristiane Neri Nobre[2]

Resumo

A SVM é um algoritmo de aprendizagem supervisionada muito utilizado em problemas de classificação de dados. Entretanto, seu uso prático é limitado devido ao fato de que a qualidade da solução está diretamente relacionada à função *kernel* escolhida, e ao ajuste de seus parâmetros. Realizamos neste trabalho uma comparação entre um algoritmo genético (AG) e um algoritmo por enxame de partículas (PSO) no ajuste dos parâmetros γ e C da SVM. Após a realização de testes experimentais realizados no contexto da predição de função de proteínas, verificou-se que ambos AG e PSO são aptos para o ajuste dos parâmetros da SVM de maneira eficiente.

Palavras-chave: Proteínas, Predição de Função de Proteínas, Algoritmo Genético, Método de Enxame de Partículas, PSO, AG.

Abstract

SVM is a supervised learning algorithm widely used in data classification problems. However, its practical utilization is limited due to the fact that the quality of the solution is related to the chosen *kernel* function, and the adjustment of its parameters. In the present work, we compare a genetic algorithm to a particle swarm optimization (PSO) in setting the parameters γ and C of SVM. After running some experimental tests based on the prediction of proteins function, it is concluded that both GA and PSO are suitable to set the SVM parameters efficiently.

Keywords: Proteins, Prediction of protein function, Genetic Algorithm, Particle Swarm Optmization, PSO, GA.

---

*Artigo apresentado ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais como pré-requisito para obtenção do título de Bacharel em Ciência da Computação.

[1]Bacharelado em Ciência da Computação da PUC Minas, Brasil – marcosfelipeti@gmail.com.
[2]Doutor em Ciência da Computação, E-mail:nobre@pucminas.br
Instituto de Ciências Exatas e de Informática da PUC Minas, Brasil .

1 INTRODUÇÃO

A Máquina Vetor de Suporte, do inglês *Support Vector Machine* (SVM), é um algoritmo de aprendizagem supervisionada largamente utilizado em problemas de classificação de dados, tais como diagnósticos médicos (CONFORTI; GUIDO, 2010), reconhecimento de imagens (GUO; LI; CHAN, 2000), tomadas de decisão (SANGITAB; DESHMUKH, 2011), e bioinformática (RESENDE et al., 2012). Se comparado a outros classificadores, a SVM se destaca na sua capacidade de resolver os problemas de classificação binária lineares e não-lineares, tendo por base o encontro de um hiperplano que fará a distinção entre as classes dos exemplos de entrada no vetor de suporte.

A escolha da função *kernel* e seus parâmetros são cruciais na computação de similaridade entre os padrões de entrada e a representação dos mesmos no espaço vetorial da SVM. Muitas heurísticas tem sido escolhidas para o ajuste dos parâmetros livres da SVM, porém esses se diferem quanto ao tempo de execução computacional.

O mapeamento do código genético de um organismo também nomeado Projeto Genoma, possibilitou a identificação de muitas proteínas. Contudo, uma grande parte destas proteínas tem sua função desconhecida. O conhecimento da função estrutural de uma proteína pode trazer benefícios comerciais ou ser aplicado nas áreas de saúde, agropecuária e indústria, como abordado em (COSTA; BITTENCOURT; SOUTO, 2005) e (RESENDE et al., 2012). Devido à importância das proteínas em aspectos biológicos e industriais, o uso de técnicas computacionais é uma alternativa aos dispendiosos testes laboratoriais de cristalografia e raio-X para a predição de função das mesmas.

A proposta deste trabalho é analisar o comportamento de um Algoritmo Genético (AG), um algoritmo de enxame de partículas, do inglês *Particles Swarm Optmization* (PSO), e o *grid-search* no ajuste dos parâmetros da SVM. Para a realização dos testes, utilizaremos o contexto de predição de função de proteínas.

Este trabalho está estruturado da seguinte forma: conceitos teóricos do AG, PSO, SVM, e das enzimas analisadas, bem como os trabalhos relacionados, podem ser encontrados na Seção 2. A metodologia é apresentada na Seção 3 seguida dos principais resultados experimentais (Seção 4). Na Seção 5 encontram-se as conclusões sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

2.1 SVM Não Linear

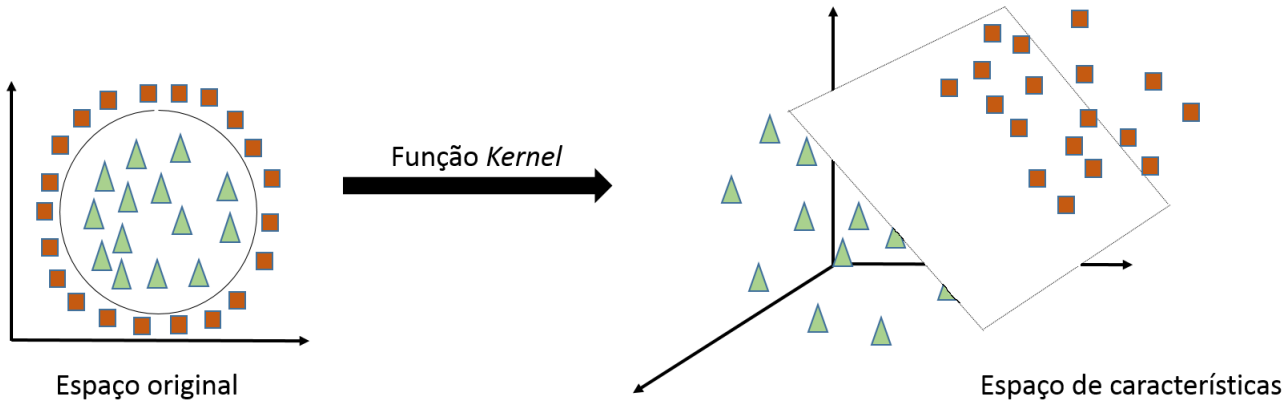
A SVM é baseada na Teoria de Aprendizagem Estatística cujo princípio é a Minimização de Risco Estrutural. O objetivo da SVM é encontrar um hiperplano que separa os exemplos de entrada em classes distintas no conjunto de treinamento, maximizando a distância entre tais

conjuntos através do hiperplano ótimo.

O algoritmo de aprendizagem criado por Vapnik e Lerner (1963) era utilizado para trabalhar com um conjunto de dados linearmente separáveis, ou que possuíssem uma distribuição aproximadamente linear. Entretanto, viu-se que em muitas aplicações não era possível dividir de maneira satisfatória os dados de treino por um hiperplano. Para a resolução de tal impasse, em 1992 Boser, Guyon e Vapnik (1992) criaram a SVM não linear.

A SVM trabalha com problemas de classificação não lineares através da escolha de uma função *kernel* que realiza o mapeamento do conjunto de treino de seu espaço original para um novo espaço de dimensão superior, também conhecido como espaço de características como abordado em (BEN-HUR; WESTON, 2010). A Figura 1 ilustra o exemplo de um mapeamento para o espaço de características realizado pela função *kernel*.

Figura 1 – Mapeamento do espaço original para o espaço de características



Fonte: Elaborado pelo autor

Considere um *kernel* K como sendo uma função que recebe dois pontos x_i e x_j no vetor de entradas e realiza o produto escalar desses dados no espaço de características (HERBRICH, 2002). Seja $\Phi : X \longrightarrow \Omega$ um mapeamento, em que X é o espaço de entradas e Ω denota o espaço de características. A escolha apropriada de Φ faz com que o conjunto de treinamento mapeado em Ω possa ser separado por uma SVM linear. Temos assim a seguinte equação:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{1}$$

É de extrema importância a escolha de uma função *kernel*, bem como o ajuste de seus parâmetros, uma vez que isso influencia diretamente nos resultados a serem obtidos pelo classificador (RESENDE et al., 2012). Os *kernels* mais utilizados na prática são os Polinomiais, os Gaussianos ou RBF (Radial-Basis Function) e os Sigmoidais. Cada um deles apresenta parâmetros que devem ser determinados pelo usuário ao utilizar a SVM. A Tabela 1 apresenta os exemplos das funções *kernel* abordadas.

O *kernel* polinomial para $d = 1$ é denominado linear. O parâmetro γ determina a largura da função de Gauss. Os parâmetros do *kernel* sigmoidal δ e k descrevem a escala dos dados de entrada (para $\delta > 0$) e o limiar de mapeamento, respectivamente.

Tabela 1 – Principais funções *kernel*

kernels	Função $K(x_i, x_j)$	Parâmetros
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$	δ k e d
Gaussiano (RBF)	$exp(-\gamma \left\|x_i - x_j\right\|^2)$	γ
Sigmoidal	$tanh(\delta(x_i \cdot x_j) + k)$	δ e k

Fonte: Adaptado de (LEIJOTO et al., 2014)

A SVM apresenta um outro parâmetro de Custo C que deve também ser ajustado. Tal parâmetro permite ao usuário controlar a solução de compromisso entre a maximização da margem (flexibilidade das margens dos hiperplanos) e a classificação do conjunto de treinamento sem erros (LESSMANN; STAHLBOCK; CRONE, 2006).

2.2 Algoritmo Genético e Elitismo

Os Algoritmos Genéticos (AGs), segundo (LINDEN, 2008) são algoritmos matemáticos estocásticos que se baseiam nos princípios de seleção natural descobertos por Charles Darwin. Estes são uma meta-heurística que constituem uma técnica de busca e otimização inspirada em recombinação genética. Um AG com o uso de elitismo, possui o fluxo apresentado pelo Algorithm 1:

Algorithm 1: Algoritmo Genético com Elitismo

Saída: Cromossomo com resultado ótimo

```
1 inicio
2     P ⟵ CriaPopulação(P);
3     AvaliaPopulação(P);
4     enquanto não atingir critério de convergência faça
5         Selecionados ⟵ SelecioneCromossomos(P);
6         Cruzamento(Selecionados);
7         Mutação(P);
8         Elitismo();
9         AvaliaPopulação(P);
```

Fonte: Elaborado pelo autor

CriaPopulação(P): Criação de uma população aleatória de cromossomos, onde cada cromossomo apresenta uma solução possível para o problema em questão. A codificação de um cromossomo geralmente é realizada em bits.

AvaliaPopulação(P): A avaliação da adaptabilidade (ou *fitness*) de cada cromossomo da população se dá através de uma função objetivo, que retorna um valor real e informa a chance que um cromossomo tem de sobreviver nas gerações vindouras.

SelecioneCromossomos(P): Seleção dos cromossomos mais adaptados que terão a chance de recombinar-se geneticamente.

Cruzamento(CromossomosSelecionados): Cruzamento de cromossomos que envolve uma recombinação genética, em que os filhos herdam características (genes) dos pais.

Mutação(P): Mutação é o que ocorre com a modificação de alguns bits de um cromossomo cujo objetivo é aumentar a diversidade genética da população.

Elitismo(): Ao criar-se uma nova população através do cruzamento e mutação, corre-se o risco de perder os melhores cromossomos. Na resolução de tal problema podemos utilizar o elitismo. Esse método consiste na inserção dos cromossomos com melhores valores de *fitness* na nova população.

2.3 O Método do Enxame de Partículas (PSO)

O método de enxame de partículas foi originalmente criado por Kennedy e Eberhart (1995). O objetivo principal de tal trabalho é o de simular o comportamento social do bando de pássaros matematicamente. Mais tarde os autores descobriram que com algumas modificações, o modelo de comportamento social poderia ser utilizado como uma heurística de otimização, o que foi aplicado no treinamento de uma Rede Neural Artificial.

Como descrito em (SOUZA; FAGUNDES, 2013) e (KENNEDY; EBERHART, 1995), quando os pássaros levantam voo, estes ficam inicialmente distribuídos de maneira aleatória no ar. Entretanto, após um certo período de tempo, os pássaros seguem todos em uma mesma direção, que indica as regiões onde podem ser encontrados alimento ou ninho. De maneira análoga, as partículas (pássaros) no PSO inicialmente são distribuídas em posições aleatórias, e conforme a velocidade de cada partícula, o enxame (bando) segue explorando uma região ótima de acordo com as partículas que obtiveram o melhor *fitness*. O Algoritmo 2 descreve a estrutura de um PSO simples:

Algorithm 2: PSO Simples

Saída: Partícula com resultado ótimo

```
1 inicio
2     E ⟵ InicializaPosiçãoEnxame(E);
3     InicializaVelocidadeEnxame(E);
4     AvaliaEnxame(E);
5     enquanto não atingir critério de convergência faça
6         AtualizaVelocidade(E);
7         AtualizaPosição(E);
8         AvaliaEnxame(P);
```

Fonte: Elaborado pelo autor

InicializaPosiçãoEnxame(E): Criação de um enxame de partículas com posições aleatórias no espaço de busca.

InicializaVelocidadeEnxame(E): Inicialização do vetor de velocidades de cada partícula do enxame.

AvaliaEnxame(P): A avaliação da adaptabilidade (ou *fitness*) de cada partícula do enxame se dá através de uma função objetivo, que retorna um valor real.

AtualizaVelocidade(E): A atualização da velocidade de cada partícula, no conceito de topologia global, é realizada de acordo com a Equação 2:

$$V_i(k+1) = V_i(k)*w+C_1*rand_1*(p\_best-X_i(k))+C_2*rand_2*(g\_best_i-X_i(k)) \quad (2)$$

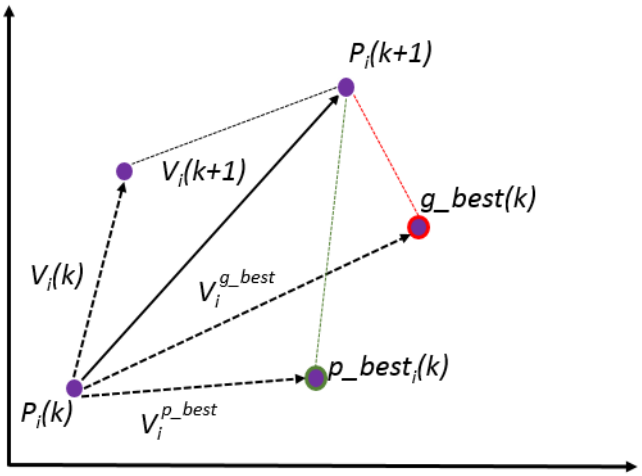
onde k é a iteração atual; V_i é a velocidade da partícula i; w fator de inércia (peso) da partícula; $rand_1$ e $rand_2$ são números aleatórios que evitam que a partícula fique presa numa solução ótima local; C_1 e C_2 são parâmetros de confiança que variam num intervalo de 1 a 4 (geralmente $C_1 = C_2$); X_i é a posição da partícula i; p_best é o valor do melhor *fitness* encontrado pela partícula até o momento; g_best_i é o melhor valor de *fitness* encontrado por uma partícula i do enxame.

AtualizaPosição(E): A posição de cada partícula deve ser computada em conformidade com a Equação 3:

$$X_i(k+1) = X_i(k) + V_i(k) \quad (3)$$

Verifica-se dessa maneira, que a solução ótima no algorimo PSO é obtida pela resultante de sua velocidade V atual, um conhecimento adquirido pela experiência da própria partícula (p_best) e o aprendizado adquirido através da comunidade global (g_best_i), como abordado em (HASSAN; COHANIM; WECK, 2005) e observado na Figura 2.

Figura 2 – Representação das atualizações da velocidade e posição de uma partícula no PSO



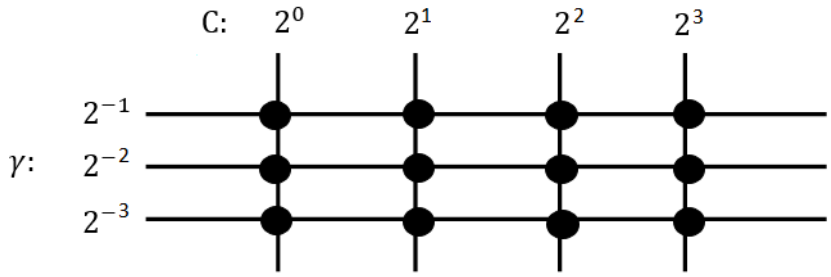
Fonte: Adaptado de (HASSAN; COHANIM; WECK, 2005)

2.4 *Grid-search*

O *Grid-search* é uma heurística que consiste em uma grade que realiza diferentes combinações de parâmetros baseadas em um intervalo de busca. Em conjunto com a SVM, essa heurística procura ajustar os parâmetros γ e C através do teste de sequências exponenciais para os valores encontrados durante a busca exaustiva (CHANG; LIN, 2011).

A solução do *Grid-search* está diretamente relacionada ao *step* escolhido em sua execução, sendo que, quanto menor o *step* escolhido, mais refinada é a solução. A Figura 3 ilustra um exemplo do funcionamento do *Grid-search*.

Figura 3 – Exemplo de Grid com um *step* igual a 1



Fonte: Elaborado pelo autor

2.5 Enzimas

As enzimas segundo (NELSON; COX, 2011) são proteínas catalisadoras das reações bioquímicas. Tal característica mostra que somente com sua presença, e sem serem consumidas durante o processo, as enzimas conseguem acelerar os processos bioquímicos. O número de

transformações moleculares das enzimas mensura a eficiência das mesmas como catalisadoras.

Segundo os critérios estabelecidos pela União Internacional de Bioquímica (IUB), e como pode ser visto em (DOBSON; DOIG, 2005) e (NELSON; COX, 2011), as enzimas estabelecem seis classes. As Oxidoredutases são enzimas catalisadoras de reações de transferência de elétrons (o substrato oxidado é um hidrogênio), isto é, reações de oxirredução. As Transferases catalisam a transferência de grupos entre duas moléculas, tais grupamentos funcionais podem ser amina, fosfato, acil, dentre outros. As enzimas que catalisam a reação de hidrólise de várias ligações covalentes são conhecidas como Hidrolases. As Liases são enzimas que catalisam a cisão de ligações C-C, C-O, C-N, através de hidrólise ou oxidação; estas realizam também a remoção de moléculas de água, amônia e gás carbônico. As Isomerases realizam a catálise de reações de interconversão entre isômeros ópticos ou geométricos. Finalmente, as Ligases são enzimas que catalisam reações de síntese de uma nova molécula a partir da ligação entre duas moléculas, com a concomitante hidrólise de ATP (a custa de energia).

Cada enzima descrita recebe um número de classificação, conhecido por E.C. (Enzyme Commission) de acordo com a International Union of Biochemistry and Molecular Biology (IUBMB), esse número é composto por 4 dígitos: 1) Classe; 2) Subclasse dentro da classe; 3) grupos químicos específicos que participam da reação; 4) a enzima, propriamente dita.

2.6 Trabalhos Correlatos

Ren e Bai (2010) realizaram uma comparação entre a utilização de um algoritmo genético e um algoritmo de enxame de partículas para a otimização dos parâmetros da máquina vetor de suporte . Os autores investigaram como o tamanho da população (ou enxame) influencia na solução encontrada por ambos os algoritmos implementados. Nos experimentos o tamanho da população variou de 10 a 30 cromossomos (ou partículas), e a conclusão final é que ambos, GA e PSO, são aptos para o ajuste dos parâmetros da SVM a um custo computacional aceitável em comparação com o *grid-search*.

No trabalho desenvolvido por Rojas e Fernandez-Reyes (2005), o ajuste dos parâmetros da SVM se deu através do uso de algoritmos genéticos. Os experimentos foram realizados no contexto que envolvem problemas de classificação, e os autores utilizaram uma variedade de bases de dados que envolviam dados reais e artificiais. A RBF (*Radial Basis Function*) foi escolhida nos experimentos e os autores atingiram um desempenho superior a 98% nos dados extraídos da base *human serum proteomic*.

Em (HUANG; WANG, 2006) os autores utilizaram um algoritmo genético para a realização de duas tarefas simultâneas: a seleção de características e o ajuste dos parâmetros da máquina vetor de suporte. Para os experimentos, Huang e Wang fizeram o uso de 11 conjuntos de dados reais extraídos da base de dados UCI. A função escolhida para os experimentos foi a RBF (*Radial Basis Function*) e, comparando-se o AG com o *grid-search*, os autores verificaram que o uso do AG obtinha uma melhor precisão devido sua função de também selecionar as

características dos conjuntos de dados.

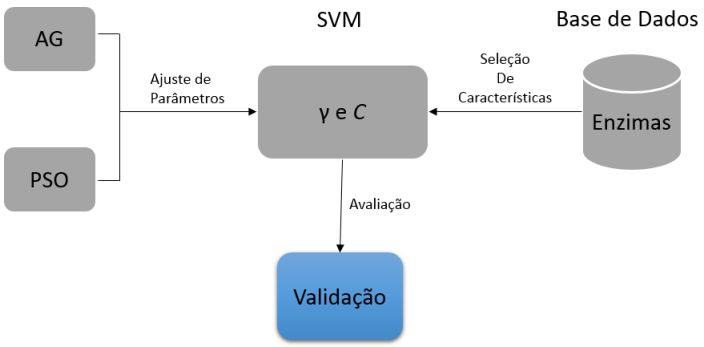
İlhan e Tezel (2013) desenvolveram um algoritmo genético para a seleção de rótulos SNPs (*Single Nucleotide Polymorphisms*). Além disso, os autores utilizaram o algoritmo de enxame de partículas com topologia global (PSO) para a otimização dos parâmetros C e γ da SVM. Para a realização dos testes os autores definiram um enxame de 20 partículas, valores de confiança (C_1 e C_2) igual 2 e um valor de inércia igual a 1. Tais valores foram escolhidos após vários testes do tipo tentativa e erro.

Em (LIN et al., 2008) uma metodologia utilizando o algoritmo de enxame de partículas (PSO) foi desenvolvida para a seleção de características e otimização de parâmetros da SVM. Os autores conseguiram resultados de precisão superiores às metodologias que utilizaram o *grid-search* no mesmo contexto, além de concluírem que o modelo PSO-SVM desenvolvido possui resultados similares a outros modelos GA-PSO presentes na literatura.

3 MATERIAIS E MÉTODOS

O AG e o PSO discutidos no presente trabalho foram desenvolvidos utilizando-se a linguagem C, uma vez que esses executarão em conjunto com a C-SVM, uma SVM multi-classe desenvolvida por (CHANG; LIN, 2011). De acordo com outros AGs desenvolvidos na literatura para o ajuste de parâmetros, concluímos que a RBF é a função *kernel* que apresenta os melhores resultados na classificação. Sendo assim, este trabalho se ocupa do ajuste dos parâmetros γ e C da SVM. A Figura 7 mostra o fluxograma que explicita a metodologia adotada.

Figura 4 – Fluxograma da metodologia adotada



Fonte: Elaborado pelo autor

3.1 Base de Dados

O presente trabalho utilizou o mesmo conjunto de proteínas usado por (LEIJOTO et al., 2014) e (SANTOS; ZARATE; NOBRE, 2015) para a avaliação do AG e PSO. Tais proteínas,

foram extraídas do PDB (*Protein Data Bank*) (BERMAN et al., 2000), que é o banco de proteínas mais completo e utilizado por trabalhos da mesma ordem na literatura. A Tabela 2 mostra a quantidade enzimas de cada classe que foram utilizadas para os experimentos.

Tabela 2 – Classes de Enzimas

EC	Classe	Quantidade
1	Oxidoreductases	76
2	Transferases	120
3	Hidrolases	161
4	Liases	60
5	Isomerases	57
6	Ligases	18

Fonte: Dados da pesquisa

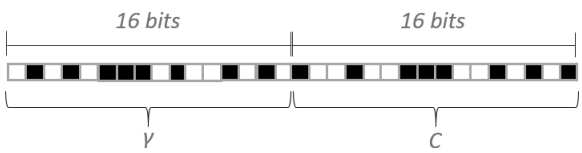
No trabalho desenvolvido por (LEIJOTO et al., 2014) foram selecionadas 11 características físico-químicas das enzimas com o uso de um algoritmo genético; com isso os autores conseguiram uma taxa de precisão de 71%. Em (SANTOS; ZARATE; NOBRE, 2015) os autores desenvolveram um AG paralelo para a seleção de 19 características físico-químicas das enzimas, e com essa técnica, uma taxa de precisão de 78% foi alcançada. Os trabalhos citados utilizaram o algoritmo de busca *grid-search* para o ajuste de parâmetros γ e C da SVM.

As características físico-químicas utilizadas por (LEIJOTO et al., 2014) e (SANTOS; ZARATE; NOBRE, 2015) foram extraídas do Sting_DB. Esse é uma base de dados do Laboratório de Biologia Computacional da Embrapa Brasil, que contém uma variedade de características extraídas de todas as estruturas que formam uma proteína. Os autores citados utilizaram o módulo nomeado *Java Protein Dossier* (NESHICH et al., 2004) que possui um total de 338 características físico-químicas das proteínas.

3.2 AG-SVM

Para a codificação de cada cromossomo do AG elaborado, utilizamos um vetor de 32 bits. Assim, os dezesseis primeiros bits do nosso cromossomo representam o parâmetro γ e os dezesseis bits restantes, são atribuídos ao ajuste do parâmetro C (custo). Na Figura 5, detalhamos a representação de um cromossomo da nossa população.

Figura 5 – Representação de um cromossomo



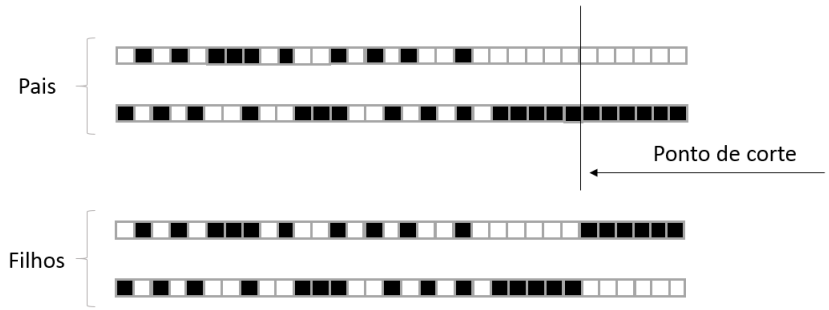
Fonte: Elaborado pelo autor

O *fitness* de cada cromossomo da nossa população foi calculado utilizando-se a SVM

como a função de avaliação. Após a avaliação de cada membro da nossa população, iniciou-se o processo de seleção dos cromossomos que deverão realizar recombinação genética. Para tanto, o método da seleção por roleta foi utilizado. Esse método consiste na seleção de um cromossomo baseado em seu valor de *fitness*, onde, quanto maior a sua adaptabilidade, maior a chance de ser selecionado (SOARES, 1997).

Neste trabalho foram implementados quatro tipos de operadores de cruzamento diferentes: cruzamento com um ponto de corte, cruzamento com dois pontos de corte, cruzamento com uso de uma máscara binária, e cruzamento através de operadores *AND*. Os melhores resultados foram obtidos com o uso do operador de um ponto de corte. A literatura recomenda uma probabilidade de cruzamento entre 60 e 90%, e no presente trabalho 70% dos indivíduos tiveram a chance de se reproduzir. A Figura 6 detalha o cruzamento com um ponto de corte adotado.

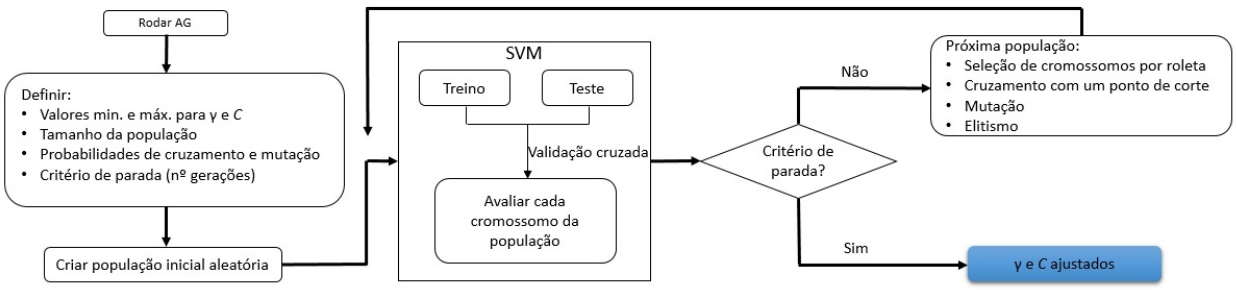
Figura 6 – Método de cruzamento com um ponto de corte



Fonte: Elaborado pelo autor

Com o objetivo de aumentar a diversidade da população, o método de mutação criado realiza a alteração de até quatro bits (12,5% do número de bits do cromossomo) aleatoriamente, tais valores foram escolhidos após a a realização de diversos testes experimentais. Um fluxograma do funcionamento do AG-SVM é apresentado na Figura 6.

Figura 7 – AG-SVM



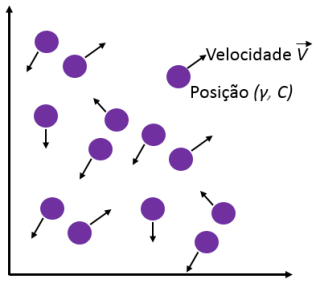
Fonte: Elaborado pelo autor

3.3 PSO-SVM

Uma partícula no PSO desenvolvido neste trabalho é representada por um vetor de componentes de duas posições, como mostrado na Figura 8. A primeira e segunda posições do vetor,

foram atribuídas aos parâmetros γ e C respectivamente, em que tais componentes representam a posição da partícula no espaço de busca.

Figura 8 – Representação de uma partícula no PSO

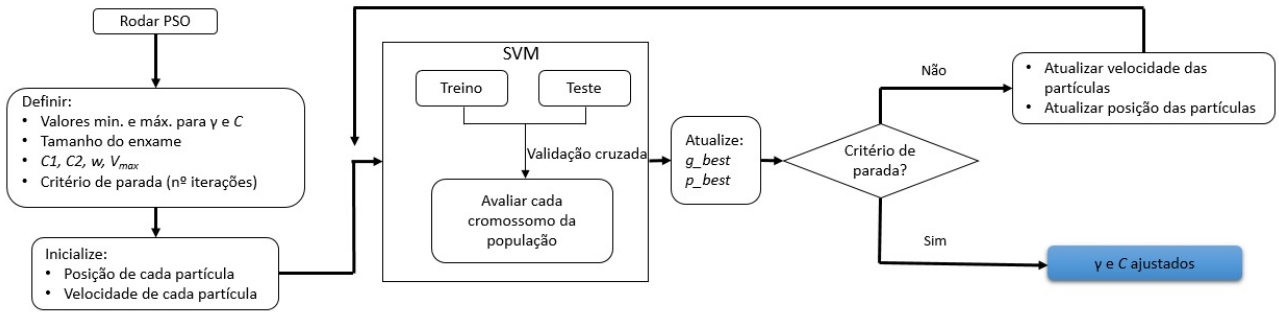


Fonte: Elaborado pelo autor

Após a realização de diversos testes, definimos neste trabalho uma velocidade máxima V_i para cada partícula i igual a 10, e os valores de C_1 e C_2 foram testados num intervalo de 1 a 4, sendo este último o que apresentou os melhores resultados. O fator de inércia w varia de acordo com o número de iterações e é obtido pelo cálculo fracionário entre um limite de peso superior e inferior iguais a 0,3 e 0,1 respectivamente.

A SVM foi utilizada como a função de avaliação de cada partícula do nosso enxame, como exibido na Figura 9. A partir do *fitness* de cada partícula, os valores de p_best e g_best são atualizados a cada iteração.

Figura 9 – PSO-SVM



Fonte: Elaborado pelo autor

3.4 Validação e Avaliação

O processo de validação do presente trabalho se baseia no método de validação cruzada. Isso é conhecido como *k-fold* e consiste em dividir o conjunto de dados em k subconjuntos de mesma cardinalidade aproximadamente. Dessa maneira, cada grupo de teste tem 10% das proteínas totais, enquanto o conjunto de treino contém 90% do total. Os conjuntos de treino e teste variam então em um intervalo de k iterações, de maneira que todos os conjuntos são utilizados

como treino e teste (KOHAVI et al., 1995). Como apresentado na literatura, consideramos aqui um valor de k igual a 10.

Para avaliar o desempenho do classificador, foram utilizados três medidas: precisão, sensibilidade e F-measure:

Precisão: A precisão é a taxa de proteínas da classe analisada que foram classificadas corretamente sobre todas as que foram classificadas nesta classe. A Equação 4 definecomo a precisão é calculada.

$$P = \frac{VP}{VP + FP} \tag{4}$$

Sensibilidade: A sensibilidade é a taxa de proteínas da classe analisada que foram corretamente classificadas sobre todas as proteínas dessa classe. A sensibilidade é calculada conforme a Equação 5.

$$S = \frac{VP}{VP + FN} \tag{5}$$

F-measure: Média harmônica de Precisão e Sensibilidade e pode ser calculada segundo a Equação 6.

$$F - measure = \frac{2 * VP}{2 * VP + FP + FN} \tag{6}$$

sendo VP=verdadeiros positivos, FP=falsos positivos e FN=falsos negativos.

4 RESULTADOS E DISCUSSÕES

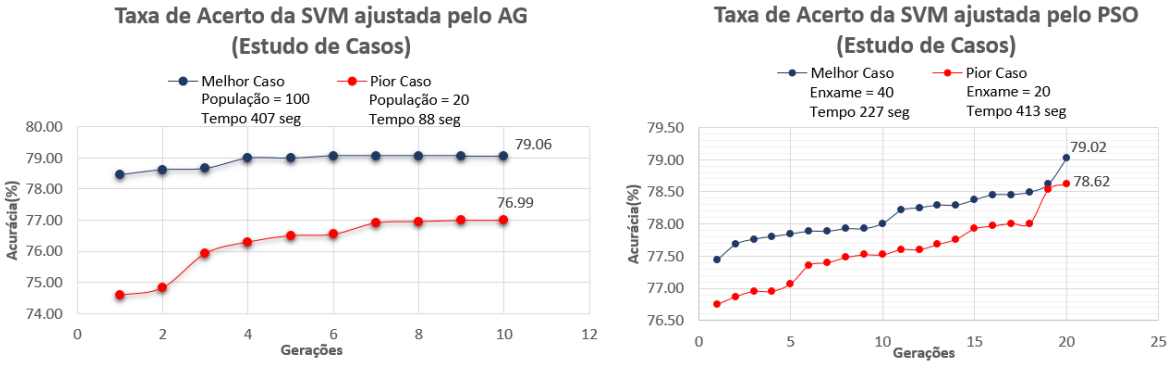
O objetivo desta seção é apresentar os resultados obtidos a partir da metodologia adotada. Uma comparação entre o AG, o PSO e o *grid-search* foi feita através da análise do valor de acurácia médio de todas as classes e o tempo de execução de cada algoritmo. Apresentaremos também, uma comparação do presente trabalho com os trabalhos desenvolvidos por (SANTOS; ZARATE; NOBRE, 2015) e (LEIJOTO et al., 2014), no que diz respeito aos critérios de validação (Precisão, Sensibilidade e F-measure) no contexto da predição de função de proteínas, visto que as bases de dados utilizados pelos autores foram as mesmas deste trabalho. Para a obtenção de tais métricas de validação, utilizamos a Weka (*Waikato Environment for Knowledge Analysis*), um conjunto de programas de máquinas de aprendizado que inclui a LIBSVM dentro de seu ambiente de execução (HALL et al., 2009).

Os experimentos foram realizados da seguinte maneira: Para ambos os algoritmos (AG e PSO) uma combinação entre o tamanho da população (ou enxame) e o número de gerações (ou iterações) foram executados 10 vezes para cada combinação. O número de representantes no AG e PSO variou num intervalo de 20, 40, 60, 80 e 100; e o critério de parada dos algoritmos compreendeu o intervalo de 10, 20 e 30 iterações. Para a comparação do tempo médio de execução, variamos o *step* do *grid-search* em 0,1; 0,5; 1,0 e 2,0. Os algoritmos foram executados em um computador Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, com 8 MB de cache e 12 GB de memória RAM.

4.1 Experimento com a Seleção de 19 Características

O Gráfico 1 mostra os resultados do melhor e pior caso no experimento realizado com as 19 características físico-químicas utilizadas por (SANTOS; ZARATE; NOBRE, 2015). No primeiro gráfico (a) temos os valores de acurácia encontrados com a execução do Algoritmo Genético. As taxas de acerto encontradas com a execução do algoritmo de enxame de partículas são mostradas no segundo gráfico (b). Os valores de γ e C para o AG foram 0.023463 e 4.715086 respectivamente. O PSO encontrou um γ=0.030510 e C=29.460919 para obtenção dos melhores valores de acurácia média.

Gráfico 1 - Estudo de Casos do AG e PSO (19 características selecionadas)



Fonte: Elaborado pelo autor

Pelo Gráfico 1, verificamos que o AG com uma população de 100 cromossomos quando executado por 10 gerações, obteve uma taxa de acerto média de 79,06%. Isto é, de um conjunto com 492 instâncias, 389 foram classificadas de maneira correta. Por outro lado, o tempo de execução médio do PSO, em seu melhor caso, foi aproximadamente 2 vezes superior ao melhor caso do AG, onde o algoritmo de enxame de partículas acertou 388 do mesmo conjunto de 492 instâncias.

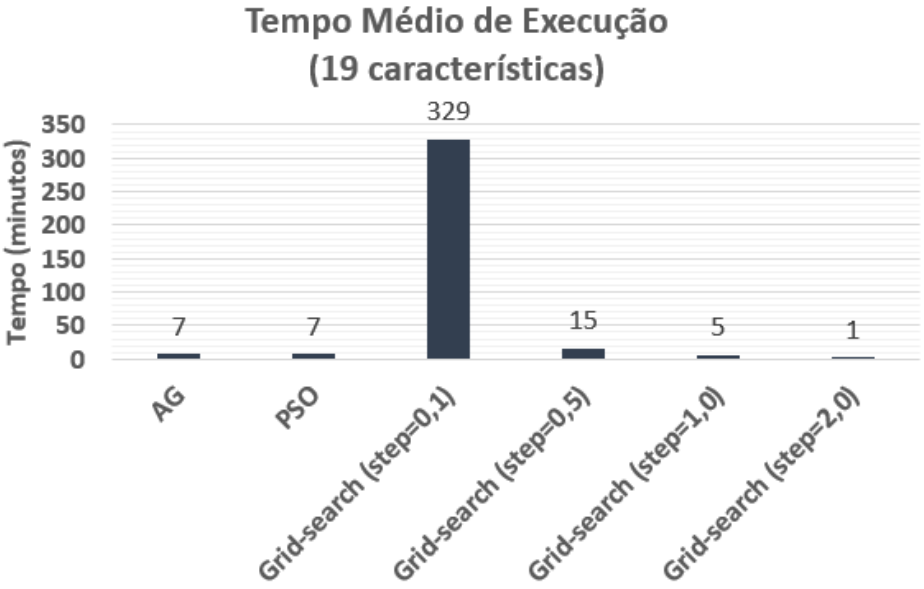
O pior caso do PSO alcançou uma taxa de acerto média de 78.62% com um tamanho de enxame igual a 20, executado por 20 iterações e um tempo de execução igual a 413 segundos. A execução do algoritmo genético em seu pior caso obteve uma acurácia média de 76,99%, com uma população de 20 cromossomos, executados por 10 gerações, e um tempo de execução de 88 segundos. Temos assim, uma diferença de 325 segundos na execução dos piores casos dos dois algoritmos.

Comparando a variação dos valores de acurácia média entre o melhor e o pior caso do AG, encontramos uma diferença de 2,2%, contra 0,41% na comparação dos estudos de casos do PSO. Isso pode ser explicado quando verificamos que a característica de uma partícula no PSO é de se comportar de maneira semelhante às suas vizinhas, o que reduz a variabilidade entre os valores de *fitness* no enxame.

O Gráfico 2 mostra o tempo médio de execução para os algoritmos AG, PSO e *Grid-search*. Observa-se que o *Grid-search* com um *step* igual a 0,1 demorou em média 329 minutos

para ser executado, tornando-se dessa maneira, inviável. Com um *step* igual a 1 o *Grid-search* teve o melhor desempenho. O AG e o PSO tiveram um tempo médio de execução de aproximadamente 7 minutos, e todos os algoritmos atingiram uma acurácia média de 79%.

Gráfico 2 - Comparação entre o tempo de execução médio entre os algoritmos propostos



Fonte: Elaborado pelo autor

As matrizes de confusão para os algoritmos AG e PSO são mostradas nas Tabelas 3 e 4, respectivamente. A Tabela 5 apresenta os resultados das métricas de avaliação (Precisão, Sensibilidade e F-measure) para os algoritmos AG e PSO comparados aos valores encontrados em (SANTOS; ZARATE; NOBRE, 2015).

Tabela 3 – Matriz de Confusão metodologia AG-SVM

	Classes Preditas pelo Classificador					
Classes Reais	Oxidoredutases	Transferases	Hidrolases	Liases	Isomerases	Ligases
Oxidoredutases	46	21	1	2	5	1
Transferases	12	98	2	6	2	0
Hidrolases	0	0	161	0	0	0
Liases	9	14	0	34	2	1
Isomerases	8	6	1	1	41	0
Ligases	4	1	1	1	2	9

Fonte: Dados da Pesquisa

Tabela 4 – Matriz de Confusão metodologia PSO-SVM

	Classes Preditas pelo Classificador					
Classes Reais	Oxidoredutases	Transferases	Hidrolases	Liases	Isomerases	Ligases
Oxidoredutases	44	20	2	3	6	1
Transferases	12	99	3	4	2	0
Hidrolases	0	0	161	0	0	0
Liases	9	13	0	34	3	1
Isomerases	8	7	1	1	40	0
Ligases	4	0	1	1	2	10

Fonte: Dados da Pesquisa

Pelas Tabelas 3 e 4 observa-se que das 161 Hidrolases utilizadas, todas foram corretamente classificadas, o que contribuiu para o aumento significativo da média em todas as métricas abordadas, sobretudo para a sensibilidade. Pode-se concluir também que as metodologias

(AG e PSO) classificaram as classes de maneira similar, com uma pequena diferença entre a classificação das Oxidoredutases, Transferases, Isomerases e Ligases.

Tabela 5 – Validação para a base de Santos, Zarate e Nobre (2015), considerando-se os três métodos analisados

	AG			PSO			Santos		
	Precisão(%)	Sensibilidade(%)	F-measure(%)	Precisão(%)	Sensibilidade(%)	F-measure(%)	Precisão(%)	Sensibilidade(%)	F-measure(%)
Oxidoredutases	58,2	65,8	59,4	57,1	57,9	57,5	57,1	57,9	57,1
Transferases	70,0	81,7	75,4	71,2	82,5	76,4	70,7	82,5	75,7
Hidrolase	97,0	100	98,5	95,8	100	97,9	95,8	100	99,7
Liases	77,3	56,7	65,4	79,1	56,7	66,0	78,6	55,0	63,0
Isomerases	78,8	71,9	72,2	75,5	70,2	72,7	75,5	70,2	71,4
Ligases	81,8	50,0	62,1	83,3	55,6	66,7	83,3	55,6	71,4
Média	77,2	70,1	72,7	77,0	70,5	72,9	76,8	70,2	70,7

Fonte: Dados da Pesquisa

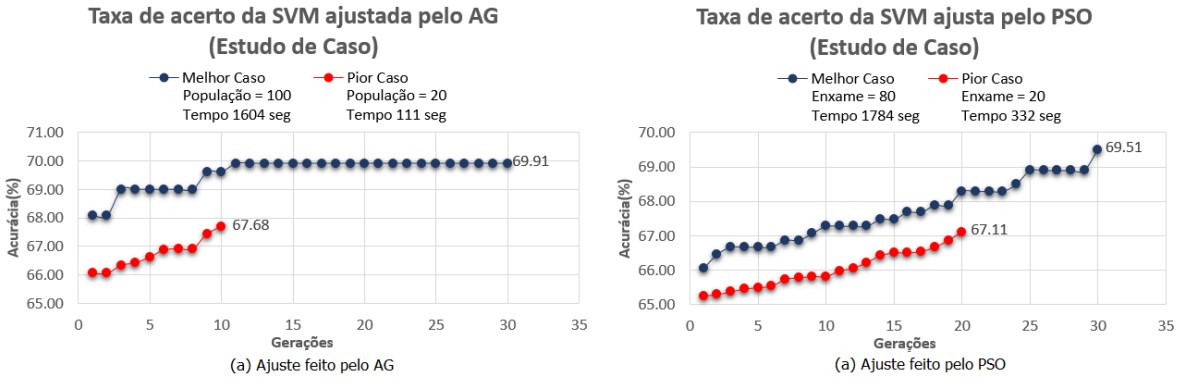
A precisão média foi de 77,2% para o Algoritmo Genético, 77,0% para o algoritmo de enxame de partículas e a metodologia adotada por (SANTOS; ZARATE; NOBRE, 2015) obteve 76,8% de precisão média. Observa-se aqui, que os valores não apresentam diferença significativa. Em todas as análises, as Oxidoredutases foram as que obtiveram a menor taxa de precisão com 58,2% para o AG e 57,1% para o PSO e a metodologia adotada em (SANTOS; ZARATE; NOBRE, 2015). Isso se deve ao fato de que 33 enzimas foram classificadas erroneamente como pertencentes às Oxidoredutases na execução do PSO.

Sabe-se que sensibilidade é a capacidade que um teste tem de discriminar, dentre as instâncias de uma classe analisada, quais são as que pertencem efetivamente à classe. Verifica-se que a metodologia proposta obteve um valor de sensibilidade médio similar à taxa de sensibilidade encontrada no trabalho desenvolvido por (SANTOS; ZARATE; NOBRE, 2015). O PSO que alcançou uma taxa de sensibilidade média de 70,5% contra uma taxa de 70,2% para o AG e a metodologia de (SANTOS; ZARATE; NOBRE, 2015). A menor sensibilidade foi obtida pelas Ligases, sendo estas o conjunto com o menor número de instâncias. Dessa maneira, de um total de 18 instâncias, 9 foram classificadas como pertencentes a outras classes de enzimas.

4.2 Experimento com a Seleção de 11 Características

Os resultados do estudo de casos no experimento utilizando as 11 características físico-químicas selecionadas por (LEIJOTO et al., 2014) são mostrados no Gráfico 3. No primeiro gráfico (a) temos as taxas de acerto médio obtidas com o AG e no segundo gráfico (b) os valores de acurácia média encontrados com a execução do PSO. A execução do AG encontrou os valores ótimos de γ=0.001110 e C=19.262581. O PSO encontrou os valores de γ e C iguais a 0.001174 e 18.542569 respectivamente.

Gráfico 3 - Estudo de Casos do AG e PSO (11 características selecionadas)



Fonte: Elaborado pelo autor

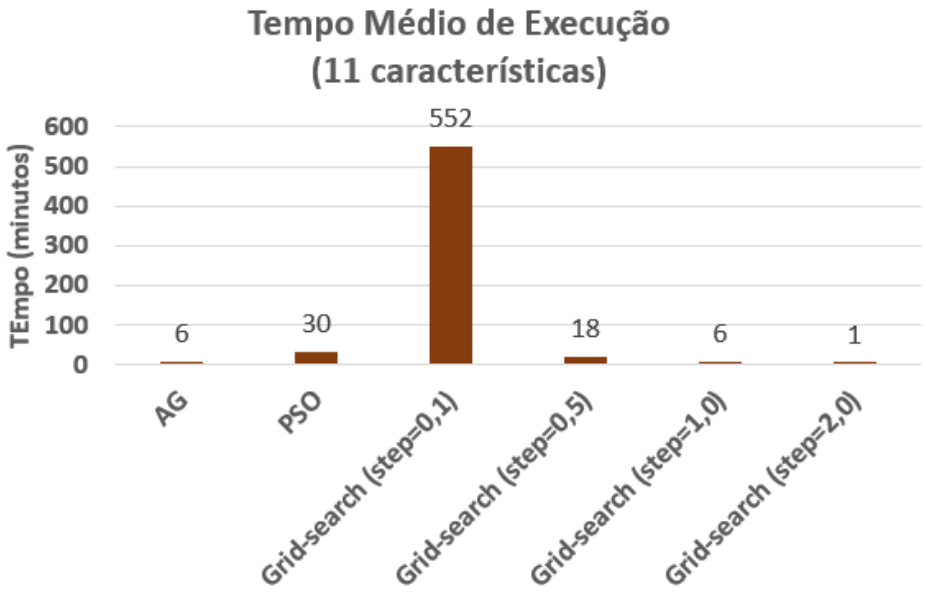
O Algoritmo Genético neste experimento, alcançou uma taxa de acerto média de 70,63%, com uma população de 100 cromossomos e que foram evoluídos por 30 gerações, como visto no primeiro gráfico (a). Verificamos ainda que o melhor caso do AG possui uma performance média superior ao melhor caso do PSO, onde temos uma diferença de tempo de execução de 180 segundos entre os algoritmos em seus melhores casos de execução.

Como mostrado nos gráficos (a) e (b), o pior caso do AG foi superior em taxa de acerto média e tempo de execução se comparado ao pior caso do PSO. Temos neste caso, uma taxa de 0,57% de diferença para os valores de acurácia média entre os algoritmos, e o AG foi aproximadamente 3 vezes mais rápido em tempo de execução que o PSO.

O segundo gráfico (b) mostra uma diferença mínima dos valores de acurácia ao longo das execuções do o melhor e pior caso do PSO. Isso é explicado pela tendência que o PSO tem de ficar preso em um ótimo local, uma vez que esse algoritmo não possui os operadores genéticos do AG para o aumento da diversidade na população.

O Gráfico 4 mostra o tempo médio de execução para os algoritmos AG, PSO e *Grid-search* no contexto das 11 características selecionadas por (LEIJOTO et al., 2014). Uma vez mais o *Grid-search* com um *step* igual a 2 o teve o melhor desempenho, seguidos do PSO com um tempo médio de 30 minutos e o AG demorou aproximadamente 6 minutos para ser executado nesse contexto. Todos os algoritmos atingiram uma acurácia média de aproximadamente 70%.

Gráfico 4 - Comparação entre o tempo de execução médio entre os algoritmos propostos



Fonte: Elaborado pelo autor

As Tabelas 6 e 7 explicitam as matrizes de confusão para os algoritmos AG e PSO executados com a base de características selecionadas por (LEIJOTO et al., 2014). As métricas de avaliação para os algoritmos AG e PSO, em comparação aos valores encontrados em (LEIJOTO et al., 2014), são exibidos na Tabela 8.

Tabela 6 – Matriz de Confusão metodologia AG-SVM

	Classes Preditas pelo Classificador					
Classes Reais	Oxidoredutases	Transferases	Hidrolases	Liases	Isomerases	Ligases
Oxidoredutases	50	10	8	2	5	1
Transferases	6	89	16	6	2	1
Hidrolases	7	25	120	5	3	1
Liases	4	13	9	34	0	0
Isomerases	5	7	5	1	39	0
Ligases	0	1	1	3	1	12

Fonte: Dados da Pesquisa

Tabela 7 – Matriz de Confusão metodologia PSO-SVM

	Classes Preditas pelo Classificador					
Classes Reais	Oxidoredutases	Transferases	Hidrolases	Liases	Isomerases	Ligases
Oxidoredutases	49	11	8	2	5	1
Transferases	6	89	16	6	2	1
Hidrolases	7	25	119	6	3	1
Liases	4	13	9	34	0	0
Isomerases	5	5	7	1	39	0
Ligases	0	1	1	3	1	12

Fonte: Dados da Pesquisa

Analisando as matrizes de confusão, pode-se observar que muitas instâncias foram classificadas como pertencentes a diferentes classes. Exemplo disso são as Transferases que na Tabela 8 apresentaram o menor valor de precisão em todoas as análises feitas. Observa-se ainda, que as metodologias (AG e PSO) obtiveram uma diferença não significativa na classificação das classes de enzimas analisadas por (LEIJOTO et al., 2014).

Tabela 8 – Validação para a base de Leijoto et al. (2014), considerando-se os três métodos analisados

	AG			PSO			Leijoto		
	Precisão(%)	Sensibilidade(%)	F-measure(%)	Precisão(%)	Sensibilidade(%)	F-measure(%)	Precisão(%)	Sensibilidade(%)	F-measure(%)
Oxidoredutases	69,4	65,8	67,6	69,0	64,5	66,7	74,0	66,0	69,0
Transferases	61,4	74,2	67,2	61,8	74,2	67,4	62,0	73,0	67,0
Hidrolases	75,5	74,5	75,0	74,4	73,9	74,1	77,0	76,0	76,0
Liases	66,7	56,7	61,3	65,4	56,7	60,7	62,0	60,0	61,0
Isomerases	78,0	68,4	72,9	78,8	68,4	72,9	76,0	70,0	73,0
Ligases	80,0	66,7	72,7	80,0	66,7	72,7	79,0	61,0	69,0
Média	71,4	69,9	69,5	70,0	69,5	69,1	71,0	68,0	70,0

Fonte: Dados da Pesquisa

A precisão média com o uso do Algoritmo Genético foi de 71,4%. Os maiores valores de precisão foram os obtidos na classificação das Ligases sendo 80% para o AG e PSO e uma precisão de 79% com a metodologia adotada por (LEIJOTO et al., 2014). Isso pode ser explicado pelo fato de entre as 15 instâncias classificadas como pertencentes às Ligases, 12 foram classificadas corretamente, na execução tanto do AG quanto do PSO.

Os valores de sensibilidade médios foram de 69,9%, 69,5% e 68,0% para o AG, PSO e a metodologia abordada por (LEIJOTO et al., 2014), respectivamente. A explicação para os baixos valores pode se dar pelas matrizes de confusão que mostram que das 60 instâncias das Liases, apenas 34 foram corretamente classificadas pelo AG e PSO, sendo esta, a classe que obteve o menor valor de sensibilidade pelo classificador, com uma taxa de apenas 56,7% para as execuções do AG e PSO respectivamente.

5 CONCLUSÃO E TRABALHOS FUTUROS

Através da metodologia adotada neste trabalho, onde foram implementados um algoritmo genético, um algoritmo de enxame de partículas e o *grid-search*, percebe-se que todos os algoritmos são aptos a serem utilizados para o ajuste dos parâmetros da máquina vetor de suporte de maneira satisfatória. Levando-se em conta a facilidade de implementação e o tempo médio de execução, recomenda-se a heurística *grid-search*.

Como proposta de trabalho futuro, propomos o uso de paralelismo junto aos algoritmos evolucionários implementados, visto que tal proposta pode contribuir no desempenho médio dos mesmos. No contexto da predição de função de proteínas, uma implementação conjunta de ambos AG e PSO, pode ser utilizada para a seleção de características das proteínas e para o ajuste de parâmetros da SVM. O uso do AG e PSO para o ajuste dos parâmetros de um outro classificador, como as redes neurais artificiais, por exemplo, também é abordado como uma proposta de trabalho futuro.

Referências

BEN-HUR, Asa; WESTON, Jason. A user's guide to support vector machines. In: **Data mining techniques for the life sciences**. [S.l.]: Springer, 2010. p. 223–239.

BERMAN, Helen M. et al. The protein data bank. **Nucleic Acids Res**, v. 28, p. 235–242, 2000.

BOSER, Bernhard E.; GUYON, Isabelle M.; VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. In: **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**. New York, NY, USA: ACM, 1992. (COLT '92), p. 144–152. ISBN 0-89791-497-X. Disponível em: <http://doi.acm.org/10.1145/130385.130401>.

CHANG, Chih-Chung; LIN, Chih-Jen. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, v. 2, p. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

CONFORTI, Domenico; GUIDO, Rosita. Kernel based support vector machine via semidefinite programming: Application to medical diagnosis. **Computers & Operations Research**, v. 37, n. 8, p. 1389 – 1394, 2010. ISSN 0305-0548. Operations Research and Data Mining in Biological Systems. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054809000562>.

COSTA, José Alfredo F; BITTENCOURT, Valnaide G; SOUTO, Marcílio CP de. **Aplicação de Multi-classificadores no Reconhecimento de Classes Estruturais de Proteínas**. [S.l.]: Press XXVIII CNMAC, 2005.

DOBSON, Paul D.; DOIG, Andrew J. Predicting enzyme class from protein structure without alignments. **Journal of Molecular Biology**, v. 345, n. 1, p. 187 – 199, 2005. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022283604013166>.

GUO, Guodong; LI, S.Z.; CHAN, Kap Luk. Face recognition by support vector machines. In: **Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on**. [S.l.: s.n.], 2000. p. 196–201.

HALL, Mark et al. The weka data mining software: an update. **ACM SIGKDD explorations newsletter**, ACM, v. 11, n. 1, p. 10–18, 2009.

HASSAN, Rania; COHANIM, Babak E.; WECK, Olivier L. de. Comparison of particle swarm optimization and the genetic algorithm. In: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS. **46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference**. Austin, Texas, 2005.

HERBRICH, Ralf. **Learning kernel classifiers**. [S.l.]: MIT Press, Cambridge, 2002.

HUANG, Cheng-Lung; WANG, Chieh-Jen. A ga-based feature selection and parameters optimizationfor support vector machines. **Expert Systems with applications**, Elsevier, v. 31, n. 2, p. 231–240, 2006.

İLHAN, İlhan; TEZEL, Gülay. A genetic algorithm–support vector machine method with parameter optimization for selecting the tag snps. **Journal of biomedical informatics**, Elsevier, v. 46, n. 2, p. 328–340, 2013.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Neural Networks, 1995. Proceedings., IEEE International Conference on**. [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. [S.l.: s.n.], 1995. v. 14, n. 2, p. 1137–1145.

LEIJOTO, Larissa Fernandes et al. A genetic algorithm for the selection of features used in the prediction of protein function. In: IEEE. **Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on**. [S.l.], 2014. p. 168–174.

LESSMANN, Stefan; STAHLBOCK, Robert; CRONE, Sven F. Genetic algorithms for support vector machine model selection. In: IEEE. **Neural Networks, 2006. IJCNN'06. International Joint Conference on**. [S.l.], 2006. p. 3063–3069.

LIN, Shih-Wei et al. Particle swarm optimization for parameter determination and feature selection of support vector machines. **Expert Systems with Applications**, v. 35, n. 4, p. 1817 – 1824, 2008. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417407003752>.

LINDEN, Ricardo. **Algoritmos Geneticos (2a edicao)**. [S.l.]: Brasport, 2008.

NELSON, D. L.; COX, M. M. Ènzimas. In: **Princípios de bioquímica de Lehninger**. [S.l.: s.n.], 2011.

NESHICH, Goran et al. Javaprotein dossier: a novel web-based data visualization tool for comprehensive analysis of protein structure. **Nucleic acids research**, Oxford Univ Press, v. 32, n. suppl 2, p. W595–W601, 2004.

REN, Yuan; BAI, Guangchen. Determination of optimal svm parameters by using ga/pso. **Journal of Computers**, v. 5, n. 8, p. 1160–1168, 2010.

RESENDE, Walkiria K et al. The use of support vector machine and genetic algorithms to predict protein function. In: IEEE. **Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on**. [S.l.], 2012. p. 1773–1778.

ROJAS, Sergio A; FERNANDEZ-REYES, Delmiro. Adapting multiple kernel parameters for support vector machines using genetic algorithms. In: IEEE. **Evolutionary Computation, 2005. The 2005 IEEE Congress on**. [S.l.], 2005. v. 1, p. 626–631.

SANGITAB, P.; DESHMUKH, S.R. Use of support vector machine, decision tree and naive bayesian techniques for wind speed classification. In: **Power and Energy Systems (ICPS), 2011 International Conference on**. [S.l.: s.n.], 2011. p. 1–8.

SANTOS, B. C.; ZARATE, L. H.; NOBRE, C. N. Algoritmo genético paralelo para predição de função de proteína. Submitido. 2015.

SOARES, Gustavo Luís. Algoritmos genéticos: estudo, novas técnicas e aplicações. **Belo Horizonte**, 1997.

SOUZA, Bruna Thabata Ribeiro de; FAGUNDES, Fabiano. **UTILIZAÇÃO DO ALGORITMO PARTICLE SWARM OPTIMIZATION PARA RESOLVER O PROBLEMA DE TIMETABLING NA ELABORAÇÃO DA GRADE DE HORÁRIOS EM UM CURSO SUPERIOR**. 2013. Dissertação (Mestrado) — Centro Universitário Luterano de Palmas, Palmas.

VAPNIK, V; LERNER, A. Pattern recognition using generalized portrait method. **Automation and Remote Control**, v. 24, 1963.