

Poda em Grafo de Classificador OPF Utilizando Algoritmo de Otimização por Colônia de Formigas

Juliana Patrícia Detroz

Programa de Pós-Graduação em Computação Aplicada
Universidade do Estado de Santa Catarina (UDESC)
Joinville - SC, Brasil
julidetroz@gmail.com

Rafael Stubs Parpinelli

Programa de Pós-Graduação em Computação Aplicada
Universidade do Estado de Santa Catarina (UDESC)
Joinville - SC, Brasil
rafael.parpinelli@udesc.br

Resumo—OPF (*Optimum-path Forest*) é um método de classificação que reduz a tarefa de reconhecimento de padrões a um problema combinatorial. Porém, o tempo de classificação deste cresce proporcionalmente ao tamanho da base de treinamento. Uma tratativa consiste na identificação de amostras de pouca relevância, a fim de reduzir o conjunto de treinamento. Desta forma, este artigo tem como objetivo propor um método de poda de grafo classificador OPF através de aplicação baseada na metaheurística *Ant Colony Optimization*. Foram realizados dois experimentos variando a influência do valor heurístico e da trilha de feromônio. Os resultados mostraram que a aplicação proposta atende ao objetivo, encontrando soluções candidatas que apresentam redução de aproximadamente 60% do número de nós em relação ao grafo original, reduzindo o tempo de classificação médio em aproximadamente 75% e alcançando taxa de acerto média de 86,5%.

Palavras-chave—Poda OPF; Floresta de Caminhos Ótimos; Otimização por Colônia de Formigas; Reconhecimento de padrões.

I. INTRODUÇÃO

A crescente quantidade de informação disponível em formato digital, como imagens e vídeos, tem promovido um maior interesse por soluções de reconhecimento de padrões que automatizem a análise destes dados. Neste contexto, o classificador por Floresta de Caminhos Ótimos (OPF, do inglês, *Optimum-path Forest*) é um método que pode ser utilizado em aplicações deste fim. Porém, o seu tempo de classificação cresce proporcionalmente ao tamanho da base de treinamento, uma vez que cada amostra desta corresponde a um nó do grafo que representa o classificador [1].

Visto isto, sugere-se que uma aplicação de reconhecimento de padrões se beneficiaria da adoção de medidas de poda no grafo do classificador OPF, com o objetivo de reduzir o tempo computacional necessário para a classificação.

Motivado por estas questões, o presente artigo propõe a aplicação de um algoritmo bioinspirado para a execução de poda em grafo do classificador OPF. Optou-se por utilizar a metaheurística de otimização por colônia de formigas, devido ao problema de poda em grafo ser do tipo combinatorial e de complexidade não polinomial.

Visto isto, esta pesquisa tem como objetivo verificar a hipótese de que o classificador resultante da aplicação do método de poda possa alcançar um resultado equivalente ou

aproximado ao do classificador original, sem poda, reduzindo o tempo de execução da etapa de classificação.

Como estudo de caso, a solução proposta foi aplicada a uma base de 774 imagens de hidrômetros e não hidrômetros. Foram realizados dois experimentos, variando-se os parâmetros de influência da trilha de feromônio e do valor heurístico, a fim de avaliar o impacto destas características no método de poda. O resultado de ambos experimentos foi comparado ao obtido pelo grafo do classificador original. O método proposto apresentou taxa de acerto média de 86,5% e 82,63% para os experimentos A e B, respectivamente, e redução superior a 60% do tamanho do grafo em relação ao classificador original.

O restante do artigo encontra-se estruturado da seguinte forma: a seção II detalha os principais conceitos relacionados ao método proposto e trabalhos correlatos ao escopo desta pesquisa. A metodologia e a solução proposta são expostas na seção III, sendo os resultados discutidos na seção seguinte. Por fim, são apresentadas as conclusões e referências.

II. REVISÃO DA LITERATURA

A seguir são detalhados os principais conceitos e definições relacionados ao escopo desta pesquisa.

A. Classificação por Floresta de Caminhos Ótimos

OPF é um método de classificação baseado em grafos, que reduz o problema de reconhecimento de padrões à computação de uma floresta de caminhos ótimos. Desta forma, esta tarefa é tratada como um problema combinatorial, no qual cada classe é representada por uma ou mais árvores, que têm como uma amostra principal, denominada protótipo. Segundo Papa et al. [1], cada nó do grafo representa uma amostra do conjunto de treinamento (z_1) e pertence à árvore do protótipo ao qual encontra-se mais fortemente conectado.

Para cada nó tem-se um vetor de características ($v(s)$), através do qual é possível determinar a dissimilaridade ($d(s, t)$) entre dois nós s e t , calculada através de uma função de distância. Por padrão, utiliza-se a distância Euclidiana, porém o método permite o uso de outras métricas.

Cada nó é conectado aos demais através de arestas definidas por uma relação de adjacência e valoradas através da função de distância. A etapa de treinamento do classificador inicia-se com um grafo completo. A partir dos nós protótipos, são

calculadas as árvores de caminhos mínimos. Usualmente, estes protótipos correspondem aos nós localizados nos limites entre duas classes distintas, sendo que, quando encontrada uma aresta que o conecte a um nó de outra classe, esta é eliminada. Desta forma, cada caminho do grafo possui um valor atribuído através de uma função de conectividade, assumindo o valor da maior aresta contida no trajeto entre um nó e o protótipo da árvore ao qual pertence. A Figura 1 exemplifica o processo de treinamento e classificação do OPF, desde a etapa inicial, representada pelo grafo completo (a), as árvores de caminhos mínimos (b) definidas na etapa de treinamento e o processo de classificação de uma nova amostra (c,d). A classificação

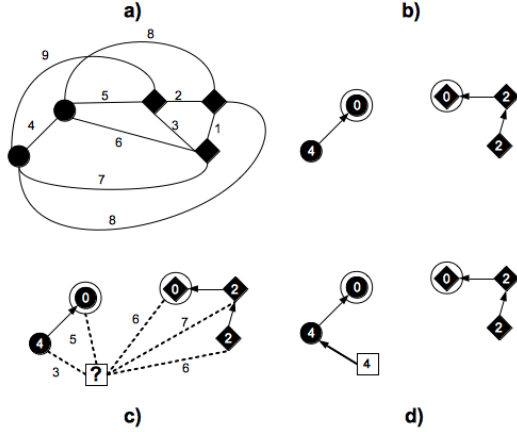


Figura 1. Funcionamento do classificador OPF. Fonte: Adaptado de Papa et al. [1]

de uma nova amostra t é realizada através da avaliação dos caminhos ótimos que conectem os protótipos a t , como se t pertencesse ao grafo de treinamento. Assim, de forma simplificada, pode-se dizer que os protótipos competem para verificar a qual deles a amostra encontra-se mais fortemente conectada. Para isto, utiliza-se uma lista dos nós, ordenada pelo custo de seus caminhos, avaliando para cada um destes o custo ótimo de $C(t)$ [1]:

$$C(t) = \min\{\max\{C(s), d(s, t)\}\}, \forall s \in Z. \quad (1)$$

A nova amostra t será conectada ao nó s do caminho que obter menor $C(t)$, sendo s definido como predecessor de t e atribuído a t o rótulo do protótipo da árvore a qual s pertence.

B. Otimização por colônia de formigas

Proposto por Dorigo et al. [2], o algoritmo de otimização por colônia de formigas (ACO, do inglês, *Ant Colony Optimization*) é um método meta-heurístico inspirado no comportamento de formigas na busca por alimento. Estes insetos cooperam de forma indireta, através do depósito da substância denominada feromônio. Desta forma, aquelas que encontram um menor caminho entre o formigueiro e a fonte de alimento percorrem este trajeto mais rápido e com maior frequência. Consequentemente, despejam maior quantidade de feromônio e atraem outras formigas, que tendem a seguir o caminho com maior quantidade desta substância, reforçando-o.

Baseado neste mecanismo, o método ACO consiste de um conjunto de agentes simples que agem de forma descentralizada, alcançando um comportamento emergente. É aplicado usualmente a problemas de otimização. O algoritmo original, denominado *Ant System*, foi proposto inicialmente a problemas combinatoriais [3]. Neste tipo de abordagem, a cada iteração, cada agente (k) percorre um caminho de forma estocástica, guiado por uma regra probabilística de transição, construindo uma possível solução.

Segundo Dorigo et al. [3], um valor da concentração de feromônio (τ) é associado a cada componente das potenciais soluções. Para problemas combinatoriais, representados por grafos, estes componentes podem ser tanto os nós quanto as arestas. Desta forma, os agentes comunicam-se indiretamente através das mudanças ocorridas no ambiente, ou seja, o depósito ou evaporação de feromônio. A quantidade depositada depende da qualidade da solução encontrada e é utilizada pelos demais agentes, guiando-os em direção a regiões mais promissoras do espaço de busca.

Além da trilha de feromônio, cada componente possui um valor de informação heurística (η) associado, o qual mede e representa a preferência dos agentes de mover-se de um componente(i) a outro(j), isto é, a sua atratividade [4]. A regra de transição é formada por uma composição da função heurística e da quantidade de feromônio, conforme a fórmula abaixo [3], sendo esta aplicada a cada componente na construção das soluções candidatas:

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \times \eta_{ij}^\beta}{\sum_{c_{il} \in N(s^P)} \tau_{il}^\alpha \times \eta_{il}^\beta} \text{ se } c_{ij} \in N(s^P), \quad (2)$$

$$0, \text{ caso contrário}$$

$N(s^P)$ é o conjunto de componentes possíveis, acessíveis por i e l é um componente ainda não visitado por k . A definição dos parâmetros α e β é um fator importante no equilíbrio entre diversificação e intensificação, determinando, respectivamente, a relativa influência da trilha de feromônio e da função heurística. Dorigo e Stützle [5] afirmam que quanto maior o valor de α , mais forte é a intensificação, ou seja, o algoritmo tende a aprimorar soluções já avaliadas, com maior quantidade de feromônio. Nas iterações iniciais, quando ainda há pouca incidência desta substância (baixo valor de τ), a diversificação é maior, havendo maior exploração do espaço de busca. De forma semelhante, β representa a influência da função heurística. A forma de atualização da trilha de feromônio é uma das características que diferencia as diversas propostas de algoritmos ACO. Conforme Dorigo et al. [3], existem 3 abordagens principais: *offline*, MAX-MIN e local. Na primeira, utilizada pelo *Ant System*, a atualização é realizada ao final de cada iteração, sendo que as potenciais soluções encontradas por todas as formigas são consideradas. O valor de feromônio de cada componente das soluções é calculado por:

$$\tau_{ij} \leftarrow (1 - \rho) \times \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (3)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (4)$$

sendo $\Delta\tau_{ij}^k = \frac{Q}{L_k}$ se a formiga k usa a aresta (i, j) em sua rota,

onde, ρ corresponde à taxa de evaporação e sendo $\Delta\tau_{ij}^k$ a quantidade de feromônio despejado em cada componente (i, j) por cada formiga k . Esta quantidade é proporcional a distância do caminho percorrido (L_k), ou seja, indica a qualidade da solução encontrada por uma formiga, já que um menor caminho L_k despejará uma maior quantidade de feromônio em seus componentes ($\frac{Q}{L_k}$). Q é uma constante.

Semelhante à atualização *offline*, o modelo MAX-MIN *Ant System* utiliza-se das mesmas fórmulas, porém somente o agente que encontrou a melhor solução atualiza a trilha e com limitada quantidade da substância. Esta é uma abordagem de estratégia elitista. Por último, a atualização local, como no modelo *Ant Colony System*, ocorre a cada passo da construção das soluções, decrementando o valor de feromônio de caminhos percorridos e incentivando os agentes a seguir por outras regiões do espaço, diversificando a busca.

C. Trabalhos relacionados

Diversos estudos foram encontrados na literatura relacionados a propostas para realização de poda em grafo ou estruturas de dados semelhantes, sendo em alguns casos utilizados métodos evolutivos ou inspirados em abordagens de enxames.

Chen et al. [6] propõem a aplicação de algoritmos genéticos (AG) para executar a poda em árvores de decisão. Nesta pesquisa foi utilizada uma representação binária (*string* de *bits*). Cada gene representa o peso de uma aresta da árvore, onde 0 indica a existência da mesma após a poda e 1 representando a remoção da aresta. Para a avaliação, a função de *fitness* considera a taxa de erros da árvore de decisão resultante de cada indivíduo. Os autores (*op. cit.*) afirmam que os experimentos demonstraram que o método proposto apresenta resultados melhores ou equivalentes a outras técnicas de poda aplicadas a árvores de decisão.

Nakamura et al. [7] propõem a utilização de um método baseado no algoritmo evolutivo *Harmony search* (HS) para encontrar o valor ótimo aproximado da taxa de perda de acurácia (*MLoss*) de um classificador OPF, quando este é submetido a um processo de poda. Até então, *MLoss* era um valor parametrizável. A metaheurística HS é um algoritmo de otimização inspirado na metáfora do processo de improvisação adotado por músicos. Segundo os autores, os experimentos sugerem que a técnica proposta obtém resultados satisfatórios. Apesar de também fazer uso de uma metaheurística estocástica e ser aplicado ao mesmo método de classificação (OPF), diferente do estudo de Nakamura et al. [7] (que aplica HS no ajuste de parâmetros de um método de poda determinístico), na presente pesquisa a abordagem populacional é adotada diretamente na realização da poda do grafo classificador.

Na pesquisa de Ramos et al. [8], é apresentado um método de poda de grafo do classificador OPF. A proposta consiste da identificação e eliminação dos nós irrelevantes, ou seja, aqueles que não pertencem a nenhum caminho ótimo utilizado na classificação das amostras de testes, fazendo uso de uma abordagem determinística. Os autores relatam que a técnica

utilizada obteve uma redução de cerca de 50% no tamanho do conjunto de treinamento, com pequeno impacto na acurácia do classificador e reduzindo o tempo de classificação. Isto reforça a importância de um método de poda para este classificador. Observa-se que em um base de dados com grande variabilidade de características, e de acordo com a estratégia de divisão desta nos conjuntos de treinamento e teste, esta abordagem pode eliminar amostras importantes do conjunto de treinamento, caso estas não possuam representantes semelhantes no conjunto de testes. Uma abordagem probabilística, como a proposta pela presente pesquisa, permite a estas amostras a chance de serem mantidas no classificador, podendo impactar positivamente na taxa de acerto.

III. METODOLOGIA

Para esta pesquisa, implementou-se o método de otimização por colônia de formigas baseado no modelo canônico *Ant System*. O algoritmo foi aplicado para realização de poda no grafo do classificador OPF, com o intuito de remover nós irrelevantes para a classificação e, conseqüentemente, possibilitar um menor tempo de execução. Como mostra a Figura 2, um mesmo grafo inicial possibilita a existência de n subgrafos distintos resultantes de um processo de poda. O objetivo é encontrar o menor grafo que mantenha uma taxa de acerto equivalente ou próxima a do grafo original. O grafo contido no canto superior esquerdo representa o classificador inicial, possuindo amostras de duas classes, das quais os objetos destacados representam os protótipos. Nele é possível observar a presença de nós muito próximos, o que poderia indicar que possuem características muito semelhantes e apenas um deles já seria suficiente para classificar corretamente uma imagem encontrada na mesma região do espaço de características. O grafo resultante da poda apresentado no canto inferior

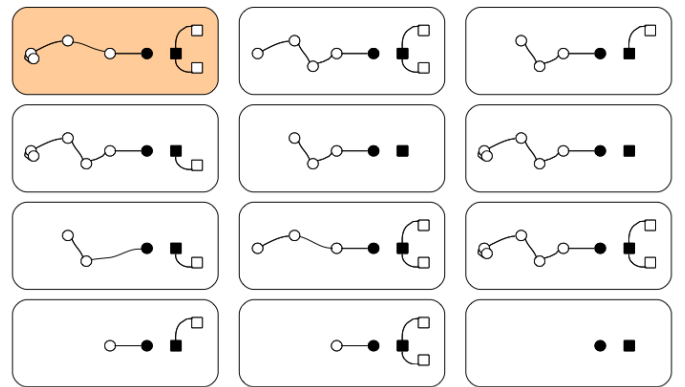


Figura 2. Possibilidades distintas de poda a partir de um grafo OPF

direito é formado apenas pelos protótipos. Para um problema linearmente decidível, existe a possibilidade desta ser uma solução aceitável. Porém, para a maior parte dos problemas de classificação, isto não é factível e o mesmo apresentaria uma taxa de acerto insatisfatória. Uma busca por todas as possibilidades de combinação de nós também não seria uma abordagem plausível, visto que um grande número de nós levaria a uma

grande quantidade de possíveis combinações, inviabilizando esta estratégia. A aplicação da metaheurística ACO apresenta-se como uma alternativa para obter uma solução satisfatória em tempo computacional aceitável. Os experimentos foram realizados utilizando uma base de 774 imagens de hidrômetros e não hidrômetros, isto é, duas classes distintas.

A extração das características de cada amostra foi obtida através da aplicação do descritor de forma Histogramas de Gradientes Orientados (HOG, do inglês, *Histograms of Oriented Gradients*)[9], resultando em um vetor de dimensão 4680.

Para a presente pesquisa, utilizou-se validação cruzada 5-fold, desta forma, no início de cada execução a base de imagens é dividida aleatoriamente em cinco partições. Destas, quatro são utilizadas para o treinamento, sendo duas para formar os subgrafos do classificador ($z1$) e duas para obter-se a informação heurística e a taxa de acerto de treinamento ($z2$). A última partição corresponde ao conjunto de teste ($z3$), que será utilizado para estimar a taxa de acerto de teste. Como resultado, os conjuntos são representados, respectivamente, na proporção de 40% (treino), 40% (validação) e 20% (teste) da base completa, mantendo também esta proporção para as classes. Adotou-se esta proporção com o intuito de conter uma maior quantidade de nós no grafo, porém possibilitando uma maior utilização das amostras de treinamento na etapa de classificação, já que a densidade é proporcional ao número de imagens que serão classificadas, ou seja, de $z2$.

Durante uma execução, o algoritmo ACO é processado cinco vezes, para que cada uma partições seja utilizada como conjunto de teste ($z3$). Desta forma, a cada execução obtém-se uma configuração diferente, porém durante esta todos agentes são aplicados à mesma distribuição de dados, formando “subgrafos” a partir do mesmo conjunto $z1$, sendo avaliados a partir da mesma base $z2$ e testados em $z3$. O Algoritmo 1 apresenta o fluxo de execução da metaheurística ACO aplicada ao problema desta pesquisa.

Algoritmo 1 Metaheurística ACO aplicada à poda de grafo OPF

```

1: procedure PODAOPF( $kFold$ ,  $nrMaxIteracao$ ,  $nrMaxAgentes$ )
2:    $inicializaParametros(txEvapora, Q)$ ;
3:    $trilhaF = inicializaTrilhaFeromonio()$ ;
4:    $divideBase(kFold)$ ;
5:   for  $p \leftarrow 1, kFold$  do
6:      $(z1, z2, z3) = defineZ1Z2Z3(2, 2, p)$ ;
7:      $cOrig = treinaOPF(z1)$ ;
8:      $txAcertoOrig = classificaOPF(z2)$ ;
9:      $hashVlHeur = funcaoHeuristica(cOrig, z2)$ ;
10:    for  $i \leftarrow 1, nrMaxIteracao$  do
11:      for  $k \leftarrow 1, nrMaxAgentes$  do
12:         $cCand = aplicaPoda(cOrig, k, trilhaF, hashVlHeur)$ ;
13:         $txAcertoZ2 = classificaOPF(cCand, z2)$ ;
14:      end for
15:       $trilhaF = atualizaTrilhaFeromonio()$ ;
16:    end for
17:     $cResult = retornaClassificadorPoda(trilhaF)$ ;
18:     $txAcertoZ3(cResult, z3)$ ;
19:  end for
20:  return  $cResult$ ;
21: end procedure

```

No contexto da presente pesquisa, cada solução construída

pelos agentes é um grafo, com número de nós variável, resultante da poda guiada pela regra de transição. Como esta possui comportamento estocástico, apesar de ser influenciada pelos valores de feromônio e heurística, possivelmente cada agente encontrará um grafo distinto. Cada componente da solução é representado pelos nós do grafo que formam o classificador OPF, resultante da etapa de treinamento. Para realizar a poda, procuram-se os nós mais relevantes para a classificação das imagens, ou seja, aqueles utilizados com maior frequência (densidade) para classificar corretamente as imagens do conjunto $z2$, e que foram definidos como predecessores. Desta maneira, estes nós serão os componentes visitados pelas formigas com mais frequência, tendendo a possuir maiores taxas de feromônio. A partir desta premissa, definiu-se como função heurística:

$$\eta_i = \text{densidade de relevância do nó,}$$

isto é, a quantidade de amostras de $z2$ classificadas corretamente com base no nó i e que tem, portanto, este nó como predecessor. O valor heurístico é diretamente proporcional à densidade de relevância.

Para atualização da trilha de feromônio, optou-se pelo método *offline*. Desta maneira aplicou-se a fórmula 3, conforme descrito na seção II. Para o contexto desta pesquisa, $\Delta\tau_i^k$ representa o valor da constante Q dividido pela quantidade de nós do classificador após a poda encontrado por cada uma das k formigas que incluíram o nó i . Atribuiu-se empiricamente os valores 100 e 0, 1 para a constante Q e para a taxa de evaporação ρ , respectivamente.

Na construção de cada solução, a regra de transição é aplicada de forma que os nós com maior concentração de feromônio e mais atrativos (valor da heurística) tenham maior probabilidade de serem escolhidos para o grafo formado por cada formiga. Porém, como o método é estocástico, os nós com menor relevância também têm a possibilidade de serem escolhidos, mesmo que com menor chance. Para isto, aplicou-se a seguinte fórmula:

$$p_i^k = \frac{((\tau_i)^\alpha \times (\eta_i)^\beta)}{((\tau_{max})^\alpha \times (\eta_{max})^\beta)} \quad (6)$$

Os valores de τ_i e η_i de cada nó foram divididos pela multiplicação da maior concentração de feromônio e maior valor de heurística encontrados. O resultado da fórmula foi multiplicado por 100 para obter um valor entre 0 e 100, a fim de possibilitar a aplicação do elemento estocástico. Para isto, foi gerado um número aleatório dentro deste intervalo, se o número encontrado for menor do que p_i^k , indica que está dentro da probabilidade do nó a ser selecionado, portanto o mesmo não é podado (é relevante). Caso contrário, o nó é eliminado.

Após aplicada a regra de transição a todos os nós, executa-se novamente a etapa de treinamento com o conjunto resultante e classificam-se as amostras de $z2$ a partir do grafo encontrado por cada um dos agentes. Com isto, tem-se os nós que foram utilizados para classificar corretamente as amostras de $z2$ e a taxa de acerto de treinamento. Após a execução para todos

os agentes, atualiza-se a trilha de feromônio e repete-se o processo para a próxima iteração. Para cada execução foram consideradas 30 iterações e 20 agentes.

Como critério de parada adotou-se o número máximo de iterações (30), porém outros fatores poderiam ser utilizados como, por exemplo, um *threshold* indicando o número mínimo de nós que compõem o grafo, a taxa de acerto mínima esperada, quando não há mais redução no número de nós em iterações sucessivas, ou uma combinação do número de nós e taxa de acerto.

IV. RESULTADOS

Para avaliar o método proposto, foram comparados os resultados obtidos pelo algoritmo implementado alterando-se os os parâmetros α e β , a fim de identificar a influência dos mesmos na poda. O primeiro experimento (A) considerou $\alpha = 1$ e $\beta = 2$, privilegiando o valor da informação heurística em relação à concentração de feromônio. No experimento (B), inverteu-se a situação, considerando $\alpha = 2$ e $\beta = 1$. Neste caso, o valor da concentração de feromônio apresenta um maior impacto na aplicação da regra de transição. Para cada configuração foram consideradas 150 execuções (30 repetições \times validação cruzada *5-fold*). Os dados apresentados a seguir correspondem à média das 150 execuções de cada experimento. A iteração *zero*(0) representa o valor encontrado para o grafo do classificador original, antes da aplicação da poda.

A Tabela I apresenta os resultados obtidos nos experimentos A e B. Os dados foram analisados considerando quatro métricas: o número de nós podados, a taxa de acerto de treinamento do classificador podado, o tempo para classificação de todas amostras contidas em $z2$ e a taxa de acerto de teste do classificador final (iteração 29), isto é, em relação ao conjunto $z3$. Para a base em questão, o grafo do classificador original possui 309 nós e tempo médio de classificação de 0,4s. Observa-se que ambos experimentos removeram mais de 60% dos nós, porém o classificador resultante do experimento A mantém maior taxa de acerto. As taxas de acerto em $z3$ dos experimentos A e B encontram-se no intervalo de confiança de $86,5 \pm 0,3669$ e $82,63 \pm 0,5442$, respectivamente, com confiança de 95%.

Tabela I
COMPARAÇÃO DE RESULTADOS DA APLICAÇÃO DE PODA

Experimento	Qtde de nós removidos	Tempo de classificação (s)	Taxa de acerto treinamento (%)	Taxa de acerto teste (%)
A	210,0 \pm 16,550	0,148 \pm 0,027	87,251 \pm 1,244	86,500 \pm 1,0255
B	256,8 \pm 20,060	0,082 \pm 0,026	83,146 \pm 2,304	82,630 \pm 1,521

A Figura 3 mostra a quantidade média de nós eliminados por iteração para a configuração A e B obtidos no treinamento ($z2$). Em ambos é possível observar uma maior convergência nas iterações iniciais, ocorrendo uma intensificação em seguida, até atingir a estabilização.

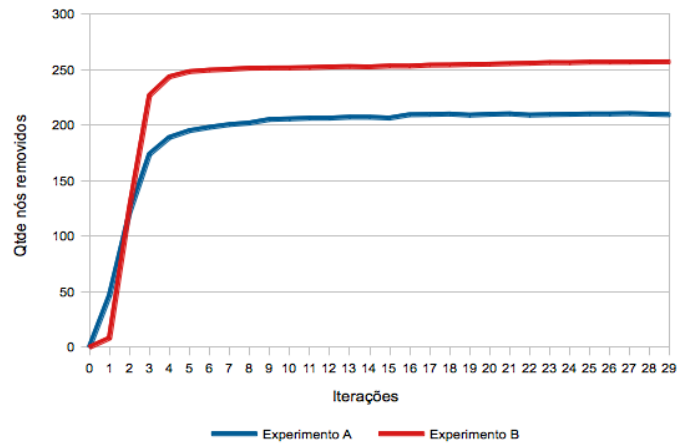


Figura 3. Gráfico de convergência de poda.

Nota-se que alterando apenas a influência da trilha de feromônio e função heurística na regra de transição, o algoritmo mostra um comportamento diferente. No experimento B, que prioriza α , ou seja, a trilha de feromônio, o método apresentou uma redução maior no número de nós já nas primeiras iterações em comparação ao comportamento observado no experimento A. Isto explica-se pelo fato da qualidade da solução encontrada ter sido representada pela quantidade de nós do classificador resultante. Logo, o experimento que prioriza a trilha de feromônio apresenta poda mais acentuada. Já na terceira iteração, o número médio de nós removidos foi de 226, enquanto que no experimento A, nesta mesma iteração o número médio de nós eliminados foi de 173. Utilizando-se valores de α e β menores, a convergência tende a ser mais lenta, privilegiando a manutenção da diversidade nas iterações iniciais.

Obseva-se ainda que no experimento B a redução de nós acentua-se na terceira iteração. Visto que a trilha de feromônio é reforçada a cada iteração, o resultado sugere que nas iterações anteriores esta ainda não apresenta valor significativo, fazendo com que todos os nós tenham praticamente a mesma probabilidade de serem removidos.

Como mostra a Figura 4, a taxa de acerto média de treinamento a cada iteração manteve-se relativamente estável para o experimento A, mesmo após a redução de aproximadamente 66% dos nós. É possível notar um leve incremento na taxa de acerto de treinamento.

A maior eliminação de nós observada no experimento B refletiu na qualidade dos classificadores podados. A taxa de acerto média de treinamento dos classificadores após a poda caiu em aproximadamente 3% em relação ao grafo original, comportamento que não ocorreu para o experimento A. Ao priorizar β , isto é, a função heurística, no experimento A houve uma maior retenção dos nós com maior relevância, o que reflete-se positivamente na taxa de acerto.

Em relação ao tempo para classificação das amostras de $z2$, a redução do grafo mostrou-se como uma alternativa satisfatória. A Figura 5 apresenta o gráfico de tempo de

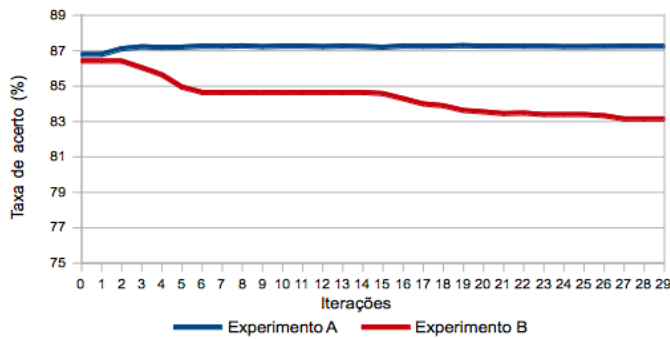


Figura 4. Gráfico de taxa de acerto x iteração.

classificação médio por iteração que os classificadores construídos pelos agentes levaram para rotular todas as imagens de z_2 , ou seja, durante o treinamento. É possível observar uma redução a medida que o número de iterações aumenta e, conforme a Figura 3, a medida que o número de nós diminui.

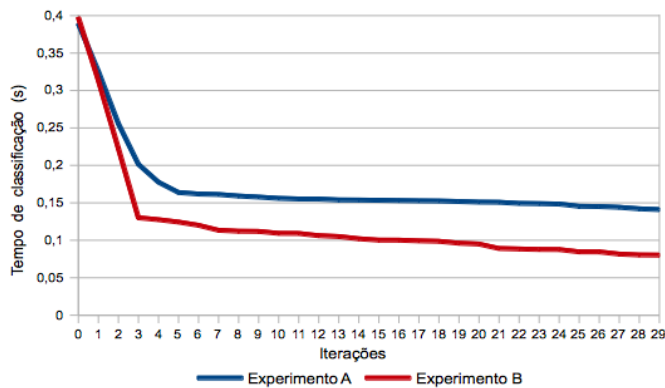


Figura 5. Tempo de Classificação.

Como a maximização da taxa de acerto e minimização do número de nós do classificador são objetivos conflitantes, é provável que não exista uma única solução ótima que atenda a ambos. Desta forma, pode-se comparar os resultados dos dois experimentos através do gráfico de Pareto[10], apresentado na Figura 6. Cada ponto representa a taxa de acerto média de treinamento e a quantidade média de nós do classificador obtido na última iteração de A e B.

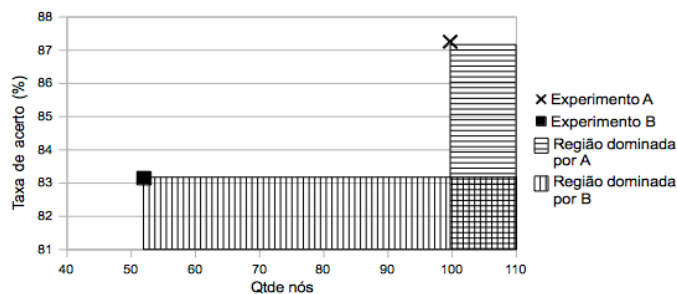


Figura 6. Gráfico de Pareto - Taxa de acerto x qtde nós.

Neste é possível observar o conjunto Pareto não-dominado e, de acordo com a importância dada a cada um dos objetivos, escolher a solução que melhor atenda à aplicação. Por exemplo, nota-se que é possível alcançar uma taxa de acerto de cerca de 87,25% com aproximadamente 99 nós. Já para uma maior redução do número de nós obtém-se menor taxa de acerto, como no resultado obtido em B, onde classificadores com 52 nós apresentaram taxa de acerto média de 83,14%.

V. CONCLUSÃO

Com a aplicação do algoritmo proposto foi possível observar que o método ACO é eficiente para a realização de um mecanismo de poda em grafo classificador OPF, podendo este ser adaptado a grafos diversos. Os resultados obtidos mostram que, para a base de imagens utilizada, a aplicação da poda resultou em um classificador menos complexo, eliminando as amostras irrelevantes, sem apresentar impacto significativo na taxa de acerto e possibilitando a classificação das imagens em menor tempo. Desta forma, corrobora-se a hipótese inicial de que com um método de poda pode-se encontrar um classificador OPF de grafo reduzido que alcance resultado equivalente ou aproximado ao do classificador original, diminuindo o tempo de classificação.

Como trabalhos futuros, sugere-se a implementação de outros métodos de atualização da trilha de feromônio, incluindo assim um mecanismo de diversificação. Além disto, é relevante comparar os resultados obtidos a outros métodos de poda, além de aplicar o método proposto a outros conjuntos de dados, para ampliar o escopo de validação.

REFERÊNCIAS

- [1] J. P. Papa, A. X. Falcao, and C. T. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.
- [2] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, 1996.
- [3] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *Computational Intelligence Magazine, IEEE*, vol. 1, no. 4, pp. 28–39, 2006.
- [4] O. C. García, F. H. Triguero, and T. Stützle, "A review on the ant colony optimization metaheuristic: basis, models and new trends," *Mathware & soft computing*, vol. 9, no. 3, pp. 141–175, 2002.
- [5] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
- [6] J. Chen, X. Wang, and J. Zhai, "Pruning decision tree using genetic algorithms," in *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, vol. 3. IEEE, 2009, pp. 244–248.
- [7] R. Y. M. Nakamura, C. R. Pereira, J. P. Papa, and A. X. Falcao, "Optimum-path forest pruning parameter estimation through harmony search," in *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIB-GRAPI Conference on*. IEEE, 2011, pp. 181–188.
- [8] C. C. O. Ramos, A. N. De Sousa, J. P. Papa, and A. X. Falcão, "A new approach for nontechnical losses detection based on optimum-path forest," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 181–189, 2011.
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
- [10] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.