

# AValiação Empírica de uma Estratégia de Computação Evolutiva Aplicada a um Problema de Otimização Combinatória Multimodal

LEONARDO R. EMMENDORFER\*, AURORA T. R. POZO†

\*Programa de Pós-Graduação em Métodos Numéricos em Engenharia  
Universidade Federal do Paraná  
Curitiba, PR, Brasil

†Departamento de Informática  
Universidade Federal do Paraná  
Curitiba, PR, Brasil

Emails: leonardo@inf.ufpr.br, aurora@inf.ufpr.br

**Abstract**— Classical evolutionary computation algorithms are not effective when solving globally multimodal optimization problems as convergence to a single optimum occurs. Later approaches often use some mechanism for maintaining population diversity as an auxiliary tool in the process. Those algorithms usually trust on sophisticated probabilistic models for capturing the problem structure. This paper presents an empirical evaluation of  $\varphi$ -PBIL, which is based on clustering as a diversity maintaining mechanism, extracting relevant information about the problem structure directly from the clustering models. A new parameter set is obtained and a new empirical comparison between  $\varphi$ -PBIL and a competitor is carried out. It shows that  $\varphi$ -PBIL is more effective and substantially more efficient than a state-of-the-art algorithm for several instances of a benchmark combinatorial optimization problem.

**Keywords**— Evolutionary computation, optimization, clustering

**Resumo**— Algoritmos clássicos de computação evolutiva não são efetivos em resolver problemas de otimização globalmente multimodais, devido a convergência para apenas um dos ótimos. Abordagens mais recentes recorrem a algum mecanismo de manutenção da diversidade da população, como sendo uma ferramenta auxiliar do processo, confiando em modelos probabilísticos sofisticados para a identificação da estrutura do problema. Este artigo apresenta um novo estudo empírico do  $\varphi$ -PBIL, que é um algoritmo baseado em agrupamentos para manutenção da diversidade, extraindo informações relevantes sobre a estrutura do problema a partir dos próprios modelos de agrupamento. Um novo conjunto de parâmetros é obtido aqui e uma nova comparação empírica entre o  $\varphi$ -PBIL e um concorrente é efetuada, onde fica evidente que, para diversas instâncias de um problema de otimização combinatória considerado *benchmark*, o  $\varphi$ -PBIL é mais efetivo e substancialmente mais eficiente que um algoritmo referência, considerado representante do estado da arte.

**Palavras-chave**— Computação evolutiva, otimização, agrupamento.

## 1 Introdução

A otimização combinatória abrange classes de problemas que envolvem variáveis inteiras e constitui atualmente uma área de pesquisa muito ativa. Uma das possíveis abordagens para a resolução destes problemas é através de técnicas de Computação Evolutiva (CE), como os algoritmos genéticos, os quais se caracterizam pela simplicidade de sua especificação e implementação. Um algoritmo de CE resolve um problema de otimização através da evolução de uma população de soluções para o problema em questão, até que ocorra a convergência, quando se espera que os ótimos globais tenham sido atingidos. Duas etapas estão frequentemente presentes: seleção de indivíduos promissores e recombinação destes indivíduos, de modo a obter uma nova população. A seleção dos indivíduos dá-se após avaliar seus respectivos valores de *fitness*, que é uma medida relacionada à função objetivo do problema.

Problemas globalmente multimodais são aqueles em que ocorre mais de um ótimo global. Um exemplo é o problema do particionamento de grafos, onde duas situações podem ocorrer, ocasionando a existência de diversas soluções ótimas: primeiro, regularidades na própria estrutura do grafo em questão. Segundo, e

mais importante, a codificação do problema em variáveis binárias leva sempre a, pelo menos, duas soluções ótimas já que a rotulagem de cada partição com 0s ou com um 1s é indiferente.

A existência de diversas soluções ótimas constitui um problema para os algoritmos genéticos, que são construídos de modo a ocorrer a convergência da população para apenas um ótimo global. Existe, nestes casos, a necessidade de aplicar algum mecanismo para garantir a manutenção estável de um certo grau de diversidade na população. A utilização de algoritmos de agrupamento vem sendo avaliada como uma alternativa possível para esta manutenção de diversidade. Por exemplo, em (Pelikan and Goldberg, 2000) o algoritmo k-médias (McQueen, 1967) é aplicado, agrupando indivíduos genotipicamente semelhantes. No trabalho aqui apresentado é avaliada uma abordagem similar: o algoritmo  $\varphi$ -PBIL (Emmendorfer and Pozo, 2007) – aprendizado incremental baseado em população e guiado por conceitos. A principal diferença em relação a outros trabalhos é que a combinação entre subpopulações (ou agrupamentos) diferentes é incentivada no  $\varphi$ -PBIL, seguindo uma hipótese de que esta combinação pode promover um grau desejável de exploração do espaço de busca. Sabe-se que esta combinação entre indivíduos diferentes pode ser

benéfica, e este assunto vem sendo discutido pelo menos desde Holland (1975). A forma mais conhecida de recombinação da população é o operador de cruzamento, presente no algoritmo genético original (sGA), o qual é bastante limitado em relação à preservação da estrutura do problema.

O objetivo deste trabalho é realizar uma nova avaliação empírica do algoritmo  $\varphi$ -PBIL, que foi proposto em (Emmendorfer and Pozo, 2007), buscando e utilizando um conjunto de parâmetros que se mostre adequado para a classe dos problemas multimodais considerados aqui. Através de um conjunto de experimentos realizados com o problema de particionamento de grafos, o qual é considerado *benchmark*, verifica-se que o algoritmo  $\varphi$ -PBIL é bastante efetivo e eficiente, resolvendo satisfatoriamente diversas instâncias deste problema globalmente multimodal.

O texto está organizado da seguinte forma: a seção 2 discute algumas preocupações importantes em computação evolutiva, particularmente em relação à solução de problemas multimodais. A seção 3 revisa o algoritmo  $\varphi$ -PBIL, o qual aborda soluções para estas preocupações através de um mecanismo centrado no agrupamento da população. Esta seção inclui uma investigação empírica sobre conjuntos de valores alternativos para os parâmetros, além dos adotados em Emmendorfer and Pozo (2007). Na seção 4 são apresentados os resultados de uma nova avaliação empírica, onde o  $\varphi$ -PBIL, utilizando o conjunto de parâmetros encontrado na seção 3, é comparado a um algoritmo representante do estado da arte em computação evolutiva: o UEBNA (Peña et al., 2005) – unsupervised estimation of Bayesian network algorithm. Finalmente, na seção 5, são discutidos os resultados e implicações deste trabalho.

## 2 Computação evolutiva

Algoritmos de computação evolutiva procuram resolver problemas de busca e otimização através da manutenção de sucessivas populações de indivíduos, os quais representam soluções do problema. Mecanismos bioinspirados de combinação e variação aplicados a uma população atual levam à obtenção de novas populações, permitindo explorar o espaço de busca.

No sGA (Holland, 1975) os mecanismos de recombinação consistem na aplicação de operadores genéticos, como cruzamento e mutação, diretamente sobre a codificação dos indivíduos. Ao contrário, os chamados algoritmos de estimação de distribuição (EDAs) seguem também a abordagem evolutiva, porém a etapa da geração da nova população é diferente do sGA. Nos EDAs, os indivíduos novos são gerados a partir de uma distribuição conjunta de probabilidades, obtida a partir de indivíduos selecionados anteriormente.

O histórico da evolução dos EDAs levou à confiança em modelos de probabilidade sucessivamente mais flexíveis e complexos, até se atingirem, no estado da arte, algoritmos que se baseiam na indução de

redes Bayesianas. Esta abordagem, que se baseia no aprendizado da ligação entre as variáveis do problema através da inferência das dependências entre elas, vem atingindo bons resultados em diversos problemas artificiais e algumas aplicações, porém o custo computacional associado à etapa de indução do modelo constitui ainda um ponto a ser melhorado.

Dois mecanismos são frequentemente relatados como sendo os mais importantes para garantir o sucesso de um algoritmo de computação evolutiva: o aprendizado de ligação e a manutenção de diversidade.

O aprendizado de ligação visa a identificação de subestruturas, as quais devem ser mantidas durante a recombinação (Harik, 1997). Enquanto o algoritmo genético original (sGA), com cruzamento em um ponto, depende da ordenação correta dos genes no cromossomo para obter sucesso, outras abordagens procuram capturar estas interações entre genes através de mecanismos adicionais.

Problemas que apresentam múltiplos ótimos globais demandam mecanismos que sejam capazes de garantir manutenção da diversidade na população. Estas técnicas procuram manter subpopulações separadas por certos períodos, evitando a recombinação entre indivíduos de subpopulações diferentes e prevenindo a convergência prematura para um dos ótimos. Além disso, técnicas de manutenção de diversidade são também aplicadas em problemas unimodais (Mahfoud, 1995). Verifica-se que estas técnicas podem garantir uma busca mais efetiva, mantendo soluções sub-ótimas até que o verdadeiro ótimo global se torne mais claro.

Recentemente, a questão da otimização multimodal foi abordada através de um EDA. Em Peña et al. (2005) é apresentado o algoritmo UEBNA, que é uma solução baseada no aprendizado não supervisionado de redes Bayesianas. A rede é aprendida considerando a inclusão de uma variável não-observada,  $c$ , a qual representa a pertinência a uma das subpopulações. Espera-se que o algoritmo seja capaz de detectar a atribuição correta dos ótimos globais a subpopulações diferentes, ao mesmo tempo em que captura as dependências entre genes.

Em um outro trabalho recente (Pelikan and Goldberg, 2000) foi aplicado um EDA univariado – o qual assume independência entre os genes – a problemas multimodais simétricos, adotando o algoritmo de agrupamento particional  $k$ -médias como mecanismo de manutenção de diversidade.

O potencial benefício proposto por estes trabalhos é o de controlar e evitar a possibilidade de cruzamento entre subpopulações, alegando que esta abordagem gera frequentemente indivíduos inferiores em termos de *fitness*. Por outro lado, sabe-se que a combinação criteriosa da informação contida em indivíduos diferentes pode promover uma efetiva exploração do espaço de busca.

Na próxima seção um trabalho recente é revisado, o qual propõe-se a realizar esta exploração, efetuando adequadamente a combinação entre subpopulações di-

ferentes, e evitando etapas computacionalmente caras de aprendizado ao longo do processo.

### 3 Um algoritmo de computação evolutiva baseado em agrupamentos

O algoritmo  $\varphi$ -PBIL procura resolver os problemas da manutenção de diversidade e do aprendizado de ligação através da adoção de um mecanismo principal baseado no agrupamento dos indivíduos e na combinação da informação adquirida através destes agrupamentos (Emmendorfer and Pozo, 2007). A figura 1 mostra o pseudocódigo do algoritmo.

$\varphi$ -PBIL é um EDA que segue uma arquitetura incremental, onde um único indivíduo é gerado a cada iteração, em oposição à abordagem mais comum da adoção de uma sucessão de etapas claramente definidas do processo evolutivo, chamadas de “gerações”. Um número  $k$  de agrupamentos é mantido e continuamente atualizado. Sempre que um novo indivíduo é gerado, ele substitui o pior indivíduo na população (desde que o *fitness* do novo indivíduo é maior ou igual ao do antigo) e, na seqüência, as hipóteses de agrupamento e modelos probabilísticos para cada subpopulação são atualizados, refletindo esta alteração na população. Cada agrupamento define uma subpopulação e, como apenas variáveis binárias são admitidas, então os modelos probabilísticos para as subpopulações correspondem a proporções binomiais  $\hat{\Pi} = (\hat{\pi}_{i,j})$  que denotam a proporção de indivíduos que tem o valor 1 para cada gene  $j$  em cada subpopulação  $i$ . Cada subpopulação tem associado um vetor de probabilidades (PV)  $\hat{\pi}_{i,\cdot}$ . O critério de parada é a perda da diversidade nos PVs: o algoritmo termina quando todos os  $\hat{\pi}_{i,j}$ s saturam (atingem algum valor maior que 0,95 ou menor que 0,05).

A geração de um novo indivíduo segue aproximadamente o mesmo processo comum a outros EDAs. No nosso caso, existem dois procedimentos para a geração de um indivíduo: um, mais simples, que parte da escolha de um dos PVs de modo proporcional ao *fitness* médio dos indivíduos, e gera um novo indivíduo amostrando a partir do PV escolhido. O outro procedimento busca obter um novo indivíduo que é resultado do cruzamento de duas subpopulações existentes.

Este mecanismo de cruzamento entre subpopulações é a principal característica do  $\varphi$ -PBIL, que o diferencia dos demais EDAs que também se preocupam com esta questão da manutenção de diversidade. Este mecanismo permite a combinação da informação relevante de dois PVs, procurando sempre manter a porção mais informativa de cada “pai”. A probabilidade de realizar este tipo de cruzamento a cada novo indivíduo gerado é determinada pelo parâmetro  $p_c$ .

Neste processo, dois agrupamentos pai –  $A$  ou  $B$  – são escolhidos, aleatoriamente e proporcionalmente ao *fitness* médio. O novo indivíduo é gerado a partir de um PV temporário, o qual é criado copiando-se proporções de um dos pais, escolhendo sempre qual pai vai fornecer cada posição  $j$  do PV –  $\hat{\pi}_{A,j}$  ou  $\hat{\pi}_{B,j}$ .

1. *Inicialização*: Gerar uma população inicial aleatória de tamanho  $N_0$ , calcular o *fitness* dos indivíduos e selecionar apenas os  $N_w < N_0$  melhores.
2. *Aprendizado*: Aprender agrupamentos a partir da população. Para cada cluster, um vetor (PV) de proporções binomiais é obtido. Duas matrizes são calculadas: uma para as proporções  $\hat{\Pi} = (\hat{\pi}_{i,j})$  e outra para as medidas de informação  $\hat{W} = (\hat{w}_{i,j})$ .
3. *Geração de indivíduo*: Gerar um novo indivíduo  $H$ , escolhendo aleatoriamente entre os seguintes procedimentos: (i) gerar um indivíduo amostrando a partir de um dos PVs, ou: (ii) aplicar recombinação entre PV's de dois pais, utilizando a informação em  $\hat{W}$ , e então amostrar a partir do novo PV temporário para gerar o novo indivíduo.
4. *Seleção*: Calcular o *fitness* do novo indivíduo  $H$ ,  $F_H$ . Se  $H$  não é inferior ao pior indivíduo da população, então apagar este pior indivíduo e inserir  $H$  na população. Caso contrário, o novo indivíduo é descartado.
5. Atualizar os agrupamentos e as matrizes  $\hat{\Pi}$  e  $\hat{W}$ .
6. Repetir passos 3 a 5 enquanto critérios de convergência não forem satisfeitos.

Figura 1: Pseudocódigo do algoritmo  $\varphi$ -PBIL

Para guiar esta escolha, de modo a buscar sempre o pai mais informativo, é adotada uma medida  $\hat{w}_{i,j}$ , que avalia o quanto o agrupamento  $i$  é informativo para a distribuição do gene  $j$ . Esta é a diferença entre a entropia da distribuição do gene  $j$ , antes e depois de observar o cluster  $i$ .

Um mecanismo de perturbação dos PVs foi adicionado, de modo a incrementar a busca local, permitindo que um certo alelo seja gerado mesmo quando a proporção esteja saturada indicando o alelo complementar. Para isso, a proporção de 1s em um gene para os indivíduos de um agrupamento pode ser obtida através do estimador de Wilson (Agresti and Coull, 1998):

$$(total\_de\_uns + 2) / (total\_de\_indivíduos + 4)$$

ao invés do estimador de máxima verossimilhança (média amostral). O parâmetro  $p_w$  determina a probabilidade de aplicação do estimador de Wilson a cada novo indivíduo gerado.

Um outro recurso visa favorecer a resolução de problemas hierárquicos e com blocos construtores sobrepostos. Para isto, são mantidas sempre duas hipóteses de agrupamento: a atual e uma, anterior, que tenha apresentado a melhor performance em termos de número de indivíduos gerados e aceitos. Com isso, blocos construtores detectados anteriormente são mantidos e utilizados por um período de tempo mais longo do processo evolutivo. A probabilidade de aplicação de uma hipótese anterior a cada novo indivíduo gerado

é controlada pelo parâmetro  $p_{old}$ .

Em relação aos parâmetros do algoritmo, três deles não possuem valor *default*: tamanho da população inicial  $N_0$ , tamanho da população de trabalho  $N_w$  e número de clusters  $k$ .

Para três outros parâmetros, foram ajustados valores *default* através de investigação empírica, mostrada na figura 2. Adotamos três problemas-teste conhecidos na literatura: se e somente se hierárquico (HIFF) misturado (Watson et al., 1998), armadilha concatenada-5 (Pelikan et al., 1999) e particionamento de grafos (Peña et al., 2005), com tamanhos de instância 64, 50 e 42, correspondendo respectivamente às instâncias Pshuff64, Ptrapfive50 e Pcatring42 (a mesma que em Peña et al. (2005)). O número de ótimos globais varia entre as instâncias: Pcatring42 apresenta 6 ótimos globais, enquanto Ptrapfive50 possui um único e Pshuff64 possui 2. Os demais parâmetros foram, para esta avaliação, ajustados em valores que foram empiricamente verificados como sendo próximos aos mínimos necessários para a resolução de cada problema. Assim, foram ajustados  $N_0 = 3.000$ ,  $N_w = 300$  e  $k = 15$  para Pshuff64 e Ptrapfive50 e  $N_0 = 2.500$ ,  $N_w = 250$  e  $k = 10$  para Pcatring42.

A mudança em cada um dos parâmetros pode causar efeitos significativos no comportamento do algoritmo. Valores extremos para  $p_{old}$  e  $p_c$  claramente fazem a performance reduzir para todos os problemas testados. Já o incremento de  $p_w$ , por sua vez, parece exercer impacto negativo sobre a velocidade de convergência. Por outro lado, este mesmo parâmetro parece exercer influência positiva sobre a efetividade, principalmente para a instância Pshuff64.

Analisando os resultados, e observando que valores extremos para os três parâmetros testados demonstram ser indesejáveis, então se fixaram os valores *default* para  $p_c$ ,  $p_{old}$  e  $p_w$  todos em 50%, os quais representam um compromisso de eficiência e efetividade razoável para todos os problemas testados. Entretanto, a avaliação marginal dos parâmetros que levam à efetividade máxima para alguns problemas leva a considerar também o conjunto  $p_c = 0, 50$ ,  $p_{old} = 0, 25$  e  $p_w = 0, 75$  embora se perceba, nos gráficos da figura 2, que valores maiores para  $p_w$  podem levar a um incremento no número de avaliações de *fitness* realizadas.

#### 4 Avaliação Empírica

O objetivo desta seção é apresentar uma avaliação empírica do algoritmo  $\varphi$ -PBIL resolvendo um problema de otimização combinatória multimodal bastante conhecido; o particionamento de grafos.

O objetivo do problema do particionamento de grafos em duas partes (ou bisseção) é o de dividir o conjunto de nós de um dado grafo em dois conjuntos de igual tamanho, de modo que o número de vértices entre os conjuntos seja mínimo. O *fitness* de uma solução é calculado como o número total de nós menos o número de vértices conectando os dois conjuntos.

A codificação adotada é baseada em um vetor binário onde cada gene representa o rótulo da partição de um nó do grafo: 0 ou 1. Uma solução factível é aquela em que o número de genes em 1 é igual ao número de genes em 0, já que os conjuntos devem ter tamanhos iguais. O particionamento de grafos é um problema potencialmente multimodal, devido a eventuais regularidades na própria estrutura do grafo e também devido a codificação do problema, já que a rotulagem com 1 ou com 0 é indiferente.

É possível perceber que a geração de um novo indivíduo não é uma operação fechada sobre este conjunto de possíveis soluções, logo um operador de reparação foi adotado nos experimentos realizados aqui, bem como em Peña et al. (2005). Este operador é aplicado sempre que existe desequilíbrio entre 1s e 0s no indivíduo gerado. Um dos genes que possua o valor majoritário (0 ou 1) é escolhido aleatoriamente e invertido, e o processo repete até que a diferença esteja corrigida.

As instâncias verificadas neste trabalho são as mesmas consideradas em Peña et al. (2005): Pgrid16, Pgrid36, Pgrid64, Pcat28, Pcat42, Pcat56, Pcatring28, Pcatring42, Pcatring56 e Pcatring84, com número de ótimos globais variando entre 2 e 6 e o número de nós variando entre 16 e 84.

Uma comparação entre  $\varphi$ -PBIL e UEBNA é descrita agora. UEBNA é um algoritmo baseado no aprendizado não-supervisionado de redes Bayesianas, que se mostrou competente na resolução de problemas de otimização multimodais (Peña et al., 2005), apresentando efetividade e eficiência superiores quando comparado aos outros algoritmos importantes como o EBNA, que é baseado no aprendizado supervisionado de redes Bayesianas.

Todos os resultados para o UEBNA apresentados aqui foram extraídos de Peña et al. (2005). Dez rodadas independentes de cada algoritmo são executadas para cada instância. O conjunto de valores  $p_c = 0, 50$ ,  $p_{old} = 0, 25$  e  $p_w = 0, 75$ , encontrados na seção 3, é mantido para as 10 instâncias na avaliação do  $\varphi$ -PBIL. Além disso, o tamanho da população inicial  $N_0 = 4.000$  é o mesmo valor adotado para o UEBNA. Adicionalmente, o tamanho da população de trabalho é  $N_w = 400$  e o valor de  $k$  é obtido, assim como em Peña et al. (2005), de modo a maximizar a performance do algoritmo, sendo selecionado do intervalo  $k \in [2, 3...20]$ .

A tabela 1 resume os resultados das comparações. As linhas diferenciam cada problema e algoritmo, e as colunas mostram resultados de estatísticas sobre os experimentos realizados. As informações reportadas no formato média  $\pm$  desvio padrão (*sd*) são  $\bar{O}t.$  e  $Av.$ , que correspondem, respectivamente, ao número de ótimos encontrados e o número de avaliações de *fitness* efetuadas.

É possível perceber que, para todos os problemas, o algoritmo  $\varphi$ -PBIL é capaz de obter convergência para todos os ótimos globais com grande vantagem em termos de eficiência, ou seja, após um número consi-

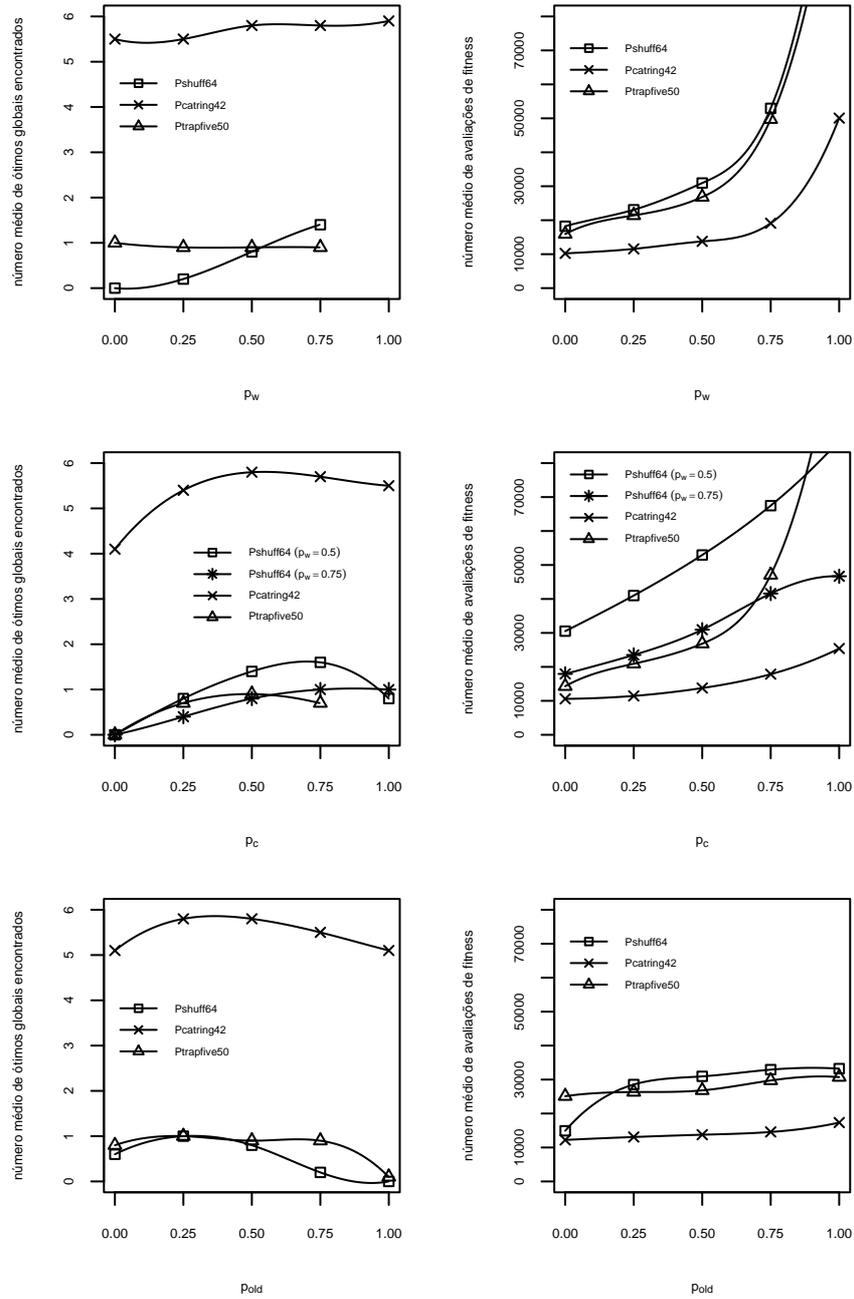


Figura 2: Verificação empírica da sensibilidade do  $\varphi$ -PBIL a três parâmetros  $p_c$  (probabilidade de cruzamento),  $p_{old}$  (probabilidade de usar hipóteses antigas de agrupamento) and  $p_w$  (probabilidade de usar o estimador de Wilson), mostrando o número médio de ótimos globais encontrados e mantidos e o número médio de avaliações de *fitness* até obter convergência. Cinco valores de cada parâmetro (0%, 25%, 50%, 75% e 100%) são considerados em cada linha. Onde não estiver explícito, deve-se assumir os valores *default* para os parâmetros  $p_c = 50\%$ ,  $p_w = 50\%$  and  $p_{old} = 50\%$ . Cada ponto representa a média de 10 rodadas independentes do algoritmo (replicações). As linhas nos gráficos são resultado de interpolação *spline*. Sempre que o número médio de avaliações de *fitness* excede 100.000 as estatísticas para o número de ótimos globais e seu respectivo ponto e segmento de linha não são mostrados no gráfico.

Tabela 1: Efetividade e eficiência do  $\varphi$ -PBIL, comparadas à do UEBNA para 10 instâncias do problema de bisseção de grafos. Um total de 10 rodadas independentes do algoritmo são executadas para cada problema, sendo que todas as rodadas reportadas atingiram pelo menos um pico (ótimo global).

Problema	EDA	Ót.±sd	Av.±sd	Problema	EDA	Ót.±sd	Av.±sd
Pgrid16 (2 picos)	UEBNA $k = 4$	2,0±0,0	51400±2366	Pcat56 (2 picos)	UEBNA $k = 4$	2,0±0,0	96400±2366
	$\varphi$ -PBIL $k = 5$	2,0±0,0	10126±606		$\varphi$ -PBIL $k = 10$	2,0±0,0	46151±4362
Pgrid36 (2 picos)	UEBNA $k = 2$	2,0±0,0	85600±8462	Pcatring28 (4 picos)	UEBNA $k = 2$	4,0±0,0	54700±949
	$\varphi$ -PBIL $k = 5$	2,0±0,0	28963±10754		$\varphi$ -PBIL $k = 5$	4,0±0,0	12694±853
Pgrid64 (2 picos)	UEBNA $k = 4$	2,0±0,0	124900±3479	Pcatring56 (4 picos)	UEBNA $k = 8$	3,8±0,4	96400±1897
	$\varphi$ -PBIL $k = 10$	2,0±0,0	64245±10999		$\varphi$ -PBIL $k = 10$	3,9±0,3	48837±12115
Pcat28 (2 picos)	UEBNA $k = 2$	2,0±0,0	57100±2846	Pcatring42 (6 picos)	UEBNA $k = 6$	5,9±0,3	75700±3302
	$\varphi$ -PBIL $k = 5$	2,0±0,0	14311±1299		$\varphi$ -PBIL $k = 15$	6,0±0,0	32361±1513
Pcat42 (2 picos)	UEBNA $k = 2$	2,0±0,0	73900±1449	Pcatring84 (6 picos)	UEBNA $k = 10$	4,8±0,8	121000±3162
	$\varphi$ -PBIL $k = 10$	2,0±0,0	29714±3644		$\varphi$ -PBIL $k = 20$	5,7±0,7	84539±9300

deravelmente menor de avaliações de *fitness*. Além disso, para problemas maiores, existe também vantagem em termos de número de ótimos encontrados. Para Pcatring56 e Pcatring42, por exemplo, existe ligeira vantagem para  $\varphi$ -PBIL enquanto que, para Pcatring84, o algoritmo  $\varphi$ -PBIL atinge um resultado consideravelmente superior, o que permite supor que  $\varphi$ -PBIL possua escalabilidade mais adequada que a de seu competidor. Em Emmendorfer and Pozo (2007) uma comparação semelhante foi conduzida onde o  $\varphi$ -PBIL foi rodado com o conjunto de parâmetros *default* e sem o operador de reparação. Lá a vantagem do  $\varphi$ -PBIL em relação ao UEBNA não se mostrou tão evidente quanto aqui, principalmente para instâncias maiores.

É importante ressaltar que o  $\varphi$ -PBIL consegue resultados adequados com populações relativamente pequenas, e com a complexidade computacional relacionada à atualização do  $k$ -médias, que é menor que a do aprendizado de redes Bayesianas.

## 5 Conclusão

Verificou-se que um EDA bastante simples, o qual confia basicamente num esquema de combinação entre agrupamentos para explorar o espaço de busca, é capaz de resolver adequadamente um problema globalmente multimodal, dispensando a adoção de modelos probabilísticos mais complexos ou de etapas computacionalmente custosas no processo de otimização.

Outras classes de problemas devem ser abordadas em trabalhos futuros, de modo a verificar até que ponto a metodologia proposta, implementada na forma no algoritmo  $\varphi$ -PBIL, pode ser uma alternativa viável e geral para computação evolutiva.

## Referências

Agresti, A. and Coull, B. (1998). Approximate is better than exact for interval estimation of binomial proportions, *The American Statistician* **58**: 119–126.

Emmendorfer, L. and Pozo, A. T. R. (2007). Otimização multimodal através de computação evolutiva e análise de agrupamentos, *Anais do VI Encontro Nacional de Inteligência Artificial*, Rio de Janeiro, Brasil.

Harik, G. R. (1997). *Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms*, PhD thesis, University of Michigan.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI.

Mahfoud, S. W. (1995). *Niching methods for genetic algorithms*, Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, USA.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on Mathematics, Statistics and Probability*, pp. 281–296.

Pelikan, M. and Goldberg, D. E. (2000). Genetic algorithms, clustering, and the breaking of symmetry, *Parallel Problem Solving from Nature VI*, pp. 385–394.

Pelikan, M., Goldberg, D. E. and E.Cantu-Paz (1999). BOA: The bayesian optimization algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pp. 525–532.

Peña, J., Lozano, J. and Larrañaga, P. (2005). Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of bayesian networks, *Evol. Comput.* **13**(1): 43–66.

Watson, R. A., Hornby, G. and Pollack, J. B. (1998). Modeling building-block interdependency, *Parallel Problem Solving from Nature V*.