Otimização Multimodal através de Computação Evolutiva e Análise de Agrupamentos

Leonardo Ramos Emmendorfer[1], **Aurora Trinidad Ramirez Pozo**[2]

[1]Programa de Pós-Graduação em Métodos Numéricos em Engenharia
Universidade Federal do Paraná (UFPR)

[2]Departamento de Informática
Universidade Federal do Paraná (UFPR) – Centro Politécnico
Caixa Postal 19081 – CEP 81531-990
Curitiba, PR – Brasil

{leonardo,aurora}@inf.ufpr.br

Abstract. *Diversity preservation has shown to be very important for identification and maintenance of the problem structure, as much as for keeping several global optima during the process, in the context of evolutionary computation. The currently most important algorithms adopt diversity preservation techniques just as an auxiliary tool in the process, while trusting on more sophisticated models for the identification of the problem structure. A novel approach is proposed, where a clustering algorithm plays a central role in the evolutionary process, beyond just maintaining the diversity required. Empirical results show the effectiveness of this new approach when solving multimodal optimization problems.*

Resumo. *A preservação de diversidade tem se mostrado muito importante em computação evolutiva, tanto para permitir a identificação e manutenção da estrutura do problema, quanto para manter múltiplos ótimos globais ao longo do processo evolutivo. Os principais algoritmos existentes procuram utilizar a manutenção de diversidade apenas como uma ferramenta auxiliar do processo, confiando em modelos mais sofisticados para a identificação da estrutura do problema. Uma nova abordagem é proposta, onde um algoritmo de agrupamento exerce um papel central no processo evolutivo, além de apenas garantir a diversidade necessária. Resultados empíricos mostram que a nova abordagem é efetiva na resolução de problemas multimodais.*

1. Introdução

As técnicas de Computação Evolutiva (CE), como os Algoritmos Genéticos (GA), se destacam pela sua ampla aplicabilidade, apesar da 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 freqüentemente presentes: seleção de indivíduos promissores e recombinação destes indivíduos, de modo a obter uma nova população. No algoritmo genético simples (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, sendo capazes de detectar dependências da ordem do tamanho do problema.

Geralmente os algoritmos de CE, incluindo os EDAs, necessitam de algum mecanismo que garanta a manutenção da diversidade da população. De fato, a manutenção da diversidade é importante para que o algoritmo identifique corretamente a estrutura do problema em questão e, com isso, gere novos indivíduos que respeitem as interações existentes entre as variáveis. No caso de problemas globalmente multimodais, ou seja, com múltiplos ótimos globais, é desejável também que o algoritmo de CE seja capaz de manter sub-populações (ou nichos), de modo a promover a convergência em cada uma destas sub-populações de modo controlado, evitando o cruzamento entre indivíduos de sub-populações diferentes. Entre as diversas técnicas adotadas, incluem-se algoritmos de agrupamento, como o k-médias [McQueen 1967].

Neste trabalho, é proposta uma nova metodologia de CE baseada na aplicação de algoritmos de agrupamento para separar os indivíduos por similaridade, a qual se diferencia das demais abordagens conhecidas pois a combinação entre subpopulações é incentivada aqui, pois se admite que esta combinação potencializa a exploração do espaço de busca. Assim, os problemas de manutenção de diversidade e otimização multimodal são tratados sob um único mecanismo, relativamente simples.

As próximas duas seções apresentam algumas das dificuldades enfrentadas pelos algoritmos de CE: o aprendizado da estrutura do problema (ou aprendizado de ligação) – onde também é discutida a manutenção da diversidade – e a otimização de problemas globalmente multimodais. A seguir, na seção 4, é apresentado um novo algoritmo, o qual é capaz de tratar estes problemas interrelacionados de forma unificada. Avaliações empíricas são conduzidas na seção 5 e, finalmente, na seção 6 são discutidos os resultados e implicações deste trabalho.

2. Aprendizado de Ligação

Entre as funções artificiais mais exploradas na literatura como problemas de otimização para GAs duas classes se destacam: funções aditivamente decomponíveis (AD) e funções hierarquicamente decomponíveis (HD). Nas funções AD não existe uma estrutura de dependência entre subestruturas (ou blocos construtores – BBs), de modo que a contribuição de cada BB para o fitness não depende do valor dos demais BBs. Este é o caso das funções armadilha de ordem k (*k-trap*), onde é possível particionar o conjunto de genes em agrupamentos de tamanho constante k, definindo subproblemas independentes.

Problemas aditivamente decomponíveis não modelam a interdependência entre blocos construtores, já que apresentam uma natureza separável, onde cada subproblema pode ser resolvido de forma paralela em relação aos demais. Já no caso das funções HD, a estrutura é muito mais complexa, já que existe interação entre BBs. Quando combinados,

os BBs inicialmente identificados originam novos BBs de nível superior, cuja contribuição ao fitness não é obtida meramente a partir da soma dos BBs originais. Entre os problemas HD destaca-se o HIFF [Watson et al. 1998] – Se e Somente Se Hierárquico – o qual apresenta interações da ordem do tamanho do problema. De fato, embora a hipótese de blocos construtores sugira uma certa forma de modularidade, isto não significa que os BBs devam, necessariamente, ser independentes uns dos outros, de modo que funções HD ilustram uma classe de problemas que deve ser explorada e resolvida por algoritmos de Computação Evolutiva.

Havendo interação entre os genes, seja em estruturas hierárquicas ou numa forma aditivamente decomponível, é necessário algum mecanismo que permita capturar a informação relativa a estas ligações entre os genes para o sucesso dos algoritmos de Computação Evolutiva. Estes mecanismos são denominados, de modo geral, de técnicas de aprendizado de ligação (*linkage learning*). Na sua forma mais simples, elas permitem a identificação de blocos construtores que devem ser conservados durante o cruzamento [Harik 1997]. Um bloco construtor consiste de um grupo de genes interagentes, que contribuem positivamente para a função de fitness. Em outros termos, o aprendizado de ligação é o mecanismo que promove a identificação e preservação das interações entre os genes no cromossomo, permitindo capturar e respeitar a estrutura do problema a ser resolvido ao longo do processo evolutivo.

Segundo Jakulin [Jakulin and Bratko 2004], a interação entre variáveis deve ser entendida como um todo irredutível, ou como uma dependência que não pode ser quebrada, de modo que a função distribuição de probabilidade conjunta não fatora no produto das distribuições marginais. Ou seja, uma dependência pode ser baseada em diversas interações, enquanto uma interação é uma dependência que não pode ser quebrada. Um exemplos clássico relacionado à detecção de interações em aprendizado de máquina é o do problema do XOR [Minsky and Papert 1969], o qual não é linearmente separável, logo a interação entre as duas variáveis deve ser capturada de alguma forma.

O algoritmo genético simples (sGA), com cruzamento em um ponto, confia na ordenação dos genes quando da codificação do problema. Para que haja sucesso, é necessário que os genes que codificam variáveis relacionadas estejam próximos no cromossomo, de modo que o cruzamento possa, menos provavelmente, separar estes grupos de genes. Outras estratégias mais recentes baseiam-se na adoção de esquemas alternativos de codificação, que vão desde a simples alteração da ordem dos genes, até mecanismos mais complexos, de subespecificação e super-especificação dos indivíduos, como no Messy GA [Goldberg et al. 1989]. Este algoritmo adota também um esquema de separação do processo evolutivo em etapas de identificação e combinação de BBs, procurando garantir que os blocos estejam corretamente identificados, antes de explorar combinações entre eles.

No caso de EDAs, o próprio modelo probabilístico adotado pode ser capaz de capturar dependências entre os genes. Espera-se que as dependências capturadas respeitem as interações existentes no problema. Um EDA baseado no aprendizado de redes Bayesianas pode ser capaz de capturar dependências entre os grupos de genes, como mostra a figura 1, para a função armadilha de ordem 4 [Pelikan et al. 1999]. Alguns EDAs, por outro lado, adotam modelos estatísticos mais simples, os quais assumem independência entre os genes e, portanto, não são capazes de detectar interações. Um dos mais importantes

representantes desta linha é o PBIL (Aprendizado Incremental Baseado em População) [Baluja and Caruana 1995].

Independentemente da metodologia adotada para capturar a estrutura do problema, a preservação da diversidade da população revela-se um mecanismo crucial [Mahfoud 1995], já que a detecção de um bloco construtor só é possível se uma quantidade crítica de indivíduos na população possuem aquele bloco construtor. A preservação da diversidade na população tem sido garantida através das chamadas técnicas de *niching*.

3. Otimização Multimodal Através de Computação Evolutiva

Problemas onde existem diversos ótimos globais são classificados como globalmente multimodais. Problemas de otimização multimodais representam um cenário de dificuldades para muitos algoritmos de Computação Evolutiva, devido à ocorrência de deslocamento genético (*genetic drift*), que consiste na convergência lenta para apenas um dos ótimos. A existência de diversos picos faz com que a convergência se torne lenta, já que a combinação de boas soluções vindas de partes diferentes do espaço de busca pode resultar em soluções ruins.

Nestes problemas, a preservação da diversidade torna-se ainda mais importante do que naqueles onde existe um único ótimo global, de modo que as técnicas de *niching* específicas para problemas multimodais vêm sendo propostas. Embora o *niching* tenha sido introduzido em GAs, primariamente, para manter prevenir a convergência prematura, somente mais tarde foi estendido para permitir a manutenção estável de múltiplos ótimos. Uma classe de problemas multimodais que vem recebendo atenção da comunidade é a dos problemas simétricos, que apresentam a propriedade de que os ótimos globais ocorrem em pares, de modo que a cada ótimo global corresponde o seu complementar binário [Pelikan and Goldberg 2000]. Ou seja, se um indivíduo é ótimo global de um problema simétrico, então seu complementar também o será.

Alguns trabalhos têm abordado esta questão dos problemas multimodais. [Petrowski 1997] propõe o método Clearing, o qual baseia-se na separação da população em subpopulações e no elitismo dentro de cada subpopulação. O indivíduo de maior fitness em cada subpopulação é considerado o indivíduo dominante, e todo o potencial reprodutivo é atribuído apenas para estes indivíduos dominantes. Outros trabalhos, como [Li et al. 2002] também seguem uma linha parecida, mantendo forte elitismo dentro de cada subpopulação. Uma linha diferente é seguida por [Sastry et al. 2005], onde a diversidade é mantida explicitamente no nível dos blocos construtores aprendidos por um EDA, e não no nível dos indivíduos.

Verificou-se também a possibilidade da adoção de técnicas de agrupamento para

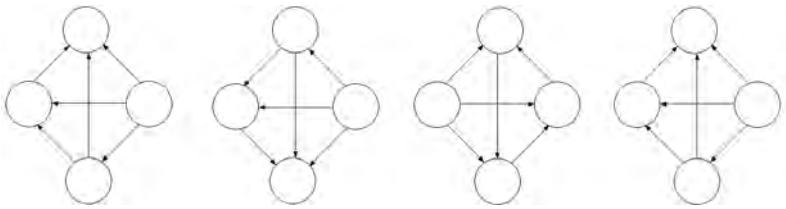

Figura 1. Estrutura de Rede Bayesiana aprendida por um EDA para o problema armadilha-4. Um vértice indica dependência entre duas variáveis.

garantir a manutenção de diversidade e separação das populações. A primeira vez que esta abordagem foi verificada para um EDA foi em [Pelikan and Goldberg 2000]. Neste trabalho, foi aplicando um EDA univariado, que assume independência entre os genes, a problemas multimodais simétricos, adotando o algoritmo de agrupamento particional k-médias [McQueen 1967] como mecanismo de manutenção de diversidade, agrupando indivíduos por similaridade de genótipo.

Em [Peña et al. 2005] é apresentado o algoritmo UEBNA – Algoritmo de Estimação de Redes Bayesianas Não-Supervisionado, que é uma solução que também se baseia no aprendizado não-supervisionado; neste caso 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 através do restante da rede.

O potencial benefício proposto e verificado pelos trabalhos revisados nesta seção é o de controlar a possibilidade de cruzamento entre subpopulações, que costuma gerar indivíduos inferiores em termos de fitness ao longo do processo de otimização. Entretanto, para uma efetiva exploração, é desejável a combinação de bons indivíduos diferentes; uma nova abordagem, baseada neste princípio, é apresentada a seguir.

4. φ-PBIL: Computação Evolutiva Baseada em Agrupamento

A manutenção da diversidade é importante para o sucesso de algoritmos de Computação Evolutiva. Entre as técnicas disponíveis, a análise de agrupamentos é capaz de separar indivíduos por similaridade de genótipo, controlando e evitando as combinações entre sub-populações diferentes, conforme revisado na seção anterior. A detecção dos blocos construtores do problema, por sua vez, fica sob responsabilidade de outros mecanismos, como o operador de cruzamento ou os modelos estatísticos dos EDAs. Para isso, um outro nível de manutenção de diversidade também é exigido, desta vez dentro de cada subpopulação, para que a estrutura do problema seja detectada adequadamente.

A diversidade em uma população pode, portanto, ser explicada tanto pela existência de diversos blocos construtores presentes nos indivíduos desta população quanto pela existência de diversos ótimos globais, sendo que ambos motivos estão interrelacionados. Quando o algoritmo ainda está distante da convergência, pode-se conjecturar que a principal fonte de diversidade da população esteja na diversidade de blocos construtores, já que os ótimos globais ainda não estão claros. Assim, a aplicação de algoritmos de agrupamento permitiria a detecção de diferentes subpopulações, onde cada uma seria caracterizada pela presença de um (ou mais) blocos construtores. O algoritmo φ-PBIL – Aprendizado Incremental Baseado em População e Guiado por Conceitos – aqui apresentado segue exatamente esta abordagem. Este algoritmo está restrito, inicialmente, a problemas com variáveis binárias.

Cada agrupamento (ou subpopulação) é caracterizado por um vetor de proporções binomiais, PV. Cada posição deste PV indica a proporção de 1s em um gene. Assim como em um EDA, novos indivíduos são gerados através da amostragem a partir deste modelo probabilístico, com duas diferenças. A primeira é que, no φ-PBIL existe um PV para cada agrupamento, logo um dos PVs deve ser selecionado aleatoriamente, de modo proporcional ao fitness médio de cada agrupamento correspondente.

Segundo, e mais importante, existe a possibilidade de combinação entre agrupamentos diferentes. Neste caso, dois PVs são combinados criteriosamente. Ou seja, ao contrário dos algoritmos revisados na seção anterior, existe a intenção clara de combinar soluções vindas de subpopulações diferentes, ou seja, possuindo blocos construtores diferentes.

O critério adotado para combinar dois PVs e detectar a interação entre os genes é baseado em uma medida advinda da teoria da informação. No processo de geração de um novo indivíduo são escolhidos, aleatoriamente e proporcionalmente ao fitness médio, dois agrupamentos A e B. Um novo PV será então montado, temporariamente, copiando-se posições de A ou B ($p_{A,j}$ ou $p_{B,j}$ para cada gene j), de modo que é escolhido sempre o "pai"mais informativo para cada gene.

Seja $w_{i,j}$ a medida da quantidade de informação do agrupamento i para o gene j, que é a diferença entre a entropia da distribuição do gene j, antes e depois de observar o agrupamento i. Então a decisão entre A e B para cada gene é simples: se $w_{B,j} > w_{A,j}$ e $w_{B,j} > 0$, então selecionar $p_{B,j}$, caso contrário selecionar $p_{A,j}$. Na figura 2 está ilustrada uma possível situação, onde o total de agrupamentos é $k = 3$. Caso, por exemplo, os agrupamentos 1 e 2 sejam escolhidos para cruzamento, então o PV resultante terá proporções altas para as posições de 1 a 8, o que deve gerar, com alta probabilidade, um indivíduo contendo a combinação de dois blocos construtores de tamanho 4 deste problema artificial.

O algoritmo está descrito na figura 3. É possível perceber que foi adotada uma arquitetura incremental, já que apenas um indivíduo é gerado a cada ciclo. Assim, o k-médias deve apenas atualizar os centróides atuais, ao invés de aprender hipóteses de agrupamento totalmente novas.

Alguns recursos foram adicionados à estrutura básica do φ-PBIL. O primeiro visa incrementar a busca local. Foi incluído um mecanismo de perturbação sobre as proporções binomiais estimadas, de modo a permitir 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 pode ser obtida através do estimador de Wilson como

$$(total\_de\_uns + 2)/(total\_de\_individuos\_no\_agrupamento + 4)$$

Assim, por exemplo, mesmo se todos os 100 indivíduos de um agrupamento possuam o valor 1 em um *locus*, mesmo assim ainda resta ainda, com o estimador de Wilson, uma probabilidade de 2% de um novo indivíduo conter um 0 naquele mesmo locus.

		gene											
		1	2	3	4	5	6	7	8	9	10	11	12
agrupamento	1	0.5	0.6	0.2	0.4	0.9	1.0	0.9	0.9	0.9	0.5	0.6	0.5
	2	1.0	1.0	1.0	0.9	0.1	0.2	0.1	0.3	0.8	0.4	0.7	0.4
	3	0.0	0.0	0.2	0.0	0.5	0.6	0.3	0.6	0.2	0.0	0.1	0.0

Figura 2. Exemplo de uma configuração de proporções binomiais $p_{i,j}$ durante o processo evolutivo, para um certo problema artificial. Os valores inscritos indicam os $p_{i,j}$s, enquando os tons de cinza representam o valor de $w_{i,j}$ correspondente, em escala crescente do preto (mínimo) até o branco (máximo).

1. *Inicialização*: Gerar uma população inicial aleatória, calcular o fitness dos indivíduos e selecionar uma fração dos melhores.
2. *Aprendizado*: Aprender agrupamentos a partir da população. Calcular a matriz $W = (w_{i,j})$ de medidas de informação e obter, para cada agrupamento, um vetor de probabilidades (PV), diretamente a partir dos centróides.
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 W, e então amostrar a partir do novo PV temporário.
4. *Seleção*: Calcular o fitness de H, F_H. Se H não é inferior ao pior indivíduo da população, então apagar este pior indivíduo e inserir H na população. Atualizar os agrupamentos.
5. *Finalização*: Repetir passos 3 e 4 enquanto critérios de convergência não forem satisfeitos.

Figura 3. Pseudocódigo do Algoritmo φ-PBIL

Um outro recurso visa favorecer a resolução de problemas hierárquicos e com blocos construtores sobrepostos. Para isto, a memória de uma hipótese anterior de agrupamento é mantida, para que blocos construtores detectados anteriormente possam ser utilizados por um período de tempo mais longo do processo evolutivo. Na realidade, além da hipótese atual, é mantida também a hipótese que vem apresentando melhor performance, no sentido em que os indivíduos gerados por ela tem sido aceitos com freqüência.

5. Avaliação Empírica

O interesse principal, nesta seção, é verificar a efetividade e eficiência do φ-PBIL em resolver problemas que, além de serem globalmente multimodais, também apresentem uma estrutura de interação entre variáveis. Inicialmente, são brevemente citados alguns problemas desta classe, os quais são considerados na avaliação. A seguir, é realizada uma comparação direta com resultados reportados para um outro algoritmo, o UEBNA [Peña et al. 2005]. O foco é restrito a problemas simétricos, assim como em outros trabalhos relacionados, revisados em seções anteriores.

Dois problemas são utilizados: twomax e bisseção de grafos. Na avaliação empírica, duas instâncias do twomax são verificadas, com tamanhos de problema $n = 50$ e $n = 100$, denotadas por Ptwomax50 e Ptwomax100 respectivamente. Já para o problema da bisseção de grafos, são consideradas 10 instâncias com topologias diferentes: Pgrid16, Pgrtid36, Pgrid64, Pcat28, Pcat42, Pcat56, Pcatring28, Pcatring42, Pcatring56 e Pcatring84. Estes problemas e instâncias são apresentados mais detalhadamente em [Peña et al. 2005].

5.1. Resultados

São apresentados aqui os resultados de uma comparação direta entre o algoritmo φ-PBIL e um outro algoritmo, que é competente para resolver problemas globalmente multimodais: o UEBNA [Peña et al. 2005]. Este algoritmo é representante do estado da arte em

computação evolutiva, e segue uma linha semalhante à dos EDAs mais competentes: o aprendizado de redes Bayesianas a cada geração. No caso do UEBNA em particular, é adotado o aprendizado não supervisionado, com a inclusão de uma variável latente, que denota a pertinência a um dos grupos. Em [Peña et al. 2005] o UEBNA é comparado com outros EDAs compententes, todos procurando resolver 12 instâncias de problemas multimodais simétricos. Deste trabalho são extraídos os resultados aqui apresentados para o UEBNA. Fica clara ali a vantagem do UEBNA em relação aos demais algoritmos.

São realizadas 10 rodadas independentes de cada algoritmo para cada instância de problema. A fim de promover uma comparação correta, os parâmetros *default* do φ-PBIL são fixados para todas as 12 instâncias. Assim, a probabilidade de aplicar cruzamento entre agrupamentos é setada em $p_c = 50\%$, e o estimador de Wilson é adotado na geração de indivíduos. Também o tamanho da população inicial é fixado em $N_0 = 4.000$, o mesmo valor que o usado pelo UEBNA. Cabe aqui uma ressalva, já que a população de trabalho do φ-PBIL é apenas uma fração da população inicial. Nos experimentos com φ-PBIL, a população de trabalho é de $N_w = 400$ indivíduos para todos os problemas. Adotou-se como único parâmetro livre o k, que representa número de agrupamentos. É reportado sempre o valor de k que maximiza performance na avaliação empírica, restrito a $k \in [2, 3...20]$.

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. Estas são separadas nos grupos *Todas as rodadas*, onde as 10 rodadas são sumarizadas e *Rodadas com sucesso*, onde apenas as rodadas que atingiram pelo menos um ótimo global correto são reportadas. As informações reportadas no formato média $\pm$ desvio padrão (*sd*) são *Ót.* e *Av.*, que correspondem, respectivamente, ao número de ótimos globais encontrados e o número de avaliações de fitness efetuadas.

É possível perceber que, para grande parte dos problemas, o φ-PBIL é capaz de obter convergência para todos os ótimos globais com grande vantagem em termos de eficiência, ou seja, após um número consideravelmente menor de avaliações de fitness. Para problemas maiores, entretanto, esta vantagem já não fica tão clara. Para Pcatring56 e Pcatring42, por exemplo, o número de ótimos encontrados é próximo ao esperado para ambos os algoritmos (com certa desvantagem para φ-PBIL em Pcatring56), porém o número de avaliações de fitness é semelhante. Além disso, em Pcatring84, o algoritmo φ-PBIL atingiu resultados inferiores tanto em efetividade quanto em eficiência. Possivelmente, o tamanho de população de trabalho escolhida (400) seja insuficiente para o φ-PBIL. Por outro lado, esta população de 400 indivíduos, pequena para os padrões da Computação Evolutiva, mostra-se suficiente para as demais instâncias testadas. Os resultados obtidos permitem considerar o φ-PBIL como uma alternativa viável entre os algoritmos existentes de Computação Evolutiva.

6. Conclusão

Este trabalho apresenta o algoritmo φ-PBIL, que representa uma abordagem nova para computação evolutiva. Verifica-se o algoritmo é eficaz e eficiente ao resolver diversas instâncias de problemas globalmente multimodais. Para isso, foi feita uma comparação direta com o algoritmo UEBNA [Peña et al. 2005]. Na maior parte das instâncias testadas, o φ-PBIL é mais eficiente e tão eficaz quanto o UEBNA.

Tabela 1. Efetividade e eficiência do φ-PBIL, comparadas à do UEBNA para 12 instâncias de problemas de otimização globalmente multimodais, com número de picos (ótimos globais) variando de 2 a 6. Um total de 10 rodadas independentes do algoritmo são executadas para cada problema.

Problema	EDA	Todas as rodadas		Rodadas com sucesso	
		Ót.±sd	Av.±sd	Ót.±sd	Av.±sd
Ptwomax50	UEBNA $k=2$	2,0±0,0	55000±0	2,0±0,0	55000±0
(2 picos)	φ-PBIL $k=2$	2,0±0,0	11434±202	2,0±0,0	11434±202
Ptwomax100	UEBNA $k=2$	2,0±0,0	76600±1265	2,0±0,0	76600±1265
(2 picos)	φ-PBIL $k=2$	2,0±0,0	13776±310	2,0±0,0	13776±310
Pgrid16	UEBNA $k=4$	2,0±0,0	51400±2366	2,0±0,0	51400±2366
(2 picos)	φ-PBIL $k=4$	2,0±0,0	14463±1469	2,0±0,0	14463±1469
Pgrid36	UEBNA $k=2$	2,0±0,0	85600±8462	2,0±0,0	85600±8462
(2 picos)	φ-PBIL $k=5$	1,2±1,0	49455±8251	2,0±0,0	45600±5947
Pgrid64	UEBNA $k=4$	2,0±0,0	124900±3479	2,0±0,0	124900±3479
(2 picos)	φ-PBIL $k=5$	1,0±1,1	130317±20219	2,0±0,0	121773±22065
Pcat28	UEBNA $k=2$	2,0±0,0	57100±2846	2,0±0,0	57100±2846
(2 picos)	φ-PBIL $k=4$	2,0±0,0	26309±2020	2,0±0,0	26309±2020
Pcat42	UEBNA $k=2$	2,0±0,0	73900±1449	2,0±0,0	73900±1449
(2 picos)	φ-PBIL $k=4$	1,9±0,3	53034±4121	1,9±0,3	53034±4121
Pcat56	UEBNA $k=4$	2,0±0,0	96400±2366	2,0±0,0	96400±2366
(2 picos)	φ-PBIL $k=4$	1,3±0,9	82047±8205	1,9±0,4	81474±9199
Pcatring28	UEBNA $k=2$	4,0±0,0	54700±949	4,0±0,0	54700±949
(4 picos)	φ-PBIL $k=8$	4,0±0,0	26498±1275	4,0±0,0	26498±1275
Pcatring56	UEBNA $k=8$	3,8±0,4	96400±1897	3,8±0,4	96400±1897
(4 picos)	φ-PBIL $k=10$	3,2±1,0	94801±6635	3,2±0,1	94801±6635
Pcatring42	UEBNA $k=6$	5,9±0,3	75700±3302	5,9±0,3	75700±3302
(6 picos)	φ-PBIL $k=20$	5,9±0,3	61296±2926	5,9±0,3	61296±2926
Pcatring84	UEBNA $k=10$	4,8±0,8	121000±3162	4,8±0,8	121000±3162
(6 picos)	φ-PBIL $k=10$	2,9±1,0	165627±13280	2,9±0,1	165627±13280

Além disso, o algoritmo φ-PBIL representa uma abordagem mais simples para a Computação Evolutiva, visto que consiste na sucessiva geração de indivíduos a partir de modelos binomiais e na atualização incremental de hipóteses de agrupamento da população, através do algoritmo k-médias. A parte mais sofisticada do algoritmo fica por conta de um mecanismo de combinação entre sub-populações, o qual confia na diversidade de blocos construtores como motivador para a separação dos indivíduos em agrupamentos diferentes. Outros algoritmos existentes adotam modelos bastante mais complexos, como a indução de redes Bayesianas.

Trabalhos futuros devem verificar a capacidade do φ-PBIL em resolver problemas hierárquicos, como o HIFF [Watson et al. 1998]. Antes disso, é necessário verificar empiricamente que o mecanismo de combinação entre sub-populações é, conforme esperado, o responsável pela combinação dos blocos construtores detectados pelo mecanismo de agrupamento. Para isso, deverá adotar-se como controle o próprio algoritmo φ-PBIL, porém com o cruzamento entre sub-populações desligado, e problemas aditivamente decomponíveis com blocos construtores de difícil detecção como problemas teste.

Referências

Baluja, S. and Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In *Int. Conf. on Machine Learning*, pages 38–46.

Goldberg, D. E., Korb, G., and Deb, G. (1989). Messy genetic algorithms: Motivation, analysis , and first results. *Complex Systems*, 3:493–530.

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.

Jakulin, A. and Bratko, I. (2004). Testing the significance of attribute interactions. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 52, New York, NY, USA. ACM Press.

Li, J.-P., Balazs, M. E., Parks, G. T., and Clarkson, P. J. (2002). A species conserving genetic algorithm for multimodal function optimization. *Evol. Comput.*, 10(3):207–234.

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. In *Proceedings of the fifth Berkley symposium on Mathematics, Statistics and Probability*, pages 281–296.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Mass.

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

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

Petrowski, A. (1997). A new selection operator dedicated to speciation. In *Proceedings of the 7th International Converence on Genetic Algorithms*, pages 144–451.

Sastry, K., Abbass, H. A., Goldberg, D. E., and Johnson, D. D. (2005). Sub-structural niching in estimation of distribution algorithms. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 671–678, New York, NY, USA. ACM Press.

Watson, R. A., Hornby, G., and Pollack, J. B. (1998). Modeling building-block interdependency. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 97–108, London, UK. Springer-Verlag.