

Aprendizado probabilístico em algoritmos evolutivos para otimização em ambientes dinâmicos

André R. Gonçalves , Fernando J. Von Zuben (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)

Faculdade de Engenharia Elétrica e de Computação (FEEC)

Universidade Estadual de Campinas (Unicamp)

Rua Albert Einstein, 400 – 13083-852 – Campinas, SP, Brasil

{andreric,vonzuben}@dca.fee.unicamp.br

Abstract – One probabilistic interpretation of the general functioning of an evolutionary algorithm, where an adaptive probability density function is manipulated, can serve as motivation for the development of algorithms that explicitly manipulate probabilistic models, instead of using genetic operators. The so-called estimation of distribution algorithms are studied here in the context of optimization in dynamic environments. Some specific operators promote prompt reaction to changing scenarios during the optimization process. In order to investigate the benefits of the proposal, two estimation of distribution algorithms are compared with two genetic algorithms, each one endowed with specific operators to promote diversity. The obtained results, considering a broad set of optimization problems, indicate that the designed probabilistic operators can significantly improve the performance when the optimal solution changes with time, particularly in situations characterized by some correlation between the previous and the current location of the global optimum.

Keywords – Dynamic environments, evolutionary computation, probabilistic models, density estimation.

1. Introdução

Algoritmos evolutivos implementam um processo estocástico e dinâmico, inspirado em princípios da evolução natural, que manipula, simultaneamente, uma população de soluções de forma a obter melhorias contínuas em um problema de otimização [3]. Este processo de melhorias é implementado por meio dos operadores de seleção e geração de novos indivíduos. O método de seleção é responsável por extrair da população as melhores soluções, seja de forma determinística ou probabilística, direcionando o algoritmo evolutivo para regiões do espaço de busca onde a probabilidade de gerar indivíduos melhores tende a ser maior.

Uma possível interpretação para o funcionamento dos algoritmos evolutivos se dá sob a óptica de probabilidades, onde o algoritmo, implicitamente, é guiado por uma função densidade de probabilidade (fdp) adaptativa. A classe de algoritmos evolutivos, denominados Algoritmos de Estimação de Distribuição (AEDs), emprega explicitamente modelos probabilísticos como operadores de exploração do espaço de busca.

Tanto os algoritmos evolutivos com operadores genéticos como os AEDs foram desenvolvidos inicialmente para tratar de problemas de otimização em ambientes estáticos. No entanto, em muitos problemas reais, o ambiente se altera no decorrer do tempo, refletindo diretamente na função-objetivo

que também será alterada. Para estes cenários, são incorporados aos algoritmos evolutivos operadores de controle de diversidade, para que estes sejam capazes de se adaptar aos novos ambientes. Alguns dos operadores comumente empregados serão aqui abordados.

O restante do trabalho é descrito como segue. Na Seção 2, uma interpretação probabilística para o funcionamento dos algoritmos evolutivos é apresentada, descrevendo brevemente os AEDs. A adaptação de algoritmos evolutivos a problemas de otimização em ambientes dinâmicos é tratada na Seção 3. Uma investigação sobre o desempenho de algumas propostas de algoritmos populacionais para otimização em ambientes dinâmicos é apresentada na Seção 4 e os resultados são mostrados na Seção 5. As considerações finais estão na Seção 6.

2. Interpretação probabilística do algoritmo evolutivo e algoritmos de estimação de distribuição

Na aplicação de meta-heurísticas populacionais, geralmente não existe nenhuma informação sobre a superfície de *fitness*, e a função densidade de probabilidade (fdp) utilizada para gerar a população inicial é uniforme e multivariada. Assim, diz-se que a população inicial é obtida de maneira aleatória.

Após avaliadas as soluções iniciais, o método de seleção é empregado, e este, por sua vez,

apontará regiões promissoras no espaço de busca. Estas informações serão utilizadas pelo algoritmo evolutivo para atualizar a distribuição de probabilidades sobre os melhores indivíduos no espaço de busca. Assim, a nova população será amostrada com base nesta fdp adaptada e, provavelmente, indivíduos mais aptos serão gerados.

A partir disso, ao invés de serem aplicados operadores de reprodução e mutação para explorar as regiões promissoras, pode-se utilizar diretamente uma fdp para estimar e amostrar sobre estas regiões. Esse é o conceito central dos Algoritmos de Estimação de Distribuição (AEDs).

Os AEDs são métodos evolutivos que empregam técnicas de estimação de distribuição, ao invés de operadores genéticos. Estes algoritmos são motivados por algumas deficiências dos algoritmos evolutivos em geral, dentre elas a ausência de um mecanismo de detecção e preservação de *blocos construtivos* [7].

A etapa fundamental nesta classe de algoritmos é como estimar a verdadeira distribuição das soluções promissoras. Na realidade, a estimação da distribuição de probabilidade conjunta, associada às variáveis que geraram os indivíduos selecionados, constitui o gargalo destes algoritmos. Será necessário então um balanço entre a precisão desta estimação e seu custo computacional.

Os algoritmos presentes nesta classe diferenciam-se basicamente pela forma como a estimação é realizada, sendo classificados de acordo com a complexidade do modelo probabilístico utilizado: *variáveis sem dependência*; *dependência aos pares*; *dependência multivariada* e *modelos de mistura* [7]. Uma revisão sobre os AEDs pode ser encontrada em [6].

3. Algoritmos evolutivos em ambientes dinâmicos

Algoritmos evolutivos (AEs) tradicionais, os quais foram basicamente construídos para tratar de problemas estacionários, tendem a alcançar baixa performance em ambientes dinâmicos, devido à ausência de mecanismos de reação às novas condições.

Uma abordagem direta para tratar de problemas de otimização em ambientes dinâmicos é a simples reinicialização do algoritmo, a cada alteração no ambiente. No entanto, esta abordagem

é muitas vezes impraticável, devido ao alto custo computacional para resolver um problema sem reutilizar as informações do passado [1].

Além disso, se for considerado que as alterações sofridas pelo ambiente são relativamente pequenas, provavelmente o novo ótimo estará, de certa forma, relacionado com o ótimo antigo. Neste caso, um algoritmo capaz de transferir conhecimento da população anterior para a atual tende a ser mais eficiente do que uma simples reinicialização.

Para serem adaptados a problemas dinâmicos, AEs devem ser dotados de operadores específicos, sejam explícitos ou implícitos, responsáveis pela manutenção da diversidade na população e, principalmente, por evitar a estagnação do algoritmo. Alguns destes operadores, como apontado por [5], são: (i) geração de diversidade após alteração do ambiente; (ii) manutenção da diversidade por toda a execução (evitar a convergência do algoritmo); (iii) emprego de memórias para armazenar e recuperar informações úteis de um passado recente; e (iv) utilização de múltiplas populações para perseguir múltiplos ótimos locais na superfície de *fitness*.

Neste trabalho, as estratégias (i), (ii) e (iii) serão consideradas pelos algoritmos evolutivos e suas performances serão comparadas sob ambientes dinâmicos com diferentes níveis de complexidade.

4. Experimentos

Nesta seção, são realizados experimentos para analisar a performance de algoritmos evolutivos e AEDs na otimização em ambientes dinâmicos.

4.1. Gerador de ambientes dinâmicos

O benchmark de picos móveis (do inglês, *Moving Peaks Benchmark* - MPB), proposto por [1] é um gerador artificial de ambientes dinâmicos que busca cobrir diversas características encontradas em ambientes dinâmicos reais. O MPB consiste de uma função paramétrica com um determinado número de picos que podem variar suas formas, posições e larguras, ao longo do tempo. A severidade das variações destes parâmetros é controlada pelo parâmetro s . Quanto maior o valor de s , mais drásticas são as modificações do ambiente.

É possível gerar ambientes dinâmicos com diferentes níveis de complexidade, alterando, por

exemplo, o número de picos n (multimodalidade do espaço) e o intervalo entre as alterações (τ).

4.2. Algoritmos analisados

O procedimento de geração de diversidade após a ocorrência de uma alteração no ambiente, como indicado por [5], foi investigado inicialmente por [2], onde foi proposto o *Hypermutation Genetic Algorithm* (HGA). No HGA, a probabilidade de mutação é elevada quando uma alteração é detectada, de forma a aumentar a diversidade, provendo ao algoritmo a capacidade de se adaptar à nova superfície de *fitness*.

Na classe de AEDs, o UMDA (do inglês, *Univariate Marginal Distribution Algorithm*) [8] é um algoritmo simples e de baixo custo computacional, e que faz duas suposições sobre o problema: as variáveis são independentes e cada variável segue uma determinada função densidade de probabilidade (em geral, a distribuição normal). Assim, o modelo probabilístico é um conjunto de n distribuições univariadas, as quais têm seus parâmetros ajustados no decorrer da execução.

Foi considerado também uma versão modificada do HGMM (do inglês, *Hybrid Gaussian Mixture Model*) [4], a qual emprega uma memória das melhores soluções obtidas em estágios passados. O HGMM utiliza um modelo de mistura gaussiano com aprendizado ao longo do tempo para guiar o algoritmo evolutivo, além de estratégias de manutenção de diversidade na população.

Um algoritmo genético tradicional com reinicializações a cada alteração do ambiente (AGr) também tem seus resultados analisados.

Os algoritmos aqui analisados são algumas das várias abordagens para tratar de ambientes dinâmicos. Diversos outros algoritmos foram propostos na literatura, cada qual com suas vantagens e desvantagens. Uma compilação sobre estes algoritmos pode ser encontrada em [5].

5. Resultados

A Tabela 1 apresenta o *offline error* para os algoritmos, HGA, AG com reinicialização (AGr), UMDA e HGMM. O *offline error* é uma métrica comumente utilizada para analisar a performance de algoritmos de otimização em ambientes dinâmicos, sendo ob-

tida pela média do erro absoluto entre o melhor indivíduo e a solução ótima, em cada instante de tempo.

Aqui, o MPB produz ambientes com 5 dimensões, definidos no intervalo $[-5,5]$. Diversos valores de número de picos (n), intervalo entre alterações do ambiente (τ) e severidade das alterações (s) foram considerados.

O *offline error* foi obtido a partir de 30 execuções independentes e, em cada execução, o ambiente é alterado 100 vezes. Em todos os algoritmos, foram utilizadas uma população de 50 indivíduos e seleção por torneio de cinco competidores. Para o UMDA, 50% dos melhores indivíduos foram utilizados para estimar os parâmetros das fdps gaussianas e a nova população é composta por 80% dos indivíduos gerados pelo modelo probabilístico e o restante de forma aleatória. Para o HGA e o AGr, foi utilizada mutação gaussiana ($p_{mut}=0,1$; $std=0,5$) e *crossover* convexo ($p_{cross}=0,8$). No HGA, para hipermutação foi utilizada uma probabilidade igual a 0,8. Para o HGMM, foram utilizados 50% dos indivíduos para estimar o modelo de mistura (parâmetro η) e a mesma quantidade é amostrada para compor a nova população.

Nos algoritmos HGA, UMDA e HGMM, foi empregada uma memória para armazenar as melhores soluções encontradas em estágios passados. Assim, quando uma alteração no ambiente é detectada, estas são reinseridas na população atual. Uma memória de tamanho 20 foi utilizada.

Com base nos resultados obtidos, observa-se que a abordagem de reinicialização (AGr) se mostrou menos eficiente do que os outros algoritmos considerados, em todos os casos analisados. Em ambientes onde o ótimo local sofre pequenas alterações ou salta entre os ótimos locais, métodos que preservam soluções do passado podem rapidamente se adaptar ao novo cenário e, assim, obter melhores resultados. Em situações em que a posição do ótimo global tende a ser independente da posição anterior, a simples reinicialização da população, que não alcançou bom resultados nos experimentos apresentados, pode ser tornar relevante.

Em diversos cenários, o HGMM obteve melhores resultados do que os outros algoritmos analisados, possivelmente pelo fato de explorar diversas regiões promissoras ao mesmo tempo, representadas pelos componentes do modelo de mistura.

s=1					
NP	τ	HGA	AGr	UMDA	HGMM
1	1000	7,51	50,06	2,40	2,18
1	10000	0,91	30,78	0,24	0,57
10	1000	25,32	67,31	20,03	19,49
10	10000	18,25	30,71	17,48	16,84
50	1000	26,28	67,76	19,83	18,50
50	10000	20,09	26,73	18,54	18,39
s=3					
1	1000	16,08	50,83	6,55	6,44
1	10000	1,81	31,13	1,64	0,71
10	1000	33,55	67,02	24,21	25,53
10	10000	19,32	30,40	20,41	17,73
50	1000	36,10	67,90	25,52	23,97
50	10000	20,81	26,84	17,44	19,44
s=6					
1	1000	24,45	48,73	16,08	14,51
1	10000	2,79	27,22	7,14	2,46
10	1000	42,73	67,12	35,23	28,94
10	10000	18,84	30,53	25,85	20,15
50	1000	42,90	67,80	33,36	28,66
50	10000	22,01	26,95	26,07	18,46

Tabela 1. Performance do HGA, AGr, UMDA e HGMM em diferentes configurações de ambientes dinâmicos, tomando s=1, s=3 e s=6.

Em ambientes altamente dinâmicos (1000 avaliações de função entre as alterações do ambiente), os algoritmos de estimação de distribuição obtiveram resultados significativamente melhores, em relação a HGA e AGr. Nestes ambientes, os algoritmos dispõem de pouco tempo para se adaptar e, como os AEDs possuem uma característica de rápida convergência, estes vêm a obter melhores resultados.

6. Considerações finais

Algoritmos de estimação de distribuição manipulam explicitamente modelos probabilísticos que direcionam a exploração para regiões do espaço de busca onde há uma maior probabilidade de gerar soluções mais bem adaptadas.

Em ambientes dinâmicos, a principal modificação realizada nos algoritmos evolutivos é a implantação de um controle efetivo da diversidade na população. Algoritmos que mantêm as soluções bastante dispersas pelo espaço de busca provavelmente se adaptarão rapidamente quando o ambiente for alterado.

Foi apresentado aqui um conjunto de es-

tratégias comumente empregadas para manutenção de diversidade na população, como: uso de memórias, multipopulações, geração de novos indivíduos quando uma alteração é detectada, ou então de forma contínua no decorrer da execução.

Simulações realizadas apontaram que, em muitos cenários, uma simples reinicialização do algoritmo a cada alteração no ambiente não é a melhor estratégia. O histórico da busca se mostra como uma importante informação e que deve ser explorada pelos algoritmos voltados para ambientes dinâmicos.

Referências

- [1] J. Branke. Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1875–1882. IEEE Press, 1999.
- [2] H.G. Cobb. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical report, Naval Research Laboratory, 1990.
- [3] M. Gen and R. Cheng. *Genetic algorithms and engineering optimization*. Wiley-interscience, 2000.
- [4] A.R. Gonçalves and F.J. Von Zuben. Hybrid evolutionary algorithm guided by a fast adaptive gaussian mixture model applied to dynamic optimization problems. In *III Workshop on Computational Intelligence - Joint Conference*, pages 553–558, 2010.
- [5] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments—a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.
- [6] P. Larrañaga. A Review of Estimation of Distribution Algorithms. In P. Larrañaga, J.A. Lozano, editor, *Estimation of Distribution Algorithms: A new tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [7] P. Larrañaga and J.A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Netherlands, 2002.
- [8] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature-PPSN IV*, pages 178–187, 1996.