

UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO HENRIQUE SILVEIRA DA COSTA

ALGORITMO GENÉTICO APLICADO A PROBLEMAS NA ECONOMIA

CURITIBA
2022

Eduardo Henrique Silveira da Costa

Algoritmo Genético Aplicado a Problemas na Economia

Monografia apresentada como requisito à obtenção do grau de Bacharel em Ciências Econômicas pelo Curso de Ciências Econômicas, Setor de Ciências Sociais Aplicada, Universidade Federal do Paraná.

Orientador: Dr. João Basilio Pereima

Curitiba
2022

TERMO DE APROVAÇÃO

Algoritmo Genético Aplicado a Problemas na Economia

Monografia apresentada como requisito à obtenção do grau de Bacharel em Ciências Econômicas pelo Curso de Ciências Econômicas, Setor de Ciências Sociais Aplicada, Universidade Federal do Paraná:

Prof. Dr. João Basilio Pereima
(Orientador/UFPR)

XXXXXXXXXXXXXXXXXXXX
(Membro/UFPR)

XXXXXXXXXXXXXXXXXXXX
(Membro/UFPR)

XXXXXXXXXXXXXXXXXXXX
(Membro/Unicamp)

XXXXXXXXXXXXXXXXXXXX
(Membro/UFSC)

Curitiba, XX de Abril de 2022.

AKNOWLEDGEMENTS

- To my parents, my heroes and ultimate examples of kindness, responsibility and character, for all the love and cherish throughout my entire life.
- To my love, Cassia Lorenza, for the unconditional love and support in difficult times when I had to exchange our talks and caresses for a computer screen. For making me always want to become a better person and specially for these big eyes I have the pleasure to see opening in the morning and closing at night.
- To my advisor, Prof. Dra, whom, through his brilliant expositions on economics and complexity, managed to broaden and renew my view and interest towards economic science.
- To my friends Alan, Augusto and e friendship I had to neglect numerous times in order to achieve knowledge and accomplishment, but even so they never gave up on offering words of encouragement.
- To all the professors on the Department of Economics at Federal University of Parnas second home for so long, and to all my colleagues in PPGDE.
- Finally, to all the hard-working Brazilian tax payers who paid for my scholarship without which I would not be able to become a PhD.

“O importante é vencer, tudo é sempre”
(Ayrton Senna)

“I think the next century (21st) will be the century of complexity”
(Stephen Hawking)

Table of Contents

1	Introdução	2
2	Computação Evolucionária	3
2.1	Introdução	3
2.2	Programação Evolucionária	5
2.3	Estratégias Evolutivas	6
2.4	Algoritmos Genéticos	7
	Bibliography	8
A	Sumário de Terminologia Natural e Terminologia Artificial	12
B	Capítulo 5 Code	14

List of Tables

2.1	METÁFORA DA COMPUTAÇÃO EVOLUCIONÁRIA	3
-----	--	---

List of Figures

RESUMO

ASDFASDFSADFSADF

Palavras-chave:

ABSTRACT

ASDFSADFADAF

Keywords: Innovation. Bandwagon Behaviour. Demand Saturation. Product Life Cycle. Complexity.

Capítulo 1

Introdução

Capítulo 2

Computação Evolucionária

O contexto da criação e desenvolvimento dos algoritmos genéticos (AG) está intrinsecamente ligado à área de estudos da computação evolucionária que, por sua vez, está sob o guarda-chuva da área de pesquisa da inteligência artificial. Com isso, neste capítulo, procura-se realizar uma contextualização histórica à computação evolucionária, à programação evolucionária, às estratégias evolutivas e aos algoritmos genéticos, objetivando, assim, um maior entendimento de como o desenvolvimento das pesquisas nestas áreas contribuíram para o campo dos algoritmos genéticos.

2.1 Introdução

Como o próprio nome sugere, a computação evolucionária (CE) tem como inspiração os processos de evolução observados nos organismos da natureza, sendo uma metáfora computacional que, de forma geral, visa solucionar problemas computacionais ou entender melhor os processos naturais de evolução. Através de uma simulação destes processos naturais, busca-se pelos indivíduos mais aptos a sobreviverem em um ambiente, assim como analisar como os processos de reprodução e mutação destes indivíduos ocorreram. O próprio ambiente é, em si, um dos elementos mais importantes neste conjunto, tendo grande influência nessa luta pela sobrevivência e a busca de parceiros para reprodução e determinando como a capacidade de se adaptar a esse meio influenciará em suas chances de passar seus genes para as próximas gerações.

Tabela 2.1: METÁFORA DA COMPUTAÇÃO EVOLUCIONÁRIA

Evolution		Problem solving
Environment	\iff	Problem
Individual	\iff	Candidate solution
Fitness	\iff	Quality

FONTE: Eiben and Smith (2015)

Ao fim de 1950, e meados da década de 60, a tecnologia havia avançado até chegar à computação digital, o que possibilitou um avanço na experimentação de novos modelos

de processos evolucionários e um grande número de estudos nas décadas seguintes. Os trabalhos de [Friedberg \(1958\)](#), [Friedberg et al. \(1959\)](#) e [Bremermann \(1962\)](#) são apontados como os primeiros registros de desenvolvimento de processos evolucionários aplicados no contexto de problemas computacionais. Os trabalhos de Friedberg podem ser considerados alguns dos primeiros estudos em *machine learning*¹ e programação automática [Back et al. \(1997\)](#).

[Bremermann \(1962\)](#), apresenta um dos primeiros trabalhos de evolução simulada aplicada à otimização linear e convexa, e equações simultâneas não lineares, assim como desenvolve, em 1965², um dos primeiros estudos teóricos sobre algoritmos evolucionários, demonstrando que a mutação ótima deve ter um valor $\frac{1}{l}$ ³ no caso de l bits codificados como indivíduos quando aplicados a problemas de problemas linearmente separáveis [Back et al. \(1997\)](#).

Com as contribuições dos trabalhos acima apresentados, as pesquisas realizadas na segunda metade dos anos 1960 estabeleceram os três principais campos de estudo em computação evolucionária (CE), sendo eles a programação evolucionária (PE), as estratégias evolutivas (EE) e os algoritmos genéticos (AG). Como apresentado por [Back et al. \(1997\)](#), na segunda metade da década de 1960, Lawrance Fogel⁴ construía as bases da programação evolucionária em San Diego, Califórnia, e John Holland fundava as bases dos algoritmos genéticos na Universidade de Michigan⁵. Por sua vez, as estratégias evolutivas eram desenvolvidas por Inge Rechenberg, Peter Bienert e Hans-Paul Schwefel em Berlim em meados de 1965⁶.

Como aponta [Back et al. \(1997\)](#), mesmo com cada uma das áreas seguindo seu próprio caminho de pesquisas ao longo dos quase 30 anos seguintes, a década de 1990 marca o encontro destes campos de estudo através dos esforços de seus pesquisadores na organização de diversos congressos com o objetivo de compartilharem os conhecimentos até então absorvidos, culminando, no início dos anos 1990, no consenso do nome **computação evolucionária** (destacado pelo autor) como o nome dessa nova grande área de pesquisa. A partir destas reuniões, o crescimento do número de interessados e novos trabalhos foi naturalmente crescendo ao longo da década. Em 1993, é criado um periódico homônimo pelo Instituto de Tecnologia de Massachusetts⁷ (MIT). Em 1994, uma das três conferências do Congresso Mundial de Inteligência Computacional organizado pelo Instituto de Engenheiros Eletricistas e Eletrônicos⁸ ([Back et al., 1997](#) apud [Eiben et al., 1994](#)).

¹Do inglês, aprendizado de máquina.

²[Bremermann H J and S](#)

³*length*, do inglês, comprimento

⁴[Fogel et al. \(1966\)](#)

⁵[Holland \(1962\)](#)

⁶[Rechenberg et al. \(1965\)](#)

⁷Evolutionary Computation ([1993](#))

⁸*IEEE World Congress on Computational Intelligence (WCCI)*

2.2 Programação Evolucionária

Desenvolvida por Lawrence Fogel na década de 1960, a programação evolucionária (PE) foi construída sobre diversos experimentos visando a previsão, sob algum critério arbitrário, de séries temporais não estacionárias através da evolução simulada dos estados das máquinas dentro de um limite de estados predeterminados, ou seja, dado os estados passados, previa-se os estados da máquina resultantes deste processo. Fogel buscou seguir um caminho diferente de pesquisa em relação ao que os trabalhos em inteligência artificial se concentrava na época, uma simulação primitiva de redes neurais. Para Fogel, havia uma grande limitação dos modelos baseados na inteligência humana em relação aos processos de criaturas com desenvolvimento contínuo do intelecto, necessário para sobrevivência em um dado ambiente (evolução).

Segundo (Back et al., 1997, pg.A2.3:3), Fogel apresenta as primeiras tentativas de “[...] (i) usar a evolução simulada para realizar previsões, (ii) incluir codificações de comprimento variável, (iii) usar representações que tomam a forma de uma sequência de instrução, (iv) incorporar uma população uma população de soluções candidatas e (v) coevoluir programas evolutivos”, considerando “a inteligência baseada na adaptação do comportamento para atingir metas em uma variedade de ambientes” (tradução nossa)⁹. Devido ao contexto computacional, esses ambientes eram representados através de uma sequência de símbolos ou caracteres de um alfabeto finito arbitrário e o problema evolutivo definido como uma sequência de instruções, ou algoritmo, aplicadas sobre o conjunto de símbolos já observados. Com isso, ao inserir um conjunto de máquinas (população) em um dado ambiente, onde cada máquina possui um valor definido como o valor de entrada, esperava-se melhorar a performance de previsão do algoritmo, à medida que o valor de saída, ou o resultado, era comparado com o próximo valor de entrada. A qualidade desta previsão era, então, medida por uma função de recompensa que indicava o quanto cada máquina da população se adaptou ao ambiente.

Cada máquina pai pode criar um ou mais descendentes, onde cada descendente é criado pelo processo aleatório de alteração de estado, ou valor, do pai. Esse processo de mutação pode ocorrer sob uma certa distribuição de probabilidade ou ser definido no início da implementação do algoritmo. Ao fim de cada geração, a função de recompensa é aplicada sobre o descendente, assim como foi feito com seu pai, para avaliar o quão apto está em relação ao ambiente. As máquinas que fornecem o maior valor de recompensa permanecem no ambiente e se tornam pais das máquinas da geração seguinte. Este processo acontece sucessivas vezes até o símbolo que o símbolo que se deseja prever seja, efetivamente, previsto. A melhor máquina irá gerar essa previsão e esse novo símbolo é adicionado na população para ser avaliado no ambiente, reiniciando, assim, o processo.

Esse algoritmo foi aplicado com êxito em problemas de previsão, identificação e controle automático, simulação de coevolução de populações, experimentos de previsão de sequência, reconhecimento de padrões e em jogos. Na década de 1980, o algoritmo estendeu-se para novas aplicações, como no ordenamento de itens no problema do cai-

⁹[...] (i) use simulated evolution to perform prediction, (ii) include variable-length encodings, (iii) use representations that take the form of a sequence of instructions, (iv) incorporate a population of candidate solutions, and (v) coevolve evolutionary programs [...] considered intelligence to be based on adapting behavior to meet goals in a range of environments.

xeiro viajante e em funções de otimização contínua, evoluindo, posteriormente, para implementações em planejamento de rotas, seleção ótima de subconjuntos e treinamento de redes neurais. No início da década de 1990, ocorre a primeira conferência anual de programação evolucionária, como exemplos de diversas aplicações de otimização na área de robótica, planejamento de caminhos e rotas, desenho e treinamento de redes neurais, controle automática entre outros (Back et al., 1997 apud Eiben et al., 1994).

2.3 Estratégias Evolutivas

No meio da década de 1960, Bienert, Rechenberg e Schwefel, três estudantes da Universidade Técnica de Berlim, estudavam modelos de otimização aplicados em problemas da área de aerotecnologia e tecnologia espacial. Uma das principais pesquisas que realizavam à época, era de um robô experimental que, em um túnel de vento, deveria realizar uma série de testes em uma estrutura tridimensional fina e flexível visando minimizar a resistência em relação ao ar. Os primeiros testes não obtiveram sucesso. Foi apenas no ano seguinte ao início dos testes que Rechenberg et al. (1965) decide utilizar um dado para decisões aleatórias elaborando, assim, a primeira versão de uma EE (Back et al., 1997, pg.A2.3:6) chamada, posteriormente, de $(1 + l)EE$.

Essa primeira versão consiste em uma sequência de instruções projetadas para otimização contínua bastante similar à busca aleatória, exceto por uma regra para a força da mutação conhecida como “regra do sucesso $\frac{1}{5}$ ” para ajuste do desvio padrão dessa mutação. Como a notação sugere, a estratégia de evolução $(1 + l)$ possui apenas um indivíduo pai que irá gerar apenas um indivíduo filho, onde ambos são confrontados e o indivíduo que representa a solução mais fraca, morre. Este indivíduo sobrevivente gera um novo filho e, assim, repetindo essa sequência diversas até se chegar a uma solução ótima. Sendo executado indivíduo a indivíduo, esse processo é computacionalmente custoso, apresentando uma convergência lenta para uma solução ótima, assim como tem a possibilidade de convergir para uma solução local.

Devido aos problemas de desempenho, os autores trabalharam em melhorias na estrutura do algoritmo, desenvolvendo uma nova versão chamada de EE multi-membro¹⁰ com população maior que 1. Novas melhorias foram realizadas nessa nova versão, resultando em dois princípios principais: $(\mu + 1)$ e (μ, λ) . No primeiro, μ indivíduos produzem λ descendentes, gerando uma população temporária de $(\mu + \lambda)$ novos indivíduos, havendo a seleção de μ indivíduos para a geração seguinte. No segundo tipo, μ indivíduos produzem λ descendentes, com $\mu < \lambda$, onde a nova população de μ indivíduos possui apenas os indivíduos selecionados do conjunto de λ descendentes, limitando o tempo de vida de um indivíduo apenas a uma geração específica.

A partir da primeira versão, a comunidade de pesquisadores da área de EE realizaram novas aplicações nas décadas seguintes que não se reduziram somente ao objetivo da otimização de valores do mundo real, como a aplicação para otimização binária em estruturas de indivíduos multicelulares usando a ideia de sub populações, estratégias evolutivas para problemas com multi critérios, entre diversas outras aplicações seguindo

¹⁰ *EE multimembered.*

a ideia principal de melhoria contínua dos indivíduos analisados (Back et al., 1997).

2.4 Algoritmos Genéticos

Conforme aponta (Back et al., 1997, pg.A2.3:4), as primeiras ideias que deram origem aos algoritmos genéticos datam do início da década de 1960 nos trabalhos de Holland (1962), que “estabeleceu uma agenda ampla e ambiciosa para compreender os princípios subjacentes dos sistemas adaptativos – sistemas que são capazes de automodificação em resposta às suas interações com os ambientes em que devem funcionar” (tradução nossa)¹¹.

Diferentemente dos estudos apresentados anteriormente de algoritmos aplicados em modelos de previsão ou otimização, Holland se debruçou sobre modelos evolutivos para entendimento de sistemas adaptativos robustos naturais e projeção de elementos adaptativos em um dado contexto. Para o autor, em sistemas adaptativos naturais, as características relativas à competição entre os agentes e de inovação ao longo destes processos naturais eram fundamentais para que os indivíduos se adaptassem ao ambiente e às mudanças imprevistas que este ambiente aplicava sobre esses indivíduos Back et al. (1997).

Conforme apresentando por (Goldberg, 1989, p. 1), eram dois os principais objetivos de John Holland e seus colegas no campo de pesquisa dos AGs, sendo eles: uma explicação bem estruturada e fundamentada dos processos de adaptação de sistemas naturais e a construção de programas computacionais de sistemas artificiais com a finalidade de incorporar importantes mecanismos destes sistemas, sendo o foco da pesquisa a robustez dos algoritmos, ou seja, o equilíbrio entre a eficiência e a eficácia necessária para a sobrevivência de possíveis soluções em muitos ambientes diferentes.

O grande diferencial da linha desenvolvida por Holland, foi a incorporação de diversos conceitos da genética que se demonstraram de alta eficiência e performance na resolução de problemas complexos utilizando poucos dados de entrada, assim como os processos de busca para encontrar soluções ótimas apresentavam inovações em relação à resolução de tais problemas e aprendizados dos elementos no ambiente ao longo dos processos evolutivos. Os elementos a serem evoluídos ao longo de um período eram representados como genomas (conjunto de todos os genes de um ser vivo) e os mecanismos de evolução como abstrações de operadores genéticos como a reprodução, cruzamento e mutação.

Em 1967, Holland realiza estudos em uma teoria geral de sistemas adaptativos, desenvolvendo no mesmo período a análise de esquemas de sistemas adaptativos e, em 1969, demonstra aplicações de alocação ótima utilizando o modelo de k-bandidos armados (Holland apud Back et al., 1997). Em 1970, Cavicchio (1970) absorveu essas ideias como uma forma de busca adaptativa e as testou em problemas de busca complexa envolvendo soluções de subrotinas e reconhecimento de padrões, assim como alguns dos primeiros estudos sobre formas elitistas de seleção e adaptação de taxas de cruzamento

¹¹Holland set out a broad and ambitious agenda for understanding the underlying principles of adaptive systems—systems that are capable of self-modification in response to their interactions with the environments in which they must function.

e mutação. [Hollstien \(1971\)](#), por sua vez, apresentou ideias aprofundadas de seleção alternada e, através de experimentações em paisagens de adaptação bidimensionais, testou diversos modelos de estratégias de reprodução com origem em técnicas utilizadas por criadores de animais. Em 1975, Holland agrupa as ideias desenvolvidas em seus trabalhos e publica o seu maior trabalho, o livro *Adaptação em Sistemas Naturais e Artificiais* [Holland \(1975\)](#). No mesmo ano, [Jong and Alan \(1975\)](#) publica experimentos demonstrando teórica e praticamente os efeitos no tamanho da população, cruzamento e mutação de AGs aplicados em uma população, seguindo as ideias de Holland.

O interesse pela área foi crescendo progressivamente nas décadas seguintes. Em 1976, pesquisadores da Universidade de Michigan, Universidade de Pittsburgh, entre outras, organizaram a primeira conferência de sistemas adaptativos, que ocorreu nos anos seguintes. Em 1979, Holland, De Jong e Sampson escalam o tamanho da conferência através de um financiamento para realizarem uma conferência interdisciplinar em sistemas adaptativos, que acabou sendo realizado em 1981 na Universidade de Michigan. Em 1985, na Universidade de Pittsburgh, ocorre a Conferência Internacional sobre Algoritmos Genéticos (ICGA) que, devido ao sucesso, passou a ser semestral nos anos seguintes. Em 1989, surge a Sociedade Internacional de Algoritmos Genéticos (ISGA), organização responsável pelo financiamento de conferências e atividades das áreas de pesquisas relacionadas aos AGs, tendo como uma de suas primeiras conquistas a criação de uma das principais conferências da comunidade, sobre os Fundamentos dos Algoritmos Genéticos (FOGA) ([Back et al., 1997](#), pg.A2.3:5).

References

- (1993). Editorial Introduction. *Evolutionary Computation*, 1(1):iii–v.
- Back, T., Fogel, D. B., and Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*. IOP Publishing Ltd., GBR, 1st edition.
- Bremermann, H. J. (1962). Optimization through evolution and recombination. page 12.
- Bremermann H J, R. M. and S, S. Search by evolution biophysics and cybernetic. page 157–67.
- Cavicchio, D. J. (1970). Adaptive search using simulated evolution. Accepted: 2006-02-03T18:12:14Z.
- Eiben, A., Hinterding, R., and Michalewicz, Z. (1994). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*.
- Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Berlin, Heidelberg.
- Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Artificial intelligence through simulated evolution.
- Friedberg, R. M. (1958). A learning machine: Part i. *IBM Journal of Research and Development*, 2(1):2–13.
- Friedberg, R. M., Dunham, B., and North, J. H. (1959). A learning machine: Part ii. *IBM J. Res. Dev.*, 3:282–287.
- Goldberg, V. A. P. o. H. D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, Reading, Mass, 13th ed. edição edition.
- Holland, J. Nonlinear environments permitting efficient adaptation.
- Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *J. ACM*, 9(3):297–314.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Adaptation in

natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, Oxford, England. Pages: viii, 183.

Hollstien, R. B. (1971). *Artificial genetic adaptation in computer control systems*. OCLC: 726084376.

Jong, D. and Alan, K. (1975). Analysis of the behavior of a class of genetic adaptive systems. Accepted: 2006-02-03T19:05:13Z.

Rechenberg, I., Toms, B., and Establishment, R. A. (1965). *Cybernetic Solution Path of an Experimental Problem.*. Library translation / Royal Aircraft Establishment. Ministry of Aviation.

Appendices

Appendix A

Sumário de Terminologia Natural e Terminologia Artificial

Terminologia Natural	Terminologia Artificial	Terminologia Artificial (Inglês)	Descrição Curta
Locus	Locus	<i>Locus</i>	Posição de um gene (ou característica) dentro de uma cadeia de caracteres.
Alelo	Alelo	<i>Allele</i>	Valor do gene (ou característica), sendo 0 ou 1.
Gene	Característica	<i>Gene</i>	Característica, caractere ou algoritmo de uma cadeia de caracteres.
Bloco de construção	Bloco de Construção	<i>Building Block</i>	Subvetor de alto desempenho. Responsável por formar novos vetores.
Cromossomo ou Indivíduo	Vetor ou Cadeia de caracteres	<i>String</i>	Conjunto de características agrupadas através de um conjunto de caracteres. Cada matriz pode ser uma possível solução para um problema ou fazer parte de uma solução global.
Genótipo	Estrutura	<i>Structure</i>	Conjunto de vetores que mantém uma carga ou estrutura genética.
Fenótipo	Estrutura decodificada	<i>Decoded structure</i>	Nova carga ou estrutura genética após as alterações das características principais de uma
População.	População	<i>Population</i>	Conjunto de indivíduos construídos, inicialmente, de forma aleatória, onde o AG irá realizar suas buscas. Cada geração é composta por uma população que pode ser uma possível solução para o problema de otimização.
Geração	Iteração	<i>Iteration</i>	Formação de uma nova população após a aplicação dos operadores de reprodução, cruzamento e mutação.
Reprodução ou Seleção	Operador de Reprodução	<i>Reproduction</i>	Seleção dos indivíduos com maior aptidão ao ambiente.
Cruzamento	Operador de Cruzamento	<i>Crossover</i>	Troca de genes, ou características, entre os indivíduos de uma população.
Mutação	Operador de Mutação	<i>Mutation</i>	Processo aleatório de alteração de uma ou mais características de um indivíduo.
Valor de Aptidão	Valor de Aptidão	<i>Fitness value</i>	Quantificação da qualidade de um indivíduo ou sua aptidão em relação ao ambiente.
Paisagem ou Horizonte de aptidão	Paisagem de Aptidão	<i>Fitness landscape</i>	Espaço pelo qual o AG faz sua busca por soluções, ou seja, diz respeito ao problema a ser resolvido.
Função de aptidão	Função objetivo	<i>objective function</i>	Função contendo os parâmetros para resolução do problema.

Appendix B

Capítulo 5 Code

```
def teste_func():  
    return ...
```