

# Evolutionary Computing

Matheus Barros, Jorás Custódio, Rentao Falcão, Diogo Duarte

*Inspur 4<sup>o</sup> semestre, São Paulo SP*

---

## Abstract

Neste paper é possível observar algumas abstrações de Evolutionary computing incluindo Swarm Intelligence para o processamento de dados, Algoritmos Genéticos que seguem a linha do evolucionismo de Darwin, Neuro Co-evolução Coletiva aplicadas para enxames de robos e predição de mercado financeiro em uma série temporal.

*Keywords:* Evolutionary Computing, Swarm Intelligence, Feature Selection, Big data,

---

## 1. Abordagens de inteligência de enxame para a seleção de recursos em Mineração de Dados

### 1.1. Abstract

[1] Swarm intelligence é um método reconhecido pela sua simplicidade e  
5 com grande potencial de pesquisa extensa em áreas aplicadas. A questão a ser  
tratada é Big Data, cujo problema advém de sua dimensionalidade e grande  
número de features, o que dificulta a tarefa de machine learning especialmente  
com algoritmos de classificação. Para contornar essa questão é realizado o Fea-  
ture Selection [2] que seleciona pequenos subsets com features informativos, no  
10 entanto não é uma tarefa fácil devido à extensão e complexidade do espaço de  
pesquisa. O intuito desta seção do paper é fazer um highlight de diferentes  
métodos de inteligência de enxame para a realização de seleção de recursos na  
mineração de dados.

---

<sup>1</sup>Camada Física

## 1.2. Introduction

15 Há algumas questões a serem tratadas em tarefas de classificação que com feature selection melhora o seu desempenho de performance. Como por exemplo a redução de ruídos, remoção features redundantes e também na retirada de features irrelevantes que podem afetar o resultado final do modelo. Tendo como benefício além da performance de conhecimento a economização de recursos e  
20 tornando o classificador mais simples e fácil de entender [3].

Evolutionary Computation (EC) é uma família de técnicas de otimização baseados em população, Swarm Intelligence (SI) é um braço de EC que consiste em algoritmos inspirados pelo comportamento de animais ou insetos sociais. Alguns métodos de SI bem representativos são: particle swarm optimization  
25 (PSO) [4]; ant colony optimization (ACO) [5]; artificial bee colony optimization (ABC) [6]. Nesses algoritmos a população consistente de um número de indivíduos explora e compartilha seu conhecimento sobre o espaço de busca com outros membros ajudando o enxame inteiro a melhorar seu posicionamento no espaço de busca, convergindo em um objetivo final ótimo [4]. Em comparação  
30 com Genetic Algorithms (GA) [7] os algoritmos SI em geral convergem mais rápido e tem melhor performance quando o custo operacional é baixo.

É válido ressaltar para esclarecimentos mais adiante que há duas maneiras diferentes de representar feature selection que consiste em uma representação básica como um vetor contínuo e a representação binária, no entanto não vão  
35 ser evidenciados neste paper. Com isso feature selection baseados em SI são divididos em duas categorias principais classificadas por objetivos singulares e múltiplos de algoritmos já que os mecanismos de busca variam quando tem que lidar com diferentes números de objetivos.

## 1.3. Inteligência de Enxame para Seleção de Recursos

### 40 1.3.1. Particle Swarm Optimization

Proposto por Eberhart e Kennedy em 1995[8] o PSO foi feito para otimizações contínuas de problema. O enxame de partículas é simultâneo, onde cada partícula representa uma solução possível. A partícula guarda a sua melhor solução

(pbest) e também é guardado a melhor solução descoberta pelo enxame (gbest) até o dado momento. As melhores duas posições são esperadas de levar o enxame a explorar novas áreas de busca. A posição de cada partícula é representada por um vetor. Algumas aplicações que foram implementadas com PSO são: mineração de texto, transmissão de dados, análise de imagem, medicina.

### 1.3.2. *Artificial Bee Colony Optimization*

Inicialmente havia sido implementado para otimizações numéricas em 2005 por Kraboga [6] foi proposto o algoritmo de enxame de abelha ABC. A representação da solução de cada candidato é exposta como uma fonte de alimento. Em um enxame de abelha há três tipos de abelhas, as employed bees, onlooker bees e scout bees. Em cada saída de alimento há uma employed bee se certificando de melhorar a qualidade da comida pesquisando as fontes de alimentos das redondezas. O diferencial de onlooker bees é que diferente das employed eles são mais suscetíveis a procurar fontes de maior qualidade. Se uma fonte de comida não for melhorada em um número de iterações a abelha que era encarregada se torna uma scout bees que seleciona uma nova fonte de comida randomizada.

### 1.3.3. *Ant Colony Optimization*

Proposto por Dorigo e Di Carlo [5] em 1999, o ACO foi desenvolvido pra resolver problemas discretos de otimização denotado por módulos. Cada formiga se move próximo às bordas conectando módulos adjacentes para construir uma solução completa otimizando o problema gradualmente. Com uma função de adequação a solução é avaliada. O positivo da solução é utilizado para atualizar a informação (chamado de feromônio) na borda em que a solução foi usada. Consequentemente uma solução boa contém uma quantidade maior de feromônio. As formigas eram então atraídas por esses caminhos com soluções de feromônio melhores, sendo que essas mesmas formigas criam novos caminhos com a promessa de novas soluções.

## 2. Modelo híbrido de teoria do caos, perceptron multicamadas e algoritmos evolucionários, para predição de mercado financeiro em uma série temporal

### 75 2.1. Abstract

Os preços de ativos financeiros são muito dinâmicos e não lineares, o que torna complexa a predição. Por isso, a busca por modelos para realizar estimativas a partir de um conjunto de dados continua sendo um grande desafio. Dentre os métodos de predição, surgiram alguns modelos promissores  
80 de “deep learning”, mas que não alcançaram resultados notórios. No entanto, um grande número de pesquisas apontam para uma relação entre a teoria do caos e o mercado financeiro. Desta forma, propõe-se um modelo híbrido de três estágios: (1) Uso da teoria do caos para a construção de um espaço fásico; (2) Aplicação do modelo de rede neural artificial de Perceptron Multicamadas; e  
85 (3) Aplicação do algoritmo evolucionário Multi-Objective Particle Swarm Optimization (MOPSO) ou Non-dominated Sorting Genetic Algorithm (NSGA-II), para testar qual demonstra melhores resultados. Em ambos os modelos híbridos, o terceiro estágio aprimorou o resultado produzido pelo segundo estágio.

### 2.2. Introduction

90 A predição trabalha com séries temporais, que no caso do mercado financeiro é uma coleção de observações cronologicamente ordenadas acerca de ativos financeiros e outras variáveis relevantes[9]. As séries temporais de ativos financeiros possuem algumas características que tornam sua previsão altamente complicada como:

- 95 1. Séries temporais de ativos financeiros se comportam como um passeio aleatório, o que torna sua predição, de um ponto de vista teórico, impossível [10].
2. Séries temporais de ativos financeiros comumente possuem muitos ruídos e anomalias [11].

100 3. Todas as propriedades estatísticas relacionadas ao mercado financeiro variam com o tempo [10].

No entanto, métodos de predição tradicionais, como por regressão por exemplo, apesar de serem poderosos no ramo da estatística, não renderam bons resultados. Com o desenvolvimento de algoritmos mais inteligentes de “deep learning” e algoritmos evolucionários, estes foram propostos para tentativa de  
105 predição de ativos financeiros e percebeu-se que os modelos que combinavam diferentes modelos preditivos, chamados híbridos, renderam melhores resultados do que se empregados individualmente. A teoria do caos, introduzida por Henri Poincaré [12, 13], apresenta uma nova forma de se modelar comportamentos dinâmicos não lineares de um sistema complexo ao incorporar a série  
110 temporal em seu espaço fásico correspondente, usando os devidos parâmetros. O perceptron multicamadas é um modelo de rede neural artificial capaz de lidar bem com relações não lineares complexas. Com estes conhecimentos, propõe-se dois modelos híbridos de três estágios para predição de ativos financeiros em uma série temporal. No primeiro modelo híbrido, o caos no dataset é modelado  
115 utilizando-se dos menores parâmetros possíveis (estágio 1). A série temporal resultante alimenta o modelo de perceptron multicamadas (estágio 2). Então o residual deste processo é checado pela presença de caos. Se houver caos, ele é modelado e sua série temporal alimenta o modelo de auto-regressão MOPSO (estágio 3). Por fim, os resultados do estágio 3 são somados algebricamente aos  
120 resultados do estágio 2 para se obter as predições otimizadas. O segundo modelo híbrido é igual ao primeiro, apenas substituindo o modelo de auto-regressão do terceiro estágio para o modelo de auto-regressão NSGA-II.

### 2.3. *Desing para experimentação*

125 Para verificar a performance dos modelos híbridos, são utilizados três indicadores: Erro Quadrático Médio, Dstat (mudança direcional estatística) [14] e o Coeficiente de Desigualdade de Thiel. Para comparação com os modelos híbridos propostos foram escolhidos modelos que não possuem o terceiro estágio, ou seja, que utilizam apenas da teoria do caos e de redes neurais artificiais [15, 16]. Como

130 os autores destes modelos os utilizaram apenas na predição das taxas de câmbio monetário de JPY/USD, GBP/USD e EUR/USD, reportando apenas os valores de Erro Quadrático Médio, a comparação com o modelo de três estágios proposto se dará pela comparação dos resultados nestas três séries.

#### 2.4. Resultados

135 Ao realizar a comparação dos resultados alcançados, foi possível determinar que o segundo modelo híbrido proposto (teoria do caos + Perceptron Multicamada + NSGA-II) superou todos os outros modelos, rendendo previsões muito próximas do valores reais coletados, sobrepondo a curva da taxa de câmbio monetária em cada uma das séries temporais utilizadas.

### 140 3. Evolving behavioral specialization in robot teams to solve a collective construction task

#### 3.1. Introdução

Esta seção irá analisar o artigo ‘Evolving behavioral specialization in robot teams to solve a collective construction task’ [17]. O foco estará na análise dos métodos de neuro co-evolução coletiva (Collective Neuro Co-Evolution - 145 CONE), o artigo escrito por Nitschke *et al.* utilizou CONE para fazer o design automatizado de métodos de controle para equipes de robôs simulados, com o objetivo de aumentar a performance em resolução de tarefas coletivas. Os robôs definidos na pesquisa de interesse são definidos como: cooperativos, heterogêneos, cientes e fracamente coordenados. Cooperativos se refere ao fato que 150 os robôs simulados devem cooperar para realizar uma tarefa, heterogêneos se refere ao fato que cada robô é inicializado e evolui comportamentos diferentes, cientes refere-se ao fato que cada robô está ciente das ações tomadas pelos outros robôs e as equipes são fracamente coordenadas pois não foi especificado nem 155 predefinido nenhum protocolo para a cooperação. As equipes devem completar tarefas de aquisição de recursos e construção de estruturas com tais recursos. Tarefas de coletas requerem que os agentes (robôs individuais) dividam seu trabalho para maximizar a quantidade de recursos adquiridos. Enquanto as tarefas

de construção requer que agentes coloquem blocos em uma zona de construção  
160 com uma ordem específica para construir uma estrutura, com a restrição que  
equipes devem construir múltiplos objetos simultaneamente.

### 3.2. CONE - Co-evolução neural coletiva

A co-evolução neural coletiva é um processo de design de controladores au-  
tomatizado, com o objetivo de usar a co-evolução coletiva para adaptar uma rede  
165 neural artificial (ANN - artificial neural network) para controlar uma equipe de  
agentes, no artigo escrito por Schut *et al.* foi utilizada uma feedforward ANN,  
ou seja uma rede neural onde as conexões não formam um ciclo [18], cada rede  
neural começa inicializada com 3 camadas de neurônios: uma camada de en-  
trada, uma de saída e 1 camada escondida, que comunica entre a entrada e  
170 saída. Para cada evolução da rede neural, os controladores são avaliados, os  
pesos de cada neurônio são alterados, e uma nova camada é gerada. O intuito  
da avaliação é manter os pesos considerados mais aptos, este processo é dire-  
cionado pelos mecanismos de cooperação e competição entre as populações de  
robôs. Uma parte integral do processo está na avaliação de especialização de  
175 cada controlador, a métrica escolhida por Eiben *et al.* foi o número de vezes  
que cada controlador troca entre execução de ações motoras distintas (ações)  
durante seu tempo de vida, definida como S, especialização foi definida nos ca-  
sos onde controladores demonstraram uma pequena preferência para uma ação  
específica. A métrica S foi calculada com a seguinte equação:  $S = A \div N$  onde  
180 A define o número de vezes que um controlador troca de ações e N define o  
número de ações possível, um valor de S próximo de zero indica um alto teor de  
especialização, para o estudo sendo revisto. O limite de S para um controlador  
ser considerado especializado foi definido como 0.5, Caso um controlador seja  
considerado especializado, este recebe um rótulo apropriado a sua preferência de  
185 ação, caso múltiplos controladores recebam o mesmo rótulo, estes são agrupados  
de acordo.

### 3.3. Métodos para a co-evolução

CONE utiliza duas métricas principais para a recombinação de genótipos: Métricas de diferença de genótipo (Genotype Difference Metrics - GDM) e Métricas de diferenças de especialização (Specialization Difference Metrics - SDM), estas métricas foram usadas para regular a recombinação genética dentro e entre as populações de agentes baseado nas similaridades genéticas e o sucesso as especializações evoluídas. O propósito da métricas GDM e SDM é de regular e adaptar as recombinações de genótipos entre as populações de controladores em função de suas aptitudes e progressos, para regulamentar este processo, dois algoritmos dinâmicos foram criados, ambos recebendo como input um float que varia de [0.0 1.0], os parâmetros são: Limite de Similaridade Genética (Genetic Similarity Threshold - GST) e Limite de Similaridade de Especialização (Specialization Similarity Threshold - SST). Estes valores de entrada foram inicializados com valores aleatórios dentro do alcance aceitável previamente mencionado. A regulamentação do processo de recombinação é uma parte essencial do processo de co-evolução neural cooperativa, pois os controladores não podem ser similares ou diferentes demais uns dos outros, para que o comportamento dos controladores se agregue de forma benéfica à tarefa a ser completada. com isso em mente foram criadas duas principais métricas para regular a recombinação genética. GDM é uma métrica abrangente que regula a recombinação de genes similares em populações diferentes, este processo é ajustado ao longo do experimento. Dados dois genótipos quaisquer, A e B, estes são considerados similares se a diferença entre A e B for menor que o GST estabelecido no controlador. O valor do GST é alterado baseado na recombinação de genótipos entre populações é alterada a partir do número de recombinações prévias e da aptitude dos controladores, duas regras foram estabelecidas para alterar o valor do GST, e portanto alterar o GDM:

1. Se o número de recombinações aumentou, porém a aptitude se manteve estável ou diminuiu, deve-se diminuir o GST, para restringir o número de recombinações.



2. Se o número de recombinações diminui e a aptitude se manteve estável ou diminuiu, deve-se aumentar o GST, para aumentar o número de recombinações.

220 Assim como o GDM, o SDM também regula a recombinação genética de forma integral, porém o SDM foi criado para garantir que apenas genótipos de especializações similares se combinem, com isto em mente, dois controladores são considerado similares, se e somente se:

1.  $S(ANNi) - S(ANNj) < SST$  S sendo o grau de especialização da rede neural (ANNi) e (ANNj)
- 225 2. Se  $ANNi$  e  $ANNj$  tem o mesmo rótulo de especialização

O SST é definido como uma função das similaridades de especializações de uma equipe e da atitude média da equipe, as seguintes regras foram usadas para alterar o valor de SST, e portanto o SDM:

- 230 1. Se o S do controlador aumentou nas últimas n gerações, porém a aptitude se manteve igual ou diminuiu, deve-se diminuir o SST, para restringir as combinações entre equipes.
2. Se o S do controlador diminuiu nas últimas n gerações, e a aptitude se manteve igual ou diminuiu, deve-se aumentar o SST, para incentivar as
- 235 combinações entre equipes.

A Co-evolução neural coletiva tem 8 passos iterativos, nesta seção faremos uma breve descrição deles:

1. Inicialização: n populações são iniciadas, com i subpopulações, cada subpopulação contém m genótipos e cada genótipo contém neurônios designados em uma posição j na camada escondida da ANN da população.
- 240 2. Avaliação de todos os genótipos: cada genótipo de cada subpopulação é avaliado como um controlador completo, esta avaliação é feita comparativamente com n -1 neurônios, aleatoriamente selecionados de cada subpopulação, de todas as outras populações

- 245 3. Avaliação dos controladores de 'Elite': para cada população, deve-se con-  
struir um controlador de 'Elite', um controlador é considerado de 'Elite'  
a partir da atitude de cada subpopulação. Grupos de controladores de  
Elite são avaliados juntos em simulações até todos os genótipos terem sua  
aptitude calculada, este cálculo vem da performance na simulação, este  
250 cálculo sobrescreve qualquer cálculo anterior.
4. Seleção de pais: Se os controladores de Elite mais aptos têm um valor de  
SST que considerem ambos similares, estes irão recombinar seus genes para  
gerar novas subpopulações. Além dos controladores de elite, se o SDM de  
duas subpopulações for adequado, ambas também irão recombinar seus  
255 genes.
5. Recombinação: Durante a recombinação, pares de genótipos de elite de  
cada subpopulação são selecionados baseados em aptitudes. Para a re-  
combinação usa-se o algoritmo de one-point crossover, onde parte dos  
neurônios de elite das camadas escondidas é selecionada aleatoriamente  
260 de cada pai, e trocadas entre si, gerando novos filhos com informações  
genéticas variadas baseadas nos pais.[18]
6. Mutações: As mutações ocorrem através de burst mutation with a cauchy  
distribution [19] aplicada em cada gene de cada genótipo, ou seja, uti-  
lizando os neurônios dos controladores definidos como 'Elite' como base,  
265 cria-se novos neurônios nas novas subpopulações com informações genéticas  
das elites somadas a 'ruídos', com o intuito de aumentar a diversidade  
neural, o ruído é gerado a partir das diferenças entre os neurônios dos  
controladores de Elite.
7. Adaptação por parâmetros: se a aptidão dos controladores de Elite não  
270 tiver aumentando em: n gerações (definido pelo GST) ou w gerações  
(definido pelo SST), deve-se adaptar um dos dois parâmetros citados  
acima.
8. Condição final: Repita os passos 2 a 7 até se atingir a performance ou o  
comportamento coletivo esperado.

## 275 4. Bibliography

### References

- [1] B. H. Nguyen, B. Xue, M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, *Swarm and Evolutionary Computation* 54 (2020) 100663. doi:<https://doi.org/10.1016/j.swevo.2020.100663>.  
280 URL <https://www.sciencedirect.com/science/article/pii/S2210650219303104>
- [2] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, *CoRR* abs/1601.07996. arXiv:1601.07996.  
285 URL <http://arxiv.org/abs/1601.07996>
- [3] B. Xue, M. Zhang, W. Browne, X. Yao, A Survey on Evolutionary Computation Approaches to Feature Selection doi:10.26686/wgtn.14214497.v1.  
URL [https://openaccess.wgtn.ac.nz/articles/journal\\_](https://openaccess.wgtn.ac.nz/articles/journal_contribution/A_Survey_on_Evolutionary_Computation_Approaches_to_Feature_Selection/14214497)  
290 [contribution/A\\_Survey\\_on\\_Evolutionary\\_Computation\\_Approaches\\_](https://openaccess.wgtn.ac.nz/articles/journal_contribution/A_Survey_on_Evolutionary_Computation_Approaches_to_Feature_Selection/14214497)  
[to\\_Feature\\_Selection/14214497](https://openaccess.wgtn.ac.nz/articles/journal_contribution/A_Survey_on_Evolutionary_Computation_Approaches_to_Feature_Selection/14214497)
- [4] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of ICNN'95 - International Conference on Neural Networks*, Vol. 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.
- [5] H. Li, S. Wang, M. Ji, An improved chaotic ant colony algorithm, in: J. Wang, G. G. Yen, M. M. Polycarpou (Eds.), *Advances in Neural Networks – ISNN 2012*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 633–640.  
295
- [6] D. Karaboga, An idea based on honey bee swarm for numerical optimization, technical report - tr06, Technical Report, Erciyes University.  
300

- [7] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., USA, 1989.
- [8] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory,  
 305 in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43. doi:10.1109/MHS.1995.494215.
- [9] V. Ravi, D. Pradeepkumar, K. Deb, Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms, Swarm and Evolutionary Computation 36 (2017)  
 310 136–149. doi:https://doi.org/10.1016/j.swevo.2017.05.003.  
 URL <https://www.sciencedirect.com/science/article/pii/S2210650217300822>
- [10] T. Hellström, K. Holmström, Predictable patterns in stock returns.
- [11] M. Magdon-Ismael, A. Nicholson, Y. Abu-Mostafa, Financial markets: very  
 315 noisy information processing, Proceedings of the IEEE 86 (11) (1998) 2184–2195. doi:10.1109/5.726786.
- [12] H. Poincaré, Sur le problème des trois corps et les équations de la dynamique, Acta mathematica 13 (1) (1890) A3–A270.
- [13] G. Williams, Chaos theory tamed, CRC Press, 1997.  
 320
- [14] J. Yao, C. L. Tan, A case study on using neural networks to perform technical forecasting of forex, Neurocomputing 34 (1) (2000) 79–98. doi:https://doi.org/10.1016/S0925-2312(00)00300-3.  
 URL <https://www.sciencedirect.com/science/article/pii/S0925231200003003>  
 325
- [15] D. Pradeepkumar, V. Ravi, Forex rate prediction using chaos, neural network and particle swarm optimization (2014) 363–375.

- [16] D. Pradeepkumar, V. Ravi, Forex rate prediction: A hybrid approach using chaos theory and multivariate adaptive regression splines (2017) 219–227.
- 330 [17] G. Nitschke, M. Schut, A. Eiben, Evolving behavioral specialization in robot teams to solve a collective construction task, Swarm and Evolutionary Computation 2 (2012) 25–38. doi:<https://doi.org/10.1016/j.swevo.2011.08.002>.  
URL <https://www.sciencedirect.com/science/article/pii/S2210650211000423>
- 335
- [18] A. E. Eiben, Introduction to evolutionary computing, Springer, 2015.
- [19] F. J. Gomez, Robust non-linear control through neuroevolution, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin (2003).
- 340 URL <http://nn.cs.utexas.edu/?gomez:phd03>