

## Clustering por ES, EP e variantes

Gabriel Arruda Evangelista

### I. INTRODUÇÃO

*Clustering*, ou análise de *clusters*, consiste no problema de agrupamento de dados pela sua similaridade visando buscar uma organização de padrões em função de suas características (*features*) e é interpretado como classificação por aprendizado não supervisionado [1]. O presente trabalho propõe a análise do desempenho de alguns algoritmos genéticos (GA) para realização desta tarefa.

Foram estudadas versões tradicionais dos algoritmos *Evolution Strategies* (ES) e *Evolutionary programming* (EP) conforme descrito em [2], além de duas variantes do ES e uma do EP. Foi realizada uma busca em *grid* de conjuntos de parâmetros para cada algoritmo e uma análise estatística dos desempenhos, em termos de Taxa de Sucesso (SR), número de avaliações da função custo até a primeira solução (AES) e melhor aptidão média (MBF), obtidos para resolução do *clustering* para conjuntos similares de dados com duas *features*.

### II. TEORIA

Algoritmos genéticos em linhas gerais lançam mão da mesma ideia para seu desdobramento: populações de indivíduos em um mesmo ambiente competem. Essa competição gera a seleção de sobreviventes e passagem de suas características para gerações seguintes. Esse processo, por sua vez causa o aumento da aptidão dos indivíduos, uma função de qualidade que se deseja maximizar, até que se tenha um indivíduo que represente a solução do problema proposto [2].

As variantes de algoritmos genéticos propostas apresentam representações de indivíduos por valores reais que mapeiam as posições dos centroides no  $\mathbb{R}^2$  no problema proposto. A variação dos genes dos indivíduos tanto pela recombinação quanto pela mutação ao longo das gerações acabam maximizando a aptidão, de forma que se obtenha um conjunto de centroides que melhor agrupa os dados passados como entrada.

A função aptidão foi calculada como no algoritmo *Deterministic Annealing* (DA) para  $T = 1$  apresentado no conjunto de equações 1 [3], porém com o sinal invertido dada a diferença da natureza dos algoritmos, maximização para os GA e minimização para o DA.

$$\begin{aligned} p_{y|x} &= \exp(-d_{xy}/T) \\ Z_X &= \sum_y p_{y|x} \\ J &= \frac{T}{NC \cdot \log Z_X} \end{aligned} \quad (1)$$

Onde  $x$  representa os dados do vetor de dados,  $y$  os centroides,  $d_{xy}$  é a distância de cada ponto do vetor de dado a cada centroide,  $p_{y|x}$  é a probabilidade de pertencimento de cada ponto  $x$  a cada centroide  $y$ , e  $NC$  é o número de *clusters*.

### III. EXPERIMENTOS

#### A. Problema analisado

Os algoritmos foram analisados visando realizar o *clustering* de dados de duas *features*, podendo ser representados no  $\mathbb{R}^2$ . Para a criação dos vetores de dados, inicialmente eram gerados pontos centrais dos *clusters* de maneira aleatória com uma gaussiana centrada na origem e de desvio padrão igual ao número de *clusters*. A partir desses centroides eram gerados os demais pontos, também com distribuições gaussianas, agora com desvio padrão unitário. Dessa forma para cada lançamento eram resolvidos diferentes vetores de dados porém de características controladas. Uma estimativa do

$J_{obj}$  foi calculada em função dos pontos centrais dos *clusters* com o cálculo apresentado em 1. O critério de sucesso para que se considerasse uma solução válida para o problema era que a melhor aptidão atingisse o  $J_{obj}$  com uma tolerância de 5%.

O número de *clusters* foi variado para verificar o comportamento dos algoritmos com o aumento da complexidade do problema. O problema foi exposto aos algoritmos com  $NC \in \{5, 10, 20, 30\}$ .

Visando indicar uma régua comum para análise das aptidões finais alcançadas por cada algoritmo, definiu-se como limite computacional  $5 \cdot 10^5$  avaliações da função custo. A cada geração, para os parâmetros testados, os algoritmos não executavam mais do que poucas centenas de avaliações, número pequeno em relação ao limite computacional. Por isso, não houve preocupação para o tratamento com populações fracionadas ao final das execuções, fazendo com que esse limite fosse ligeiramente extrapolado para conclusão das últimas iterações. No entanto, devido a baixa relevância numérica, e por se entender que os algoritmos já estavam em fase de *exploitation*, onde há pouco avanço em ganho de aptidão, os erros causados por este excesso de avaliações foram considerados irrelevantes.

#### B. Algoritmos empregados

Como falado na seção anterior, os algoritmos estudados empregam sempre a representação por valores reais. Além disso é comum a todos o emprego da autoadaptação não correlacionada de passo múltiplo para a atualização na etapa de mutação. No entanto, as variantes apresentam as seguintes características particulares:

1) *ES*: Esta variante de algoritmo genético clássica apresenta as características apresentadas em I.

Tabela I: Características do ES

Recombinação	X discreto e $\sigma$ linear
Mutação	Perturbação gaussiana não correlacionada de passo múltiplo
Seleção de pais	Aleatória uniforme
Seleção de Sobreviventes	Elitismo determinístico ( $\mu, \lambda$ )

2) *ES memético (ESmeme)*: A variante memética do ES foi proposta em função de outro algoritmo típico para solução de *clustering* o *Generalized Lloyd Algorithm* (GLA). Esse é um algoritmo determinístico que calcula a posição dos centroides na próxima iteração em função de suas distâncias aos dados. O memetismo implementado realiza a seleção dos sobreviventes periodicamente com base nas aptidões de seus vizinhos. Esses são calculados após realizado determinado número de iterações do GLA, conforme esquema proposto por Baldwin para ocorrência de seleção. Com isso surgem dois novos parâmetros, o número de passos e o período para seleção de sobreviventes modificada. No trabalho apenas o segundo foi variado e foi realizada apenas uma iteração do GLA para o cálculo dos vizinhos.

3) *ES com subpopulações (ESmultimodal)*: Foi proposto também uma variante do ES que utiliza o conceito de subpopulações. Essas apresentavam o número de integrantes constantes, visando manter a diversidade genética. Porém, há migração periódica de indivíduos de maior aptidão entre as ilhas aleatórias buscando melhoria e fuga de mínimos locais pela população em geral.

4) *EP*: Implementação do EP conforme descrito como o meta-EP em [2], com características apresentadas na Tabela II. Além da ausência da recombinação destaca-se ainda a diferença do método de seleção de sobreviventes. No EP, esta etapa se dá por um processo método estocástico, no caso foi utilizado o Stochastic Universal Sampling (SUS), considerando ainda os pais da geração presente como possíveis candidatos.

Tabela II: Características do EP

Recombinação	-
Mutação	Perturbação gaussiana não correlacionada de passo múltiplo
Seleção de pais	Aleatória uniforme
Seleção de Sobreviventes	SUS ( $\mu + \lambda$ )

5) *EP - Fast*: Este algoritmo sugerido em aula busca melhorar o desempenho do EP modificando a atualização dos genes X dos indivíduos durante a mutação. Nesse caso, como no *Fast Simulated Annealing* (FSA) a distribuição normal é substituída pela distribuição de Cauchy (representada pelo  $\mathcal{C}$ ) da maneira apresentada na equação 2.

$$\begin{aligned}\sigma' &= \sigma \cdot e^{\tau_1 \cdot \mathcal{N}(0,1) + \tau_2 \cdot \mathcal{N}_i(0,1)} \\ x'_i &= x_i + \sigma' \cdot \mathcal{C}_i(0,1)\end{aligned}\quad (2)$$

### C. Variação e seleção de Parâmetros

Os algoritmos apresentados dependem de diferentes parâmetros para consecução de suas iterações. As configurações (algoritmo + parâmetros), foram obtidas a partir de um *grid* de valores fixos cujas combinações foram testadas. Para todos os algoritmos foram analisados os seguintes parâmetros:

- $\mu \in \{10, 30, 60, 90\}$
- $\lambda/\mu \in \{3, 5, 7, 9\}$

Para o ES memético foi ainda variado período para seleção modificada,  $epoch \in \{5, 10, 50\}$ . Já para o ES com ilhas, o período para migração foi variado para  $epoch \in \{10, 50\}$  e o número de ilhas  $n \in \{3, 5, 9\}$ , a quantidade de indivíduos migrantes foi mantido constante e unitário.

Devido a restrição de tempo para realização do trabalho foram realizadas rodadas preliminares de 10 lançamentos para seleção da configuração correspondente de cada algoritmo, para cada nível de complexidade do problema. Para o NC=20, no entanto foram executados 30 lançamentos com cada configuração buscando obter soluções porém alguns dos algoritmos não apresentaram nenhum sucesso. Uma vez selecionados os parâmetros foram realizados 100 lançamentos de cada configuração e nível de complexidade para obtenção dos resultados.

### D. Tempo de Execução

O perfil dos tempos de execução de cada lançamento é apresentado na Tabela III. O tempo de execução dos lançamentos apresentados, de sondagem de parâmetros e de níveis de complexidade sem sucesso totalizou cerca de 537 horas. No entanto, a maior parte das execuções foram realizadas em paralelo, em termos práticos os experimentos foram realizados ao longo de cerca de 9 dias ininterruptos. O processamento foi realizado em um computador com CPU i7-11800H, 16GB de RAM DDR4 e GPU RTX 3070, sempre priorizando o processamento na GPU pelo uso da biblioteca *pytorch* com uso de tensores alocados nesta unidade para representar os dados dos algoritmos. Para os níveis de complexidade NC=5 e NC=10 houve o processamento simultâneo de todos os algoritmos. Já para o nível NC=20, devido à conclusão de etapas ou desistência de se obter resultados com outros algoritmos, parte dos lançamentos foi realizado com menos execuções ocorrendo em paralelo, o que pode explicar seu menor tempo de execução.

## IV. RESULTADOS

### A. Desempenho

Todos os algoritmos utilizados apresentaram alguma taxa de sucesso até o nível de complexidade em que NC=10. Para o nível em

Tabela III: Características de tempo decorrido por lançamento [s].

		mean	std	min	max
NC	algo				
5	EP	539	75	322	751
	ES	245	95	190	569
	ESmeme	680	74	473	857
	ESmultimodal	880	312	455	1309
	FEP	643	61	493	813
10	EP	530	50	398	670
	ES	468	44	392	606
	ESmeme	548	203	155	1144
	ESmultimodal	738	236	254	1875
	FEP	474	129	281	671
20	ES	200	98	145	555
	ESmeme	577	261	297	1631

que o NC=20 apenas o ES e o ES memético apresentaram capacidade de resolução com os parâmetros e metodologia de busca apresentado na seção III-C. Os resultados para cada nível são apresentados de maneira resumida nas figuras 1, 2, 3. Esses resumos gráficos apresentam os resultados para as melhores configurações de cada algoritmo e cada nível de complexidade.

O primeiro quadro de cada figura apresenta a taxa de sucesso, SR, e organiza os algoritmos em cada nível de acordo com sua taxa de sucesso.

No segundo, é apresentado uma distribuição da melhor aptidão normalizada em cada um dos 100 lançamentos realizados para a configuração. Esta normalização se dá de acordo com:

$$J_{rel} = -\frac{J}{J_{obj}} \quad (3)$$

Sendo o  $J_{obj}$  calculado conforme conjunto detalhado na seção III-A, para cada lançamento. Observa-se que o algoritmo apresenta resultados superiores a -1, ou seja melhores dos que o objetivo inicial. Isso pode indicar que os GA estudados apresentaram resultados que seriam obtidos com temperaturas inferiores pelo DA ou que a sobreposição dos *clusters* pode gerar outros mínimos que não correspondem aos centroides geradores iniciais. Neste quadro não são apresentados os *outliers* visando enquadrar melhor as distribuições.

No terceiro quadro das figuras de resumo dos resultados são apresentadas as distribuições do número de avaliações da função custo até o sucesso pelo critério estipulado.

Os segundos e terceiros quadros mostram os valores dos melhores ajustes e número de avaliações até o sucesso, no entanto, os indicadores centrais das distribuições são suas medianas. Os valores médios e os resultados pelos parâmetros propostos são apresentados numericamente na Tabela IV, que também apresenta as configurações com as quais se obtiveram os resultados.

Conforme os resumos de resultados apresentados graficamente nas Figuras 1, 2 e 3, e numericamente na Tabela IV é possível tecer os seguintes comentários para cada nível de complexidade:

1) *NC=5*: Para este nível de complexidade do problema se mostrou trivial para a resolução pelas variantes ES. O único algoritmo que apresentou insucesso considerável foi o EP simples, porém com uma SR ainda de 95%. Todos os algoritmos apresentaram MBF relativo melhores do que o  $J_{obj}$ . O melhor algoritmo para esse nível de complexidade em termos de AES foi o ESmultimodal com cerca de 7250 avaliações enquanto, o FEP, que também apresentou SR perfeita no experimento, apresentou AES quase 6 vezes maior.

O FEP apresentou um resultado com SR perfeita. O EP original foi o que pior performou nesse critério, ficando ainda com uma AES correspondente ao dobro do algoritmo variante.

Tabela IV: Resultados numéricos para a melhor configuração de cada algoritmo em cada nível de complexidade

NC	Algorithm	Configuration	SR	MBF rel	AES
5	ES	ES_m=10_l=30	100.00%	-0.94	12150
	ESmultimodal	ESmultimodal_m=60_l=180_e=10_n=5	100.00%	-0.95	7249
	FEP	FEP_m=10_l=30	100.00%	-0.94	42213
	ESmeme	ESmeme_m=30_l=90_e=5	99.00%	-0.96	44668
	EP	EP_m=10_l=70	95.00%	-0.96	94902
10	ESmeme	ESmeme_m=10_l=30_e=50	93.64%	-0.95	121188
	ES	ES_m=10_l=30	92.00%	-0.96	120150
	ESmultimodal	ESmultimodal_m=30_l=210_e=10_n=5	76.24%	-1.03	133361
	FEP	FEP_m=10_l=30	53.00%	-1.06	172760
	EP	EP_m=30_l=90	43.56%	-1.28	184938
20	ESmeme	ESmeme_m=60_l=600_e=50	32.00%	-2.78	277725
	ES	ES_m=90_l=630	24.75%	-3.11	234713

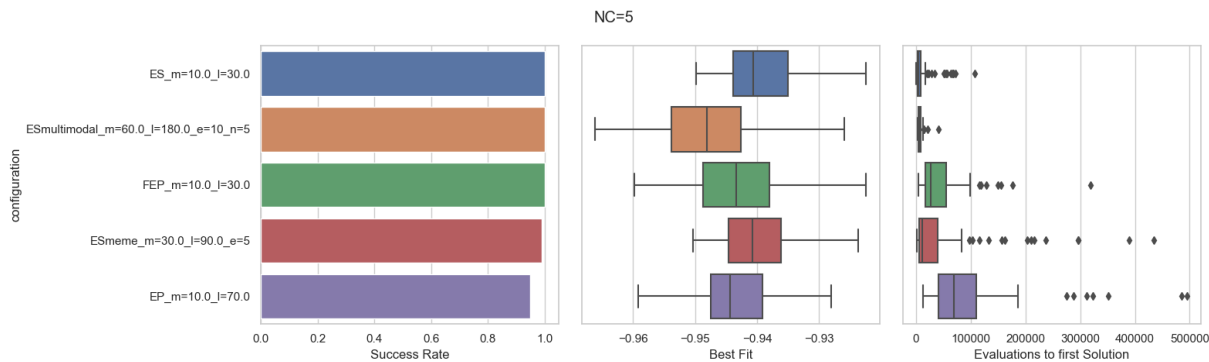


Figura 1: Quadro resumo dos resultados para NC=5

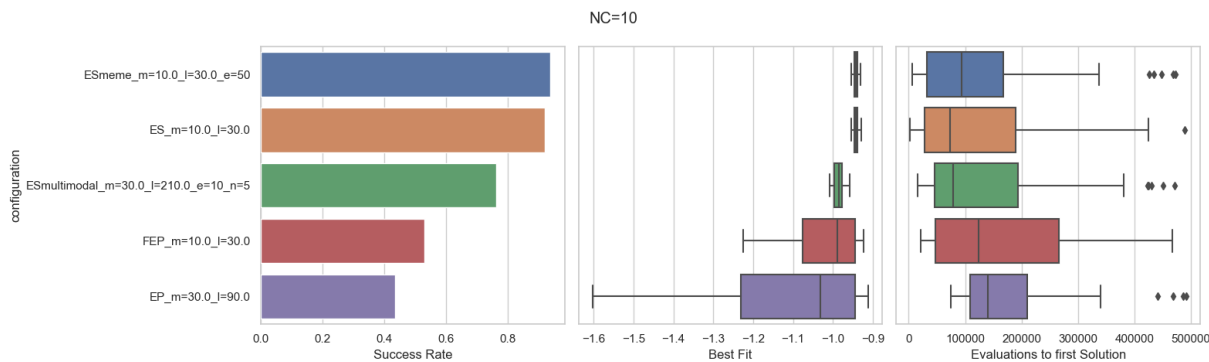


Figura 2: Quadro resumo dos resultados para NC=10

2)  $NC=10$ : Com esse número de *clusters* o problema se mostrou intermediário e houve maior variação de resultados entre os algoritmos. O ESmeme apresentou maior taxa de sucesso que os demais nesse nível, junto com o ES, com uma diferença de apenas cerca de 1%. Em relação às distribuições de melhor aptidão é possível observar que todos as variantes ES apresentaram MBF dentro do critérios de sucesso. Apenas a distribuição dos resultados do ESmultimodal ficou deslocada, já com o primeiro quartil pior do que  $J_{obj}$ . Em termos de AES as variantes ES apresentaram resultados similares entre 120000 e 130000, enquanto os EP apresentaram valores 50% superiores. Há maior lacuna entre os resultados apresentados pelas variantes ES e EP em todos critérios do que no nível anterior. E o FEP apresenta resultado já melhor do que o EP original em cerca de 10% de taxa de resultados e a mediana dos melhores resultados

3)  $NC=20$ : Para este nível de complexidade os algoritmos apresentaram maior dificuldade para resolução, sendo que apenas o ES e o ESmeme apresentaram soluções para o problema. O ESmeme

apresentou melhor taxa de sucesso, superior em 7% em relação à versão original. O MBF já se apresenta em valores mais elevados em módulo, sendo cerca de 3 vezes o valor do objetivo para ambos algoritmos.

#### B. Ganho de Escala

A Figura 4 apresenta a evolução dos resultados por algoritmo, por meio da comparação de SR, MBF e AES da melhor configuração de cada um deles com o aumento da complexidade do problema. Sobre essa é possível formular os comentários elencados a seguir.

Apesar do ESmeme não apresentar sucesso em 100% dos casos para  $NC=5$ , nos demais níveis este se mostrou melhor em relação a SR e MBF. Porém em termos de AES o seu resultado é pior do que o do ES, estando estes dois sempre melhor do que os demais em todos os critérios. A evolução dos resultados do ESmeme parece se apresentar melhor com o aumento da complexidade porém com maior custo computacional.

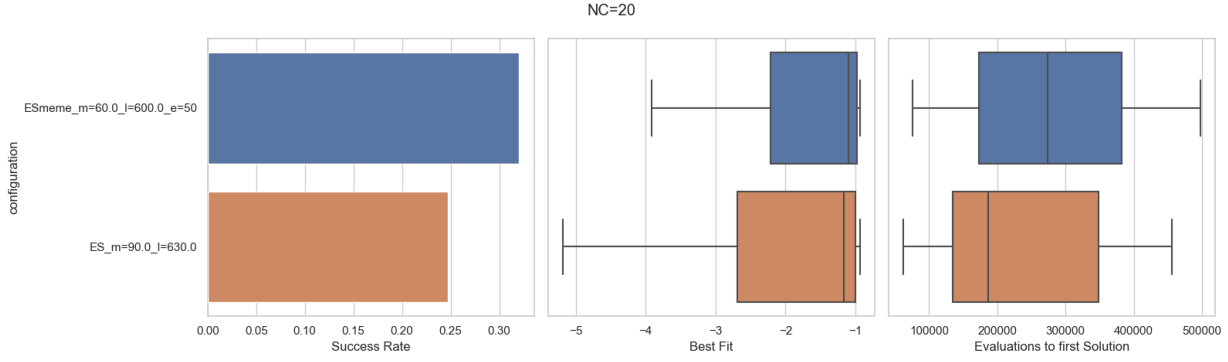


Figura 3: Quadro resumo dos resultados para NC=20

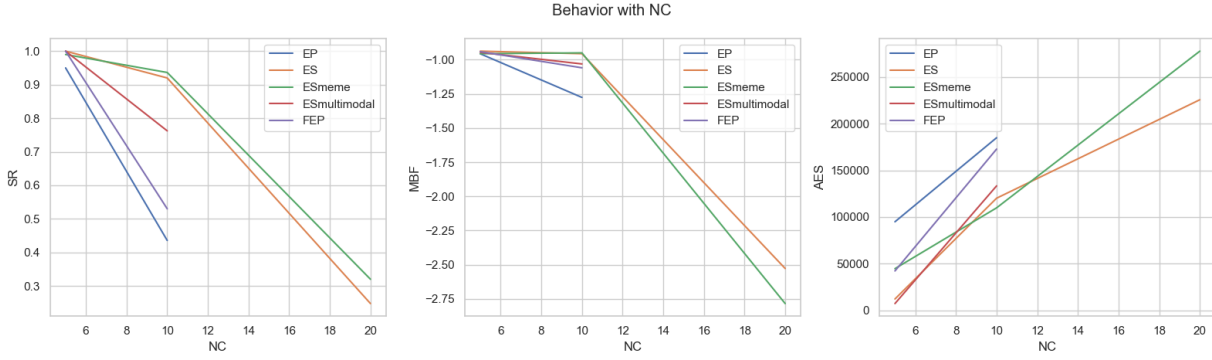


Figura 4: Influência do aumento da complexidade nos parâmetros de desempenho de cada algoritmo

Nesta visualização também se evidencia a diferença de patamares de resultados das variantes ES e ESmeme das demais configurações apresentadas em todos os critérios.

Particularmente, entre as variantes EP a visualização mostra um *gap* constante em termos de SR para os dois níveis nos quais os algoritmos apresentam resultados. Por outro lado, o FEP parece demandar maior esforço computacional com o aumento da complexidade.

## V. CONCLUSÕES

A partir dos resultados apresentados na seção IV é possível concluir que os algoritmos derivados de ES oferecem melhores resultados em todos os critérios analisados. A hipótese que se levanta é que a recombinação, ausente nas variantes EP, exerce importante papel na solução do problema analisado através da passagem de genes que representam as coordenadas de centroides "bons". Ao se lançar essa hipótese, uma possibilidade de memetismo para trabalhos futuros é a ideia de modificar a recombinação no sentido de mudar genes associados aos mesmos centroides de maneira conjunta, diferente do implementado no presente trabalho.

Para o problema analisado, ao focarmos somente nas variantes EP estudadas é natural observar a influência positiva do uso da distribuição de Cauchy para a atualização na mutação com parâmetros auto-ajustáveis assim como o que ocorre no FSA.

Também se observou que os algoritmos analisados apresentam bons desempenhos para baixas complexidades porém há dificuldade no ganho de escala, ou seja, a resolução do problema para maior grau de dificuldade. No entanto, o trabalho apresenta número restrito de parâmetros analisados.

## REFERÊNCIAS

- [1] K. Koutroumbas and S. Theodoridis, *Pattern recognition*. Academic Press, 2008.

- [2] A. E. Eiben, J. E. Smith *et al.*, *Introduction to evolutionary computing*. Springer, 2015.
- [3] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," 1998.