

UMA ANÁLISE DO TRADE-OFF DE DESEMPENHO DAS DIFERENTES CONFIGURAÇÕES DO MODELO EM ILHAS USANDO EVOLUÇÃO DIFERENCIAL

Rodolfo Ayala Lopes

Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais - Brasil
rodolfo.ufop@gmail.com

Alan Robert Resende Freitas

Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais - Brasil
alandefreitas@iceb.ufop.br

RESUMO

O Modelo em Ilhas é uma importante abordagem multipopulacional para Algoritmos Evolutivos. Ela é uma eficiente estratégia de implementação proposta para esta classe de algoritmos. Esta abordagem melhora o desempenho dos Algoritmos Evolutivos promovendo uma busca mais eficiente no espaço de soluções candidatas. No entanto, seu projeto envolve tomar decisões sobre alguns parâmetros deste modelo. Estes parâmetros impactam diretamente no desempenho da abordagem. Embora, seja possível encontrar diferentes configurações para esta abordagem, estudos anteriores ainda são insuficientes para a compreensão do *trade-off* de desempenho desta abordagem multipopulacional. Assim, o principal objetivo deste artigo é conduzir uma análise experimental sobre o *trade-off* de desempenho do Modelo em Ilhas. Visando alcançar este objetivo, diferentes configurações foram estudadas e implementadas no Modelo em Ilhas utilizando o algoritmo de Evolução Diferencial. Os resultados dos experimentos computacionais comprovam a existência do *trade-off* entre a taxa e velocidade de convergência para as diferentes configurações do modelo.

PALAVRAS CHAVE. Modelo em Ilhas, Evolução Diferencial, Algoritmos Evolutivos.

Metaheurísticas

ABSTRACT

The Island Model is a multi-population approach for Evolutionary Algorithms. It is an efficient way to implement these algorithms. This approach improves the performance of the Evolutionary Algorithms providing a better balance between exploration and exploitation in the solution space. However, its design involves taking decisions about some parameters of the model. These parameters are directly related to the Island Model performance. Although, it is possible to find different approaches to set the model in literature, there is still a lack of knowledge about the *trade-off* of the Island Model performance. Thus, the main goal of this paper is to provide an experimental investigation to fill this gap. In order to achieve this goal, the different approaches studied were coupled with the Differential Evolution algorithm. The results of experiments show that there is a *trade-off* between convergence rate and convergence speed among the different approaches.

KEYWORDS. Island Model, Differential Evolution, Evolutionary Algorithms.

Metaheuristics

1. Introdução

O Modelo em Ilhas (IM - Island Model) é uma interessante abordagem multipopulacional para Algoritmos Evolutivos (EA - Evolutionary Algorithms). Originalmente, sua inspiração é oriunda da teoria do Equilíbrio Pontuado [Cohoon et al., 1987]. Os primeiros estudos realizados sobre o IM propuseram sua aplicação para a paralelização dos Algoritmos Genéticos (GA - Genetic Algorithms) [Petty et al., 1987], [Cohoon et al., 1987]. Devido ao seu bom desempenho e aplicabilidade, atualmente, o IM tem sido aplicado a diversos outros EA's. Destaca-se sua aplicação aos métodos de Evolução Diferencial (DE - Differential Evolution) [Tasoulis et al., 2004], Estratégias Evolutivas (ES - Evolution Strategies) [Rudolph, 1991] e Programação Genética (GP - Genetic Programming) [Andre e Koza, 1998].

A ideia fundamental do IM é dividir a população de indivíduos em subpopulações. Neste modelo, as subpopulações permanecem relativamente isoladas e são tradicionalmente conhecidas como ilhas. As ilhas deste modelo possuem seus próprios EA's e elas são autônomas para evoluir suas populações de soluções candidatas. Outra característica importante do IM é o processo de migração. Neste processo, indivíduos são trocados entre as ilhas em um evento ocorrido periodicamente ao longo da execução do IM.

O IM é capaz de melhorar o desempenho dos EA's realizando uma busca mais eficiente no espaço de soluções [Ruciński et al., 2010]. Entretanto, no projeto de um IM, o usuário deve determinar alguns parâmetros obrigatórios para o modelo. Estes parâmetros apresentam impacto direto no desempenho do modelo projetado. Um dos parâmetros críticos do IM é a definição do fluxo migratório [Ruciński et al., 2010]. Este parâmetro determina as conexões entre as ilhas as quais os indivíduos serão trocados durante o processo de migração.

Na atualidade, diversas abordagens para configuração do IM podem ser encontradas na literatura [Tang et al., 2004], [Berntsson e Tang, 2005], [Ruciński et al., 2010], [Lopes et al., 2012], [Lopes et al., 2013]. Entretanto, a escolha da melhor configuração para o IM é uma etapa não trivial aos usuários. Além do mais, estes estudos ainda são insuficientes para compreender o *trade-off* do desempenho do IM. Em sua maioria, os trabalhos relacionados não avaliam adequadamente diferentes configurações para esta abordagem multipopulacional.

Diante deste cenário, a principal meta deste estudo é realizar uma análise experimental crítica das diferentes estratégias de configuração para o IM. Esta análise possibilitaria compreender melhor o conflito entre eficiência e eficácia das diferentes abordagens. Seus resultados experimentais contribuirão para futuros estudos envolvendo o projeto de IM e sua aplicabilidade a EA's.

Visando alcançar os objetivos traçados para este estudo, uma variedade de estratégias de configurações de fluxo migratório são apresentadas. Assim, encontradas na literatura, as abordagens de configuração em Anel [Cantú-Paz, 1998], Estrela [Bujok, 2011], Aleatória [Tang et al., 2004], Q-learning [Lopes et al., 2012], K-medoids [Berntsson e Tang, 2005], Roleta [Lopes et al., 2013] e Torneio [Lopes et al., 2013] são discutidas e avaliadas nos experimentos do trabalho.

Para realização da análise experimental, versões do IM usando o algoritmo de otimização Evolução Diferencial (DE) [Storn e Price, 1997] foram implementadas. A escolha deste método de otimização foi credenciada por seu sucesso de aplicação a problema reais [Onwubolu e Davendra, 2006], [Raj et al., 2012], facilidade de implementação e bom desempenho [Weber et al., 2010]. Os experimentos computacionais foram conduzidos por meio de um conjunto de funções de otimização global em larga escala. Por fim, os resultados dos experimentos retratam a competitividade das abordagens estudadas e o compromisso entre alcançar elevadas taxas de convergência e rapidez.

O restante deste artigo está organizado da seguinte forma: a Seção 2 realiza breve revisão sobre Evolução Diferencial e o Modelo em Ilhas; a Seção 3 descreve diferentes estratégias de configuração do fluxo migratório; a Seção 4 apresenta uma descrição dos experimentos computacionais e os seus resultados; e por fim, a Seção 5 apresenta as discussões e considerações finais.

2. Fundamentação Teórica

2.1. Evolução Diferencial

O método de otimização de Evolução Diferencial (DE), proposto por Storn e Price [1997], é uma técnica de otimização global sobre espaços contínuos considerada confiável, precisa, robusta e rápida [Salman et al., 2006]. As razões do sucesso deste método de otimização podem ser explicadas por sua simplicidade, facilidade de implementação e bom desempenho. Além do mais, o DE tem sido aplicado com sucesso em diversos problemas de otimização mono-objetivo, multiobjetivo e combinatórios.

O método DE faz uso de uma população de soluções candidatas ao problema de otimização, chamadas de indivíduos. Neste método, novas soluções são geradas combinando indivíduos da própria população de soluções. Desta forma, durante o processo evolutivo, novas soluções substituem as soluções atuais, se e somente se, elas possuem melhores valores de função objetivo. O DE apresenta três parâmetros de controle: o tamanho da população (NP), o fator (F) de escala das perturbações geradas pela mutação e a constante de recombinação (CR). A seguir, apresentamos o processo básico de iteração do algoritmo DE:

1. Inicializar a probabilidade de cruzamento CR e o fator de escala F ;
2. Gerar aleatoriamente NP soluções x no espaço de busca;
3. Até que um critério de parada seja atingido, os seguintes passos são repetidos:
 - Gerar nova população de soluções. Assim, para cada solução x da população atual:
 - (a) Selecionar três outras soluções a , b e c da população que sejam diferentes de x ;
 - (b) Gerar uma nova solução y . Assim, para cada posição i da solução x :
 - i. Escolher aleatoriamente um índice j entre 1 e o número de variáveis;
 - ii. Escolher um valor real r entre 0 e 1 com distribuição uniforme;
 - iii. Se $(r < CR)$ ou $(j == i)$, faça, $y_i = a_i + F(b_i - c_i)$, senão, $y_i = x_i$;
 - (c) Se y é melhor que x , x é substituído por y ;
4. Retornar a melhor solução encontrada.

Uma nova geração da população de soluções é resultante de cada repetição do *Passo 3* da iteração do algoritmo apresentado. Neste método, o critério de parada do algoritmo é definido pelo usuário. Tradicionalmente, este critério é definido como o número máximo de novas gerações da população de soluções ou a qualidade da solução encontrada pelo método.

2.2. Modelo em Ilhas

Inseridos no contexto dos EA's, o IM é uma importante abordagem multipopulacional para esta classe de metaheurísticas. Esta abordagem multipopulacional é uma eficiente e popular estratégia de implementação para EA's. Sua eficiência pode ser comprovada em ambas arquiteturas, paralelas ou sequenciais [Rudolph, 1991], [Cantú-Paz, 1998], [Tasoulis et al., 2004]. Estudos demonstram que o IM é capaz de melhorar o desempenho dos EA's devido sua boa capacidade de exploração do espaço de buscas [Ruciński et al., 2010].

O IM é uma estratégia facilmente paralelizável e em alguns casos seu ganho de desempenho alcançado pode ser superlinear, conforme demonstrado em estudos anteriores [Cantú-Paz, 2001], [Alba, 2002]. Além do seu bom desempenho em implementações paralelas, o IM é capaz de melhorar a pesquisa evolucionária por si só. Desta forma, mesmo em implementações sequenciais é possível alcançar melhorias na qualidade das soluções encontradas e no tempo de convergência [Whitley et al., 1998], [Tasoulis et al., 2004].

Basicamente, a proposta do IM é dividir a população de soluções em subpopulações relativamente isoladas. As subpopulações definidas após a divisão são também conhecidas como ilhas. Cada subpopulação possui seu próprio EA que evolui independentemente. Esta relativa independência entre as ilhas tende a melhorar o processo de otimização, estimulando a pesquisa em regiões distintas do espaço de soluções. Entretanto, outra característica importante desta abordagem multipopulacional é a colaboração entre as ilhas, em um processo conhecido como migração. O processo de migração pode ocorrer periodicamente e consiste na troca de soluções candidatas. Este processo permite melhorar o desempenho das ilhas introduzindo nas subpopulações novas informações sobre o problema de otimização.

Durante o projeto de um IM, decisões sobre a configuração dos parâmetros do modelo são necessárias. Estes parâmetros estão diretamente relacionados ao desempenho desta abordagem. Sendo assim, os parâmetros do IM são apresentados a seguir, de forma resumida:

- **Número de Ilhas:** *define o número de subpopulações do modelo;*
- **Topologia de migração:** *descreve a estrutura de comunicação do modelo;*
- **Frequência de migração:** *define com qual frequência o processo de migração irá ocorrer;*
- **Taxa de migração:** *determina quantos indivíduos irão migrar de uma subpopulação a outra;*
- **Tipo de Sincronização:** *determina o tipo de sincronização (assíncrono ou síncrono) das ilhas do modelo para realização do processo de migração;*
- **Política de migração:** *descreve quais indivíduos serão removidos e substituídos durante o processo de migração;*
- **Fluxo Migratório:** *determina o caminho dos indivíduos emigrantes dentro da estrutura de comunicação.*

2.3. Fluxo Migratório

O fluxo migratório é um importante parâmetro do IM [Ruciński et al., 2010]. Ele descreve o caminho dos indivíduos emigrantes durante a execução do processo de migração. Sua representação é realizada por um conjunto de arestas direcionadas e forma a estrutura de vizinhança do modelo. Estudos relacionados relatam que a escolha adequada do fluxo migratório pode garantir modelos mais eficientes do que aqueles com outras configurações [Lopes et al., 2012], [Lopes et al., 2014].

3. Configuração do Fluxo Migratório

De maneira geral, as estratégias de configuração do fluxo migratório podem ser classificadas em abordagens estáticas e dinâmicas. Esta classificação é realizada analisando o comportamento do fluxo migratório ao longo da execução do IM. As subseções apresentadas a seguir detalham as principais abordagens encontradas na literatura.

3.1. Configuração Estática

As configurações estáticas são definidas *a priori* e não sofrem alterações durante o processo evolutivo [Tang et al., 2004]. Em grande parte das configurações estáticas, o fluxo migratório é baseado em redes e arquiteturas de computadores. As abordagens em Anel e Estrela são as duas mais tradicionais configurações estáticas encontradas na literatura. A configuração em Anel é a abordagem mais estudada no projeto de IM. Seu formato anatômico consiste em um circuito fechado de ilhas. Por outro lado, a configuração em Estrela é formada por uma ilha central que se conecta com as demais, não havendo conexão entre as demais ilhas.

3.2. Configuração Dinâmica

Conceito recentemente citado em [Lopes et al., 2013], [Lopes et al., 2014], a proposta da configuração dinâmica do fluxo migratório é melhorar o desempenho e flexibilidade do IM. Esta abordagem é composta por um mecanismo inteligente de configuração. Seu mecanismo é responsável por aprender e ajustar o fluxo migratório, de acordo com o estágio do processo evolutivo [Lopes et al., 2012]. O fluxo migratório definido pelo mecanismo surge a partir de métodos heurísticos.

Assim, o restante desta subseção 3.2 se dedica a apresentar as principais abordagens de configuração dinâmica do fluxo migratório encontradas na literatura.

3.2.1. Configuração Aleatória

A configuração Aleatória [Tang et al., 2004] é a mais simples e tradicional abordagem dinâmica discutida na literatura. Basicamente, o processo de definição do seu fluxo migratório é realizado por ilha do modelo. Para cada ilha, seu mecanismo de seleção aleatória seleciona a origem dos indivíduos que serão recebidos durante o processo de migração. Ao final do processo de seleção, o fluxo migratório completo encontra-se definido. Ressalta-se que não é permitido enviar indivíduos para suas próprias ilhas, ou seja, as seleções cíclicas não são aceitas nesta estratégia.

A estratégia de configuração aleatória garante constante alteração do fluxo migratório e se destaca por sua dinâmica e simplicidade de implementação [Lopes et al., 2014]. Por outro lado, esta estratégia de configuração é considerada ingênua pois ela ignora qualquer informação adquirida durante o processo evolutivo [Lopes et al., 2014].

3.2.2. Configuração usando Seleção por Roleta

A abordagem de configuração usando Seleção por Roleta foi proposta por Lopes et al. [2013]. Sua concepção é similar ao mecanismo de configuração aleatória discutida na subseção anterior. No entanto, o intuito desta estratégia é explorar a dinâmica da evolução das subpopulações do modelo. Para este fim, assume-se que a aptidão dos melhores indivíduos das ilhas tem alta probabilidade de melhorar o desempenho do IM. Assim, para cada ilha, a origem da solução candidata que será recebida é definida por um método de seleção baseado na aptidão dos indivíduos.

O algoritmo da Roleta [Jaddan et al., 2005] foi o método de seleção escolhido para compor este mecanismo de configuração do fluxo migratório. Este método de seleção probabilístico é normalmente utilizado em GA's. Neste método de seleção, para cada indivíduo é atribuído uma probabilidade de seleção proporcional a sua aptidão, ou seja, os indivíduos mais aptos têm maior probabilidade de seleção.

Neste mecanismo de configuração, a probabilidade de seleção atribuída ao i -ésimo indivíduo é dada por:

$$PS_i = \frac{f_n(x_i)}{\sum_{j=1}^{NI} f_n(x_j)} \quad (1)$$

onde $f_n(x_i)$ é a aptidão normalizada do indivíduo x_i e NI é o total do número de indivíduos participantes do processo de seleção.

Para cada subpopulação do modelo, o método de seleção por Roleta é aplicado para seleção do indivíduo imigrante. A seleção é realizada entre os melhores indivíduos de todas as demais subpopulações. Assim, o processo de seleção se repete até que todas as subpopulações tenham seu próprio indivíduo imigrante definido.

3.2.3. Configuração usando Seleção por Torneio

Proposta por [Lopes et al., 2013], esta abordagem segue a mesma ideia de configuração do fluxo migratório usando Seleção por Roleta. Um mecanismo idêntico de configuração utiliza o conhecido algoritmo de Seleção por Torneio [Goldberg e Deb, 1991]. Este método de seleção por Torneio é uma robusta abordagem de seleção e comumente aplicado em GA's [Miller e Goldberg,

1995]. Neste processo de seleção, os indivíduos são selecionados por torneios no qual S indivíduos selecionados aleatoriamente são comparados com relação a sua aptidão.

Basicamente, a seleção por Torneio realiza comparações sucessivas entre soluções candidatas. Os vencedores das disputas competem com outras soluções até que o tamanho do torneio S seja alcançado. Em cada comparação, um valor aleatório com distribuição uniforme entre 0 e 1 é sorteado e comparado com o parâmetro kt , definido como $\{kt \in \mathbb{R} \mid 0 \leq kt \leq 1\}$. Caso o valor aleatório sorteado seja menor que kt , o indivíduo com maior aptidão vence a comparação, caso contrário, o pior indivíduo é selecionado.

O mecanismo desta estratégia de configuração do fluxo migratório trabalha de maneira semelhante às abordagens dinâmicas apresentadas anteriormente. Para cada ilha do modelo, o método de seleção por Torneio é executado definindo a solução imigrante recebida durante o processo de migração. A solução imigrante é determinada a partir das melhores soluções de todas as demais ilhas do modelo. Este processo se repete até que todas as ilhas definam seus indivíduos imigrantes.

3.2.4. Configuração usando K-medoids

Proposta por Bertsson e Tang [2005], a configuração do fluxo migratório usando o algoritmo K-medoids é baseada na clusterização das soluções candidatas de cada ilha. A ideia principal é encontrar grupos de ilhas que trabalham em regiões similares do espaço de buscas. O objetivo central desta estratégia é reduzir a conectividade e complexidade do modelo mantendo o bom desempenho.

O algoritmo K-medoids é um método de aprendizado não-supervisionado [Tan et al., 2005]. Este método define automaticamente k grupos baseado na similaridade do conjunto de dados [Tan et al., 2005]. Assim, busca-se encontrar grupos que minimizem a distância entre os dados a cerca dos k pontos mais representativos, onde k também é conhecido como *medoids*.

A configuração do fluxo migratório utilizando o algoritmo K-medoids pode ser resumida em duas etapas:

- *Definição dos grupos*: dado as melhores soluções de cada ilha do modelo, o algoritmo K-medoids é utilizado para encontrar os grupos de pontos similares. Entretanto, o número de grupos e a métrica de distância entre pontos deve ser definida pelo usuário;
- *Definição do fluxo migratório*: dados os grupos definidos na etapa anterior, o fluxo migratório é definido conectando apenas ilhas do mesmo grupo. Desta forma, para cada grupo, o fluxo migratório pode ser definido como aleatório ou em anel.

É importante ressaltar que a escolha do número de grupos é um problema não trivial na configuração desta estratégia. Além do mais, o custo de processamento deste mecanismo aumenta consideravelmente à medida que o número de ilhas do modelo aumenta.

3.2.5. Configuração usando Q-learning

Esta estratégia de configuração de fluxo migratório utilizando o algoritmo Q-learning foi proposta por Lopes et al. [2012]. Sua ideia principal é transformar o IM em um Sistema Multiagentes (MAS - Multi-Agent Systems). No MAS proposto para esta estratégia, cada ilha do modelo é associada a um agente. Estes agentes são responsáveis por aprender as conexões que mais colaboram para a evolução de suas subpopulações.

Os agentes projetados para este modelo agem independentemente e adquirem conhecimento por meio da experiência obtida de processos migratórios anteriores. Nesta estratégia, o processo de aprendizado é realizado pelo algoritmo Q-learning, o qual, é um tradicional método de aprendizado por reforço [Watkins e Dayan, 1992]. Seu modelo de aprendizado realiza-se por meio de interações de tentativa e erro com o ambiente [Kaelbling et al., 1996].

Nesta estratégia proposta, cada agente do modelo possui um conjunto de estados e ações. Os estados representam as subpopulações as quais o agente poderá receber indivíduos durante o processo de migração. Em contrapartida, as ações dos agentes descrevem a modificação ou manutenção da subpopulação (estado) escolhido para receber indivíduos.

Embora possua um modelo de aprendizado complexo e requeira a configuração de parâmetros essenciais, estudos demonstram que o algoritmo Q-learning converge para uma política ótima de ações [Watkins e Dayan, 1992]. Outra característica importante deste algoritmo de aprendizado é que a reinicialização dos parâmetros pode ser necessária durante sua execução.

4. Experimentos Computacionais

4.1. Configurações dos Experimentos

Para análise e comparação do desempenho das diferentes abordagens apresentadas, versões do IM considerando o tradicional algoritmo DE [Storn e Price, 1997] foram implementadas. Conforme sugerido em [Price et al., 2005], os valores dos parâmetros do DE foram definidos como, $CR = 0,9$ e F escolhido aleatoriamente no intervalo $[0,5; 1,0]$ para cada novo indivíduo gerado.

Nos experimentos projetados, variações do IM implementados em arquitetura computacional sequencial consideraram projetos com 4, 8, 16 e 32 subpopulações. Cada ilha do modelo foi composta por 25 soluções candidatas e o efeito do intervalo de migração foi avaliado para 5, 15, 25 e 50 gerações. O processo de migração foi definido como síncrono e a política de migração adotada considerou que os melhores indivíduos substituíssem indivíduos aleatórios durante o processo de migração. Ao longo deste processo, apenas um indivíduo por ilha era selecionado para migração e substituição.

Em relação as abordagens de configuração de fluxo migratório apresentadas neste artigo, seus parâmetros foram baseados em trabalhos relacionados. Para a abordagem Torneio, o tamanho do torneio (S) foi descrito como 3 e o parâmetro kt igual à 1,0, conforme [Lopes et al., 2013]. De acordo como [Lopes et al., 2012], os parâmetros da abordagem Q-learning foram fixados como $\gamma = 1,0$, $\lambda = 0,5$, $\epsilon = 1,0$. A abordagem K-medoids considera o valor de $k = 2$ e a métrica Euclidiana para mensurar a distância entre pontos, conforme sugerido em [Lopes et al., 2012].

Para esta comparação de diferentes configurações do IM, seis funções de teste do Conjunto de Funções para Otimização Global em Larga Escala foram consideradas [Tang et al., 2007]. As funções f_1 e f_2 são unimodais e as funções f_3 , f_4 , f_5 e f_6 são multimodais. Todas as funções foram definidas com 30 variáveis. Para cada configuração testada, 25 independentes execuções foram realizadas em cada função. Durante os experimentos, o processo de evolução era finalizado sempre que algum dos seguintes critérios fosse atendidos:

- O número máximo de gerações (12.000) fosse atendido;
- $\Delta f = f_{target} - f_{best} \leq 10^{-4}$;

onde f_{target} , f_{best} representam, respectivamente, o valor ótimo conhecido para a função de teste e o melhor resultado encontrado pelo algoritmo de otimização.

Durante os experimentos, três variáveis de resposta foram coletadas, (i) número de gerações para convergência, (ii) percentagem de convergência e (iii) tempo computacional de processamento. Em relação ao número de gerações, somente execuções que encontraram com sucesso o ótimo global da função são consideradas. Desta maneira, a percentagem de convergência pode ser considerada como uma medida complementar do desempenho global das configurações avaliadas.

Dada a refutação da hipótese de normalidade dos resultados, o teste não-paramétrico de múltiplo contraste ¹ descrito em [Konietschke et al., 2012] foi aplicado para a análise estatística

¹Função `mctp` (`pacote nparcomp`) do software estatístico R.

sobre o número de gerações gastas para convergência. O modelo estatístico considerado foi um planejamento desbalanceado [Montgomery, 2006], onde a configuração do IM foi tratada como um fator experimental. Para os testes, o nível de significância α foi previamente definido com 0,05.

4.2. Resultados

Comparações entre as diferentes abordagens estudadas são apresentadas ao longo da seção. Inicialmente, a Figura 1 retrata a distribuição, em relação ao desempenho, das diferentes configurações. De forma complementar, na própria Figura 1 a relação de dominância² entre as configurações experimentadas é retrata por meio da fronteira Pareto, também conhecida como conjunto Pareto ótimo³. Nestas análises, o desempenho das configurações é avaliado em termos de percentual de convergência e número médio de gerações gastas. Todas as configurações do IM foram consideradas, de acordo com os parâmetros citados na Seção 4.1. Entretanto, apenas as execuções que encontraram o ótimo global são consideradas no número médio de gerações.

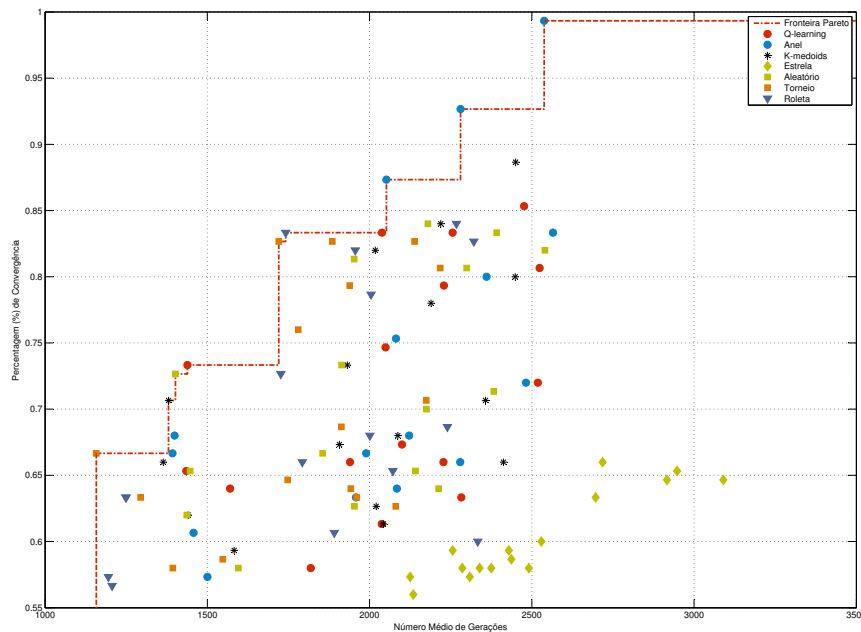


Figura 1: Gráfico do desempenho das diferentes configurações do IM em relação ao percentual de convergência e o número médio de gerações.

Observando a Figura 1, identifica-se diferença no desempenho entre as diversas configurações apresentadas. Assim, conforme o gráfico exposto, as abordagens de fluxo migratório que incorporam informações sobre o processo evolutivo, K-medoids, Q-learning, Torneio e Roleta, tendem a convergir em menor número de gerações. Por outro lado, mesmo não se valendo de nenhuma informação, a configuração em Anel apresenta resultados competitivos destacando-se em seu elevado percentual de convergência. Ao contrário da abordagem estática em Anel, a configuração em Estrela se destaca por apresentar o pior desempenho das abordagens estudadas. O fraco desempenho da abordagem em Estrela indica que seu formato não é uma boa estratégia de configuração para o projeto de IM proposto.

²Dizemos que uma solução x_1 domina x_2 se $f(x_1) \leq f(x_2)$ e $f(x_1) \neq f(x_2)$. Esta relação de dominância é escrita como $f(x_1) \prec f(x_2)$.

³O conjunto Pareto ótimo contém as soluções não dominadas em relação ao conjunto de soluções factíveis.

Percebe-se na Figura 1, o conflito entre as variáveis de validação de desempenho das configurações do IM avaliadas. Assim, a Figura 1 também representa a relação de dominância das diversas configurações. Esta relação de dominância é representada no gráfico por meio da fronteira Pareto descrita por uma linha vermelha tracejada. As configurações avaliadas, as quais encontram-se contidas nesta linha traçada no gráfico, representam o conjunto Pareto ótimo do problema de avaliação do *trade-off* de desempenho em IM. Desta forma, analisando a figura, identifica-se que 10 configurações distintas são consideradas incomparáveis entre si, conforme os critérios de dominância citados anteriormente.

Para melhor identificação das configurações pertencentes ao conjunto Pareto ótimo, a Tabela 1 apresenta os detalhes e resultados dos experimentos dessas abordagens. Conforme a tabela, é possível notar o melhor desempenho dos IM com maior número de ilhas. Observa-se também que um conjunto de parâmetros de configuração em Anel obteve os maiores percentuais de convergência e que uma configuração usando Torneio obteve a menor média de número de gerações gastas.

Configuração	Nº de Ilhas	Intervalo de Migração	% de Convergência	Nº Médio de Gerações
Q-learning (QLN-1)	32	5	73,33%	1.438,73
Q-learning (QLN-2)	32	15	83,33%	2.038,20
Anel (ANL-1)	32	15	87,33%	2.051,79
Anel (ANL-2)	32	25	92,66%	2.280,22
Anel (ANL-3)	32	50	99,33%	2.537,92
K-medoids (KMD-1)	32	5	70,66%	1.380,28
Aleatório (ALT-1)	32	5	72,66%	1.401,83
Torneio (TNO-1)	32	5	66,66%	1.157,75
Torneio (TNO-2)	32	15	82,66%	1.720,04
Roleta (RLT-1)	32	15	83,33%	1.741,56

Tabela 1: Configurações do Modelo em Ilhas pertencentes a Fronteira Pareto ótimo.

Configuração 1	Configuração 2	Valor p	Análise do Resultado
Q-learning (QLN-1)	Torneio (TNO-1)	$1,8262 * 10^{-02}$	TNO-1 melhor que QLN-1
Q-learning (QLN-2)	Torneio (TNO-1)	$8,8817 * 10^{-16}$	TNO-1 melhor que QLN-2
Anel (ANL-1)	Torneio (TNO-1)	$3,7858 * 10^{-14}$	TNO-1 melhor que ANL-1
Anel (ANL-2)	Torneio (TNO-1)	0,0000	TNO-1 melhor que ANL-2
Anel (ANL-3)	Torneio (TNO-1)	0,0000	TNO-1 melhor que ANL-3
K-medoids (KMD-1)	Torneio (TNO-1)	$6,8100 * 10^{-01}$	TNO-1 e KMD-1 são iguais
Aleatório (ALT-1)	Torneio (TNO-1)	$1,2097 * 10^{-01}$	TNO-1 melhor que ALT-1
Torneio (TNO-2)	Torneio (TNO-1)	$3,3524 * 10^{-08}$	TNO-1 melhor que TNO-2
Roleta (RLT-1)	Torneio (TNO-1)	$6,1732 * 10^{-10}$	TNO-1 melhor que RLT-1

Tabela 2: Valor p da análise estatística do número médio de gerações.

Considerando apenas as configurações do IM pertencentes a fronteira Pareto, a Tabela 2 apresenta os resultados e o valores- p dos testes estatísticos executados. Nesta análise estatística o número de gerações para a convergência foi considerado. Conforme a tabela, os resultados demonstram que a configuração nomeada *TNO-1* convergiu em um número de gerações significativamente menor que as demais, exceto, quando comparado a configuração *KMD-1*.

Em relação ao tempo computacional de execução dos experimentos, a Figura 2 apresenta a distribuição empírica do dados relativos as diferentes abordagens. Nesta análise da variação do tempo (segundos), foram considerada todas as execuções, sendo que, as mesmas foram agrupadas

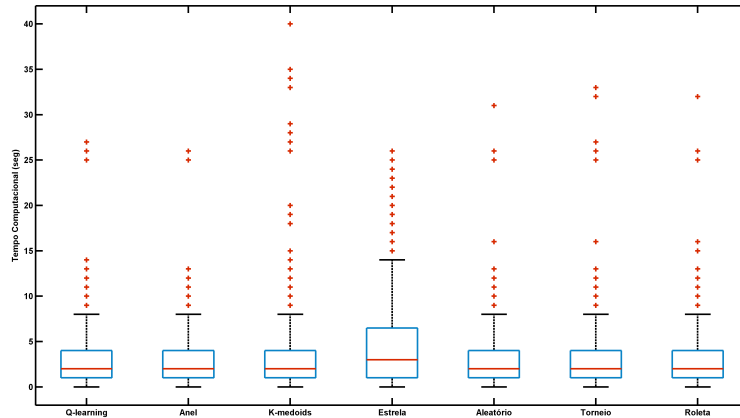


Figura 2: Gráfico da distribuição empírica do tempo computacional gasto pelas diferentes abordagens.

por abordagem de configuração de fluxo migratório. Nota-se por meio do gráfico apresentado que, em sua maioria, as abordagens apresentam tempos de execução similares. Destaca-se apenas a configuração em Estrela, por apresentar os piores resultados dentre as configurações avaliadas.

5. Considerações Finais

Neste artigo, uma análise experimental sobre o conflito entre eficiência e eficácia de diferentes configurações para o IM foi apresentada. Para esta finalidade, uma versão do IM usando o algoritmo de otimização DE foi considerada e as diferentes configurações para o modelo foram avaliadas nos experimentos. O principal objetivo deste estudo foi elucidar o compromisso entre atingir elevadas taxas de convergência e agilidade na busca pela solução ótima em diferentes configurações do IM. Assim, os resultados deste estudo trouxeram contribuições no sentido de compreender melhor o comportamento do IM e suas diferentes possibilidades de configuração.

Os resultados dos experimentos sobre um conhecido conjunto de funções de teste demonstraram que há diferença no desempenho das diversas configurações avaliadas. Em resumo, claramente nota-se um *trade-off* de desempenho das melhores configurações do IM e que as demais configurações avaliadas podem ser descartadas. Os resultados também indicam que as configurações que convergem rapidamente para o ótimo da função tendem a não obter elevadas taxas de convergência. Não obstante, as abordagens que convergem mais vezes são mais lentas em relação ao número de gerações gastas.

A análise do resultado dos experimentos sugere também que modelos maiores são mais eficientes e eficazes. A provável razão é o número maior de soluções candidatas em todo o modelo e a manutenção da diversidade das subpopulações. Outro fato interessante, retrata a rápida convergência das abordagens de configuração dinâmica do fluxo migratório. A rapidez na convergência se deve ao aumento da pressão seletiva imposta ao IM. Contudo, a comprovação da eficiência da abordagem em Anel é retratada por sua elevada taxa de convergência. Em relação ao tempo computacional, graficamente pôde-se observar tempos semelhantes entre as abordagens estudadas. Este fato demonstra que o aumento da complexidade devido a obtenção da informação para as abordagens dinâmicas não resulta em aumento significativo de tempo execução computacional.

Por fim, este estudo alcançou seus objetivos propostos produzindo conhecimentos importantes para o projeto de um IM. Além do mais, a análise do *trade-off* de desempenho das diferentes configurações avaliadas são de grande importância para ajudar futuros estudos sobre abordagens multipopulacionais para EA's.

6. Agradecimentos

Os autores agradecem a Universidade Federal de Ouro Preto e as agências brasileiras de suporte a pesquisa, CNPq e CAPES.

Referências

- Alba, E. (2002). Parallel evolutionary algorithms can achieve super-linear performance. In *Information Processing Letters*, volume 82, p. 7–13. Elsevier Science Inc.
- Andre, D. e Koza, J. R. (1998). A parallel implementation of genetic programming that achieves super-linear performance. In *Information Processing Letters*, volume 106, p. 201–218. Elsevier Science Inc., New York, NY, USA.
- Berntsson, J. e Tang, M. (2005). Dynamic optimization of migration topology in internet-based distributed genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, p. 1579–1580, ACM, New York, NY, USA.
- Bujok, P. (2011). Parallel models of adaptive differential evolution based on migration process. In *Aplimat, 10th International Conference on Applied Mathematics*, p. 357–364, Bratislava: Slovak Technical University.
- Cantú-Paz, E. (2001). Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms. In *Journal Of Heuristics*, volume 7, p. 311–334. Kluwer Academic Publishers.
- Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux Et Systems Repartis*, 10:141–171.
- Cohoon, J. P., Hegde, S. U., Martin, W. N., e Richards, D. (1987). Punctuated equilibria: a parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic algorithms and their application*, p. 148–154, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- Goldberg, D. E. e Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms*, p. 69–93. San Francisco, CA: Morgan Kaufmann.
- Jaddan, O. A., Rajamani, L., e Rao, C. R. (2005). Improved selection operator for ga. p. 269–277. Asian Research Publishing Network, Pakistan.
- Kaelbling, L. P., Littman, M. L., e Moore, A. W. (1996). Reinforcement learning: a survey. In *Journal of Artificial Intelligence Research*, volume 4, p. 237–285. Access Foundation and Morgan Kaufmann Publishers.
- Konietschke, F., Hothorn, L. A., e Brunner, E. (2012). Rank-based multiple test procedures and simultaneous confidence intervals. In *Electronic Journal of Statistics*, volume 6, p. 738–759. The Institute of Mathematical Statistics and the Bernoulli Society.
- Lopes, R. A., Silva, R. C. P., Campelo, F., e Guimarães, F. G. (2013). Dynamic selection of migration flows in island model differential evolution. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, p. 173–174. ACM.
- Lopes, R. A., Pedrosa Silva, R. C., Freitas, A. R., Campelo, F., e Guimarães, F. G. (2014). A study on the configuration of migratory flows in island model differential evolution. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, p. 1015–1022, New York, NY, USA. ACM.

- Lopes, R. A., Silva, R. C. P., Campelo, F., e Guimares, F. G. (2012). A multi-agent approach to the adaptation of migration topology in island model evolutionary algorithms. In *Brazilian Symposium on Neural Networks, Curitiba, Brazil*, p. 160–165. IEEE.
- Miller, B. L. e Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212.
- Montgomery, D. C. (2006). *Design and Analysis of Experiments*. John Wiley & Sons.
- Onwubolu, G. e Davendra, D. (2006). Scheduling flow shops using differential evolution algorithm. *European Journal of Operational Research*, 171(2):674 – 692.
- Pettey, C. C., Leuze, M. R., e Grefenstette, J. J. (1987). A parallel genetic algorithm. In *International Computer Games Association*, p. 155–161.
- Price, K., Storn, R., e Lampinen, J. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer.
- Raj, C. T., Thangaraj, R., Pant, M., Bouvry, P., e Abraham, A. (2012). Design optimization of induction motors with differential evolution algorithms with an application in textile spinning. *Applied Artificial Intelligence*, 26(9):809–831.
- Ruciński, M., Izzo, D., e Biscani, F. (2010). On the impact of the migration topology on the island model. In *Parallel Computing*, volume 36 of *Parallel Architectures and Bioinspired Algorithms*, p. 555–571. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands.
- Rudolph, G. (1991). Global optimization by means of distributed evolution strategies. In *Proceedings of the 1st International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science., p. 209–213. Springer, Berlin.
- Salman, A., Engelbrecht, A. P., e Omran, M. G. (2006). Empirical analysis of self-adaptive differential evolution. *European Journal of Operational Research*, 183:785–804.
- Storn, R. e Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.
- Tan, P.-N., Steinbach, M., e Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Tang, J., Lim, M.-H., Ong, Y.-S., e Er, M. J. (2004). Study of migration topology in island model parallel hybrid-ga for large scale quadratic assignment problems. In *Proceedings of the 8th International Conference on Control, Automation, Robotic and Vision, Kunming, China*.
- Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M., e Yang, Z. (2007). Benchmark functions for the congress on evolutionary computation (cec) 2008. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China.
- Tasoulis, D. K., Pavlidis, N., Plagianakos, V. P., e Vrahatis, M. N. (2004). Parallel differential evolution. In *IEEE Congress on Evolutionary Computation (CEC)*.
- Watkins, C. J. C. H. e Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.
- Weber, M., Tirronen, V., e Neri, F. (2010). Scale factor inheritance mechanism in distributed differential evolution. *Soft Computing*, 14:1187–1207.
- Whitley, D., Rana, S., e Heckendorn, R. (1998). The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, p. 33–47.