SELEÇÃO DE ALGORITMO PARA PROBLEMAS DE OTIMIZAÇÃO CONTÍNUOS *BLACK-BOX*

Matheus Silva Gonçalves

Rafael Holdorf Lopez

Felipe Carraro

matheusgoncalves.contato@gmail.com

rafaelholdorf@gmail.com

felipecarraro@gmail.com

Center for Optimization and Reliability on Engineering - CORE. Departamento de engenharia civil. UFSC

Rua João Pio Duarte, 88037-000. Florianópolis, SC, Brasil.

Abstract. *A tarefa de selecionar o algoritmo de otimização mais adequado para um dado problema pode ser muito complexa e, se mal executada, trazer resultados insatisfatórios. O principal objetivo deste trabalho é ajudar o usuário nessa tarefa, fornecendo um meio automatizado de realizar essa seleção. Utilizou-se da estrutura ASP (Algorithm Selection Problem) para tal. Dentro dessa estrutura, o processo de seleção pode ser visto como uma tarefa de aprendizado, no qual o método precisa aprender um mapeamento entre problema e algoritmo que tem a maior probabilidade de desempenhar melhor no mesmo. Utilizou-se dos conceitos de exploratory landscape analysis (ELA) para caracterizar os problemas de otimização. Baseado nas características de um problema, aplica-se técnicas de aprendizado de maquina para selecionar o algoritmo mais adequado. Os resultados obtidos são promissores, indicando que as medidas ELA utilizadas são representativas e a metodologia desenvolvida foi capaz de selecionar o algoritmo mais adequado para o problema em questão. Uma evolução natural é a consideração de mais algoritmos na estrutura, tornando a metodologia capaz de tratar mais problemas e de maneira mais eficiente.*

Keywords: Otimização contínua *black-box*, *Exploratory landscape analysis*, Aprendizado de máquina, Seleção de algoritmo

1 INTRODUÇÃO

Otimização é um campo que vem crescendo muito nos últimos anos, como pode ser visto pelo grande número de estudos disponíveis na literatura. Geralmente estes focam no desenvolvimento de novos algoritmos ou na aplicação destes em problemas específicos. Apesar do grande número de algoritmos de otimização desenvolvidos, não existe um que seja universalmente melhor para um grande conjunto de problemas (SMITH-MILES, 2008). Dessa forma, o uso adequado dos mesmos ainda requer conhecimento significante tanto dos métodos existentes e os casos em que trabalham melhor quanto das próprias características do problema a ser resolvido. Para o caso de problemas não triviais, a qualidade da solução ainda está muito relacionada com experiência de quem esta implementando (CARCHRAE; BECK, 2005).

Se por um lado é interessante ter muitos métodos disponíveis, por outro o surgimento constante de novos algoritmos torna extremamente difícil para usuários estarem familiarizados com as particularidades de cada um (MUÑOZ et al., 2015). Em especial, como esses métodos funcionam e para qual tipo de problema eles são mais adequados. Como um método pode trazer resultados insatisfatórios caso aplicado a um problema inadequado, tem-se que a tarefa de seleção de algoritmo é muito importante e complexa.

Com base nisso, este trabalho tem por objetivo ajudar o usuário nessa tarefa, fornecendo um meio de selecionar o algoritmo mais adequado para um dado problema, de forma automatizada e robusta. Para tal, utiliza-se a estrutura proposta por Rice (1976), o *Algorithm Selection Problem* (ASP). Esta é uma forma clara e sistemática de se abordar o problema de escolha de algoritmos, utilizada por muitos pesquisadores, não restritos apenas a área de otimização (BISCHL et al., 2012), (SMITH-MILES, 2008), (MUÑOZ; KIRLEY; HALGAMUGE, 2012), (ABELL; MALITSKY; TIERNEY, 2012).

Dentro desta estrutura, destaca-se a necessidade de elaborar um conjunto composto por algoritmos de otimização aptos a serem escolhidos. Dentre um incontável número de métodos existentes, deve-se escolher componentes que tenham capacidades complementares (MUÑOZ et al., 2015), sendo esses condizentes com o conjunto de problemas de interesse.

Vale ressaltar que opta-se por considerar estes problemas analisados como *black-box*, ou seja, expressões explicitas das funções, com relação as variáveis de projeto, não estão disponíveis (WANG; SHAN; WANG, 2004). Métodos baseados nessa premissa são úteis pela sua generalidade e facilidade de utilização (HUANG et al., 2006). Logo, exige-se que o usuário forneça o mínimo possível de informação.

Além disso, necessita-se de um conjunto de características, capazes de expor vantagens e desvantagens de um algoritmo num problema ainda não conhecido. Para tal utiliza-se *Exploratory Landscape Analysis* (ELA). Trata-se de uma maneira de se caracterizar funções *black-box* não conhecidas através de amostragem (MERSMANN; PREUSS; TRAUTMANN, 2010).

Por fim, o processo de escolha propriamente dito vem de um mapeamento entre características de um problema e algoritmo que fornece o melhor desempenho, sendo processo fundamental para a utilização do ASP. Essa relação pode ser vista como uma tarefa de aprendizado, de modo a que se possa fazer uso de métodos de aprendizado de máquina para tal, tratando esse como um problema de classificação. Após obter o mapeamento desejado, pode-se aplicá-lo a problemas ainda não conhecidos.

A seção 2 fala sobre o ASP, detalhando seus componentes em suas subseções. A seção 3

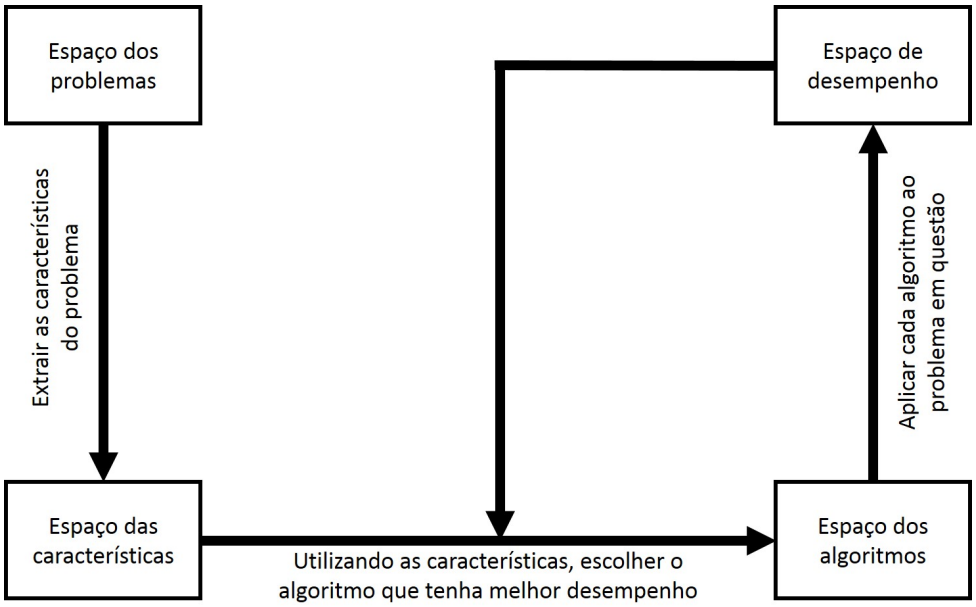

Figura 1: ASP

comenta o procedimento de análise adotado para teste da metodologia. A seção 4 apresenta os resultados obtidos, cabendo a seção 5 as conclusões pertinentes.

2 *ALGORITHM SELECTION PROBLEM*

O *Algorithm Selection Problem* (ASP) é uma estrutura desenvolvida por Rice (1976) para abordar o problemas de seleção de algoritmo, ilustrado pela questão "Com tantos algoritmos, qual tem maior chance de ser melhor para o meu problema?"(SMITH-MILES, 2008). O ASP vai de encontro com o antigo desejo de escolher o algoritmo certo para um problema de otimização não tão bem conhecido, sem enumerar todas as técnicas disponíveis (BISCHL et al., 2012).

A figura 1 ilustra a estrutura do ASP, conforme Rice (1976), sendo cada componente comentado a seguir.

1. Espaço dos problemas: Composto pela classe de problemas de interesse, como os de otimização contínuos e/ou combinatórios, entre outros;
2. Espaço dos algoritmos: Composto pelo conjunto de algoritmos aptos à solucionar os problemas de interesse;
3. Espaço das medidas de desempenho: Representam os critério que definem o desempenho de um algoritmo em particular, quando aplicado em um dado problema;
4. Espaço das características: Contém métricas extraídas de cada problema.

Percebe-se que, dado um espaço de problemas, com características representativas, algoritmos capazes de realizar o processo de otimização e medida de desempenho de interesse, a dificuldade esta em construir um modelo que seja capaz de unir todos esses elementos e selecionar o algoritmo mais adequado. Dessa forma, o problema pode ser resumido à determinar

esse mapeamento de seleção (RICE, 1976). Com o mapeamento de seleção montado, pode-se aplicar tal modelo a problemas não vistos e realizar a escolha de maneira autônoma.

2.1 Espaço dos problemas

Um problema de otimização pode apresentar inúmeras características distintas, como continuidade, número de dimensões, presença de restrições e muitos outros. Nesse trabalho, optou-se por utilizar um conjunto reduzido, composto pelas seguintes classes:

- Contínuos: Tanto função objetivo quanto variáveis de projeto são contínuas;
- Restrições do tipo *box constraints*: Fornecem limites superior e inferior às variáveis. Permitem que o problema possa facilmente ser tratado como se não possuísse restrições (YUAN; HE; LENG, 2008);
- Baixa dimensionalidade: Problemas com menos de 100 variáveis de projeto (MAHDAVI; SHIRI; RAHNAMAYAN, 2015);
- Custo de avaliação da função objetivo: Baixo;
- Formulação *black-box*: Expressões explícitas da função objetivo, com relação as variáveis de projeto, não estão disponíveis (WANG; SHAN; WANG, 2004), nem informações referentes aos seus gradientes (HUANG et al., 2006).

As 4 primeiras classes são atribuídas para reduzir o número de algoritmos de otimização necessários para cobrir os possíveis problemas, assim como reduzir o custo computacional para elaboração do método. A última vem de encontro com o interesse de reduzir a necessidade de experiência do usuário. Ao se tratar o problema como uma *black-box*, exige-se que o usuário forneça o mínimo de informação possível.

2.2 Espaço das características

As características devem ser mensuráveis, capazes de capturar alguma estrutura conhecida do problema, expor a complexidade dos mesmos e fornecer conhecimento quanto pontos fortes e fracos dos algoritmos considerados (MUÑOZ et al., 2015). Dessa forma, pretende-se que qualquer vantagem e desvantagem conhecida de diferentes algoritmos possam estar relacionadas as mesmas (SMITH-MILES, 2008). Assim, selecionar as características é um ponto crucial dessa estrutura (MUÑOZ et al., 2015), frequentemente o mais importante (RICE, 1976).

Para extrair as características de um problema *black-box* utiliza-se aqui do conceito de *Exploratory landscape analysis* (ELA). Conceitualmente, é um conjunto de técnicas utilizadas para retirar conhecimento sobre as propriedades de problemas de otimização ainda não conhecidos (BISCHL et al., 2012), principalmente no que diz respeitos às características importantes para o processo de otimização (MERSMANN et al., 2011). Grosseiramente, pode-se entender tal conceito como a representação gráfica da Hipersuperfície formada por variáveis de projeto e função objetivo (CAAMAÑO et al., 2012). Essas técnicas são concebidas para serem meios baratos e automáticos de extrair as propriedades de um problema, a partir de exemplos concretos (MERSMANN; PREUSS; TRAUTMANN, 2010).

Muitos autores utilizam métodos ELA, em especial para ASP (MCCLYMONT; WALKER; DUPENOIS, 2012), (MCCLYMONT, 2013), (CAAMAÑO et al., 2012), (MALAN; ENGELBRECHT, 2013), (MUNOZ; SMITH-MILES, 2015), (BISCHL et al., 2012), (KERSCHKE et

al., 2015). Isso se deve à observação de que, se um problema é bem conhecido, alguém pode escolher um algoritmo de otimização pertinente ao mesmo. Pelos métodos ELA podem-se diferenciar classes de problemas, gerando uma maneira automática de selecionar o algoritmo mais eficiente para cada caso.

Quando passa-se o foco para funções *black-box*, devem-se considerar que a falta de informação sobre o problema aumenta a dificuldade em computar as características das funções. Dessa forma, não há alternativas: para obter as características ELA é necessário realizar um processo de amostragem (ABELL; MALITSKY; TIERNEY, 2012).

Diversas características são citadas na literatura para indicar a complexidade de problemas de otimização. Destacam-se aqui as principais classificações encontradas, para mais detalhes, pode-se consultar (MERSMANN et al., 2011), (CAAMAÑO et al., 2012), (MERSMANN; PREUSS; TRAUTMANN, 2010), (MUÑOZ et al., 2015), (PITZER; AFFENZELLER, 2011), (PREUSS; STOEAN; STOEAN, 2011). Vale aqui a ressalva de que diferentes classificações podem apresentar relação entre si (MALAN; ENGELBRECHT, 2013).

- Separabilidade: Esta relacionada com a possibilidade de se reduzir um problema à subproblemas de menor dimensão (MERSMANN; PREUSS; TRAUTMANN, 2010). Se for possível dividir um problema de D dimensões em D subproblemas, que compartilhem mesmo ótimo global, dizemos que o mesmo é separável. Caso contrário, ele é não-separável;
- Estrutura Global: Diz respeito à forma com que os ótimos locais estão agrupados. A existência de um padrão no agrupamento destes ótimos locais constitui a estrutura global da *landscape*, que pode ser usada para guiar a otimização (MUÑOZ et al., 2015). Um problema possui uma forte estrutura global, caso seus ótimos locais possuam um agrupamento que oriente o otimizador ao ótimo global. Sua estrutura é fraca, caso não forneça orientação e pode ser *deceptive*, caso oriente para um ponto diferente do ótimo global;
- Escala Variável: Diz respeito à variações de magnitude diferente na função objetivo, dependendo da direção no espaço de busca. Caso isso ocorra, diz-se que o problema tem grande variação de escala.
- Modalidade: Divide o problema em duas classe: Unimodal, caso possua apenas um ótimo local e multimodal, caso possua mais de um ótimo.
- Suavidade: Outra medida relacionada ao número de ótimos locais, porém levando em conta também sua disposição. Um problema é suave caso uma sequência de passos em uma direção não apresente muitos ótimos locais, sendo rugosa se pontos vizinhos apresentem grande frequência de mudança entre subidas e descidas na função objetivo (PITZER; AFFENZELLER, 2011), (MUÑOZ et al., 2015)

É necessário obter formas de quantificar as características comentadas, para o caso em que as mesmas não são conhecidas, principalmente em se tratando de formulação do tipo *black-box*. Para isso, utiliza-se aqui algumas medidas relatadas na literatura.

FDC. É uma medida capaz de diferenciar funções unimodais de multimodais, além de verificar a existência de uma forte estrutura global (MUÑOZ et al., 2015). O cálculo do FDC, apresentado na eq. 1, utiliza a correlação de *Pearson* entre a distância Euclidiana dos pontos amostrados ao candidato mais apto da amostra e o valor da função objetivo dos mesmos

(MÜLLER; SBALZARINI, 2011).

$$\begin{aligned} FDC &= \frac{C_{FD}}{S_F S_D} \\ C_{FD} &= \frac{1}{n}\sum_{j=1}^{n}(f^{(j)} - \bar{f})(d^{(j)} - \bar{d}) \end{aligned} \tag{1}$$

Onde $\bar{f}, \bar{d}, S_F, S_D$, $f^{(j)}$ e $d^{(j)}$ são a média, o desvio padrão e o valor referente ao indivíduo j dos valores amostrados de função objetivo e distâncias, respectivamente, com n sendo o número de amostras realizadas. Vale ressaltar que as distâncias Euclidianas são medidas até o melhor ponto obtido na amostragem, visto que o ótimo global não é conhecido antes da resolução do problema.

O coeficiente deve ser próximo de 1 para problemas globalmente convexos e se aproximar de zero para o caso de ausência de uma estrutura global. Tal medida é invariável com relação à translações e rotações no domínio (MUÑOZ et al., 2015).

Regressões. As medidas que utilizam esse conceito baseiam-se em regressões lineares e quadráticas sobre os dados amostrados (MUÑOZ et al., 2015). Dessa forma, pode-se inferir separabilidade, multimodalidade, estrutura global e variabilidade de escala do problema (MERSMANN et al., 2011). Bischl et al. (2012) considera que o custo computacional de se obterem essas medidas é baixo.

As medidas aqui utilizadas baseiam-se em três modelos diferentes para regressão: linear, puramente quadrático e quadrático. Chama-se de puramente quadrático o modelo que não considera interação entre as variáveis de predição. Para cada um desses modelos, calcularam-se dois parâmetros:

- R^2 das aproximações, que avaliam modalidade e existência de estrutura global (MUÑOZ et al., 2015), (MUÑOZ et al., 2015), (MERSMANN et al., 2011).
- Razão entre os valores máximos e mínimos absolutos dos coeficientes da regressão. que indicam se o problema apresenta escala variável (MUÑOZ et al., 2015), (MUÑOZ et al., 2015), (MERSMANN et al., 2011).

As medidas resultantes são:

- R2L = R^2 da aproximação linear;
- R2Qp = R^2 da aproximação puramente quadrática;
- R2Q = R^2 da aproximação quadrática;
- CL = Razão entre os coeficientes máximos e mínimos absolutos da aproximação linear;
- CQp = Razão entre os coeficientes máximos e mínimos absolutos da aproximação puramente quadrática;
- CQ = Razão entre os coeficientes máximos e mínimos absolutos da aproximação quadrática;
- R2R = Razão entre R2Qp e R2Q. Medida incluida para indicar separabilidade, visto se

aproximar de 1 para ausência de interação entre variáveis.

ICoFis. É um método proposto por Muñoz et al. (2015), inicialmente para tratar as características de rugosidade e suavidade. Apesar disso, o método se mostrou eficaz para a estimativa da variação de escala do problema (MUÑOZ et al., 2015).

O Método parte de uma caminhada aleatória ao longo da *landscape*, calculando símbolos, que indicam o comportamento da função, sendo estes símbolos calculados como:

- $\bar{1}$ se $\frac{\Delta y}{\|\Delta x\|} \leq \epsilon$
- 0 se $\left|\frac{\Delta y}{\|\Delta x\|}\right| \leq \epsilon$
- 1 se $\frac{\Delta y}{\|\Delta x\|} > \epsilon$

Dois símbolos consecutivos constituem um bloco, que representa um objeto (inclinação, pico ou área neutra). O conteúdo de informação é definido pela Eq. 2.

$$H(\epsilon) = -\sum_{a \neq b} p_{ab} log_6(p_{ab}) \tag{2}$$

onde a e b são um dos três símbolos comentados e p_{ab} é a probabilidade de encontrar o bloco ab na sequência de símbolos. Realiza-se esse processo para vários valores de ϵ diferentes. Com isso gera-se a curva de entropia (H por ϵ), onde duas medidas são obtidas:

1. Hmax (Máximo conteúdo de informação), Eq. 3.

$$Hmax = max(H(\epsilon)) \tag{3}$$

2. *Es* (Máximo de ϵ para o qual a curva ainda possui alguma informação), dado pela Eq. 4.

$$Es = log_{10}(min_{\epsilon}(\epsilon : H(\epsilon < 0.05) \tag{4}$$

Utiliza-se aqui uma amostra aleatória uniformemente distribuída, sendo a caminhada gerada por ordenamento por vizinho mais próximo.

Distribuição de probabilidade. Esses métodos consideram os valores da função objetivo amostrados como uma *probability density function* (PDF) (MERSMANN et al., 2011). Essas medidas são utilizadas para avaliar multimodalidade, força da estrutura global (MERSMANN et al., 2011), (MUÑOZ; KIRLEY; HALGAMUGE, 2012) e suavidade (BISCHL et al., 2012). Bischl et al. (2012) considera essas característica como de custo computacional baixo.

Aqui, utilizam-se as seguintes medidas, calculadas sem correção de bias:

- skw: *Skewness* da distribuição de valores da função objetivo, Eq. 5.

$$skw = \frac{E(x-\mu)^3}{\sigma^3} \tag{5}$$

- kurt: *kurtosis* da distribuição de valores da função objetivo, Eq. 6.

$$kurt = \frac{E(x-\mu)^4}{\sigma^4} \tag{6}$$

Onde μ é a média e σ é o desvio padrão da variável amostrada x e E é o operador valor esperado.

Covariância. As medidas aqui utilizadas se baseiam numa ideia proposta por Caraffini, Neri e Picinali (2014). Os mesmos utilizam uma estimativa para a matriz de covariância do problema para fornecer um valor numérico que indique o grau de separabilidade do problema. Para tal, fazem uso da Eq. 7.

$$\rho_{ij} = \frac{C_{ij}}{C_{ii}C_{jj}} \tag{7}$$

Sendo C a matriz de covariância e ρ a matriz de correlações de *Pearson*. Por essa matriz ser limitada entre -1 e 1, seu uso se torna mais interessante que a matriz de covariância.

Para determinação da medida utilizada, considera-se a média dos módulos dos valores da matriz de correlação, sem contar as diagonais. Dessa forma, a medida resultante é dada pela Eq. 8.

$$Cov = \frac{2}{m^2 - m} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} |\rho_{ij}| \tag{8}$$

Onde m é o número de variáveis do problema, que é equivalente ao número de linhas e colunas das matrizes C e ρ. Percebe-se que os somatórios utilizados englobam a diagonal superior da matriz de correlação. Espera-se então que valores de Cov próximos à zero indiquem uma função separável, sendo maior o grau de não-separabilidade quanto maior for esse valor.

Para estimar a matriz de covariância do problema, utiliza-se a matriz resultante do conjunto formado pelos 10% melhores indivíduos de uma amostragem aleatória uniforme.

Percebe-se que essa medida parece adequada para medição de separabilidade de um problema de otimização, porém a matriz de covariância estimada pode ser útil também para verificar a variação de escala. Para tal, derivou-se mais 3 medidas:

1. MaxC = valor absoluto máximo da matriz de covariância;
2. MaxEig = valor absoluto máximo dos autovalores da matriz de covariância;
3. MaxMinEig = Razão entro os valores máximo e mínimo absoluto dos autovalores da matriz de covariância.

Dispersão. Indicador de multimodalidade e força da estrutura global (MUÑOZ; KIRLEY; HALGAMUGE, 2012), (MALAN; ENGELBRECHT, 2013). A dispersão é a média das distâncias entre os indivíduos mais aptos amostrados, calculada pela equação Eq. 9.

$$Disp = \frac{1}{q(q-1)} \sum_{i=1}^{q} \sum_{j=1, j\neq i}^{q} d(x_i, x_j) \tag{9}$$

Onde q é o número de amostras para o subgrupo considerado, uma porcentagem α geralmente entre 1% e 25% do número total de amostras n e x é um indivíduo pertencente a esse

grupo. O melhor poder de discriminação dessa medida é quando α tende a zero e as dimensões são menores que 10 (MUÑOZ et al., 2015). Isso pois, com o aumento das dimensões e utilizando amostragem aleatória uniforme, o valor da dispersão converge para $\frac{1}{\sqrt{6}}$ (MORGAN; MEMBER; GALLAGHER, 2014). É uma medida invariável com relação à translações e rotações no domínio (MUÑOZ et al., 2015).

Um ponto importante aqui é a relação entre características de uma função e a variação da dispersão com diferentes níveis de α. Dada uma amostra de pontos abaixo de um nível de aptidão, se uma diminuição do nível (menos indivíduos considerados) resultar num aumento da dispersão dos pontos abaixo desse nível, isso indica multimodalidade (MALAN; ENGELBRECHT, 2013). Dessa forma, julgou-se interessante calcular a dispersão para três diferentes níveis, além de considerar também possíveis iterações. Gera-se assim um total de 6 medidas:

- Disp1 = Dispersão para 5%;
- Disp2 = Dispersão para 10%;
- Disp3 = Dispersão para 25%
- Disp12 = Razão entre Disp1 e Disp2;
- Disp23 = Razão entre Disp2 e Disp3;
- Disp01 = 1 se uma diminuição de nível resultar num aumento de dispersão e 0 caso contrário.

2.3 Espaço dos algoritmos

Para a elaboração do conjunto de algoritmos, baseia-se aqui na ideia de que diferentes métodos tem pontos fortes complementares e demonstram incomparável comportamento em diferentes casos de problemas (XU et al., 2012). Sabe-se que um conjunto de algoritmos inteligentemente construído proporciona melhor eficiência que qualquer único algoritmo, mesmo em um conjunto de problemas pequeno (BISCHL et al., 2012).

Para a construção do conjunto, algoritmos complementares devem ser avaliados (MUÑOZ et al., 2015). Ao se falar em complementar, tem-se o interesse de que eles tenham desempenho ótimo em problemas com características diferentes. Para a escolha dos mesmos, baseia-se nos dados presentes em (HANSEN et al., 2010). Nesse, tem-se uma forma de se comparar o desempenho de diferentes métodos, em diversos tipos de problemas de otimização distintos e sob uma condição padronizada de análise. Os algoritmos escolhidos são:

- *LSSTEP*: É um algoritmo baseado em busca em linha, cujos detalhes do funcionamento estão em (POSIK, 2009). Este realiza a otimização de uma variável de projeto por vez, aplicando o método STEP (*Select The Easiest Point*) (SWARZBERG; SERONT; BERSINI, 1994) para cada busca unidirecional. Como esperado, quando comparado com os demais métodos, esse é muito eficiente para resolução de problemas separáveis mas não é competitivo nos demais casos.
- *Nelder-Mead*: É um método desenvolvido por (NELDER; MEAD, 1965), sendo já bastante conhecido dentro do campo de otimização. Baseia-se aqui na implementação de (HANSEN, 2009), cuja abordagem trabalha com vários reinícios propostos por (LUERSEN; RICHE, 2004), tanto globais quanto locais. Este algoritmo tem seu melhor de-

sempenho, quando comparado aos demais métodos, em pequenas dimensões, sendo este independente da separabilidade do problema.

Enquanto o *LSSTEP* tem na separabilidade a principal característica para determinar seu sucesso no processo de otimização, o *Nelder-Mead* tem o número de dimensões do problema como critério determinante. Dessa forma, com esses dois algoritmos espera-se ter um desempenho médio semelhante ao se tratar os problemas aqui desenvolvidos, visto apresentarem pontos fortes distintos.

2.4 Espaço das medidas de desempenho

A medida deve ser simples, quantitativa, bem definida, interpretável, reproduzível, justa e relevante na prática (MUÑOZ et al., 2015). Neste caso, a mesma geralmente representa acuracidade ou tempo de execução necessário para cada algoritmo. O ERT (*expected running time*) é a medida escolhida na maioria dos estudos *benchmark* (MUÑOZ et al., 2015) (MUÑOZ; KIRLEY; HALGAMUGE, 2012) (BISCHL et al., 2012). Essa medida estima o número esperado de avaliações da função objetivo necessárias para atingir uma precisão desejada, demonstrado na Eq. 10.

$$ERT(f, y^*) = \frac{FEs(y_{Melhor} \geq y^*)}{suc} \tag{10}$$

Onde $FEs(y_{Melhor} \geq y^*)$ é o numero de avaliações da função objetivo sobre todas as tentativas, onde o melhor valor y_{Melhor} não é menor que o valor alvo y^* e suc é o numero de execuções onde a precisão desejada foi alcançada, de um total de N execuções independentes. Quando nem todas as tentativas são bem sucedidas, o ERT depende fortemente do critério de parada utilizado.

2.5 Mapeamento entre características de um problema e o algoritmo mais adequado

A obtenção da relação entre características de um problema e o algoritmo mais adequado para o mesmo pode ser visto como uma tarefa de aprendizado. Neste trabalho, trata-se esse como um problema de classificação. Ou seja, dado um conjunto de características de um problema de otimização, o método associa esse conjunto à classe correspondente ao algoritmo mais adequado para otimizá-lo. O objetivo é aprender esse mapeamento e poder aplicá-lo em um problema ainda não visto.

Para essa abordagem, necessita-se de um conjunto para o treinamento do classificador. Este é composto por um conjunto de problemas de otimização, dentro do espaço definido para esse trabalho, do qual são medidas suas características e desempenho de todos os algoritmos analisados. O modelo é então treinado com as características como entrada e a classe correspondente ao melhor algoritmo nesse problema como saída. (MUÑOZ et al., 2015).

Existem inúmeros métodos para classificação supervisionada (FERNáNDEZ-DELGADO et al., 2014). Um dos métodos mais comentados na literatura é o classificador Bayesiano ingênuo (*Nayve Bayes*). Apesar de existirem métodos com maior acuracidade (FERNáNDEZ-DELGADO et al., 2014), além deste tender a ter mais problemas com atributos irrelevantes e/ou

redundantes, este apresenta como pontos positivos pouco tempo necessário para treinamento e classificação, além de ser pouco suscetível ao *overfiting* e ser de fácil uso (KOTSIANTIS, 2007), sendo essas justificativas para sua utilização aqui.

Trata-se de um processo de classificação conceitualmente bem simples. O classificador Bayesiano usa como função discriminante a probabilidade à posteriori, dado o conhecimento do vetor de características, utilizando a regra de Bayes. O classificador Bayesiano associa então o exemplo à classe que apresentar a maior probabilidade a posteriori. Para simplificar o procedimento, para um dado conjunto de treinamento, supõe-se que as características são independentes, dadas uma classe. É a partir dessa suposição que surge o nome "ingênuo".

Apesar de geralmente se tratar de uma suposição irreal, esse método mostrou grandes resultados em uma grande variedade de problemas, inclusive problemas em que sabia-se previamente que essa suposição era violada (DOMINGOS; PAZZANI, 1997)

3 METODOLOGIA DE ANÁLISE

O procedimento experimental consiste em testar o processo de escolha de algoritmo em problemas com diferentes características, porém com classificação já conhecida, para verificar o quão adequadas são essas escolhas. Os problemas utilizados foram definidos com base em um gerador, desenvolvido pelo autor, e detalhado a seguir.

3.1 Gerador de problemas

O gerador recebe como entrada o número de dimensões, ponto referente ao ótimo global e o tipo do problema e retorna como saída os parâmetros de uma função que tenha tais características. Dentre os tipos de problema, estão as combinações possível entre estrutura global, variação de escala, suavidade, separabilidade e modalidade.

Separabilidade e Escala variável são características que podem ser adicionadas à um problema através de operações sobre o sistema de coordenadas do mesmo. Já a suavidade pode ser removida somando-se um termo periódico à uma função suave. Logo, essas 3 características podem ser adicionadas à um problema base, a fim de redefini-lo da maneira que se julgue interessante. Cabe à estrutura global e modalidade definir a forma da função base, visto serem características mais relacionadas com a própria estrutura do problema.

Problema base. O problema base deve ser suave, separável e sem grande variação de escala, de modo a permitir a adição dessas características quando for de interesse. Quanto as demais características, há 3 tipos de problemas possíveis, com as combinações de estrutura global e modalidade:

- Forte e unimodal;
- Forte e multimodal;
- Fraca e multimodal.

Para forte estrutura global, optou-se por um polinômio de quarto grau em cada dimensão, ilustrado na Eq. 11. Isso se deve ao fato de poder controlar facilmente o local do ponto ótimo, além de cumprir com as demais necessidades comentadas.

$$f(y_i) = A_i y_i^4 + B_i y_i^3 + C_i y_i^2 + D_i y_i \tag{11}$$

Onde A, B, C e D são os vetores contendo os coeficientes definidos para fornecer um problema com determinada forma, i é a dimensão em questão e y é a variável após aplicadas as operações cabíveis à x, detalhadas na próxima seção.

Para o caso unimodal esse polinômio possui apenas um ponto ótimo, enquanto que para o caso multimodal ele apresenta dois pontos por dimensão. Consegue-se garantir isso a partir da derivada mostrada na Eq. 12.

$$\begin{aligned} \frac{df(y)}{dy_i} &= 4A_i y_i^3 + 3B_i y_i^2 + 2C_i y_i + D_i = 0 \\ \frac{df(y)}{dy} &= (y - y_1^*)(y - y_2^*)(y - y_3^*) = 0 \end{aligned} \tag{12}$$

Assim, a partir dos pontos ótimos y_1^*, y_2^* e y_3^*, obtém-se os coeficientes do polinômio de interesse.

Para o caso de fraca estrutura global, a função utilizada aqui é dada pela Eq. 13.

$$f(y_i) = y_i \sin(G_i y_i^2) \tag{13}$$

Onde, nesse caso, G é o único parâmetro da função. Quanto maior o valor de G, mais oscilações a função dará e mais ótimos locais o problema apresentará.

Optou-se por fornecer outra função base para os casos de fraca estrutura global e não separável, dada pela Eq. 14.

$$\begin{aligned} f(y) &= (10 - \max_{i=1}^{21} w_i \exp(-\frac{1}{2D}(y - Y_i)^2))^2 \\ w_i &= \begin{cases} 1.1 + \frac{8(i-2)}{19} \text{ se } i > 1 \\ 10, \text{ se } i = 1 \end{cases} \end{aligned} \tag{14}$$

Onde Y é uma matriz que contém 21 ótimos locais, escolhidos aleatoriamente, sendo o primeiro deles o ótimo global, w um vetor de pesos e D o número de dimensões do problema.

Operações aplicadas. Dado que a função base já foi definida pela estrutura global e modalidade, utilizam-se operadores para adequar o problema gerado as demais características. Os operadores aqui utilizados são:

- Rotação: Aplica-se ao sistema de coordenadas inicial uma rotação, de modo a tornar um problema inicialmente separável em um não-separável. Essa rotação é gerada através de uma decomposição de *gram-schmidt*, aplicada a uma matriz gerada aleatoriamente. Tem-se resultante dessa etapa um operador linear R.
- Variação de escala: Juntamente com a rotação, aplica-se uma transformação pela multiplicação de uma matriz diagonal ao sistema de coordenadas, onde a relação entre o elemento de cada linha indica a variação de escala entre essas variáveis. Dessa etapa, resulta um

operador linear V. A transformação no sistema de coordenadas é dada pela Eq. 15.

$$y = RVx \tag{15}$$

- Suavidade: A todos os problemas é somado um termo senoidal, para evitar que algumas medidas, como regressões polinomiais, venham a apresentar resultados tendenciosos. Caso a função seja suave, o coeficiente usado não será suficiente para gerar uma oscilação completa no domínio. Para o caso rugoso, a senoide terá mais de 5 oscilações no domínio. A amplitude da senoide sempre é função dos valores do problema base, de modo a resultar amplitudes inferiores a da mesma. A função resultante é dada pela Eq. 16.

$$f(y) = f_{Base}(y) + E_i \sin(F_i(y - y^*) + \frac{3\pi}{2}) \tag{16}$$

  Onde E define a amplitude e F a frequência da senoide. Chama-se atenção agora para o termo de mudança de fase aplicado. Este foi adicionado para fazer com que o termo senoidal e a função base tenham pontos mínimos coincidentes.

As operações sobre o sistema de coordenadas, dadas pela Eq. 15, resultam em que o ponto ótimo no domínio do problema é $x^* = R^{-1}V^{-1}y^*$.

3.2 Procedimento

Utilizou-se do gerador de problemas já comentado para criar 1000 exemplos distintos. Estes possuem características aleatórias, porém cobrindo todas as combinações conceitualmente possíveis. Além disso, trabalha-se com diferentes números de dimensões, variando de 2, 3, 5 e 10. Para cada problema, utilizou-se os algoritmos *Nelder-Mead* e *LSSTEP*.

Para cada algoritmo em cada problema, calculou-se o ERT como resultado de um total de 5 execuções independentes, cujo critérios de parada são o número máximo de avaliações da função objetivo superar 10^4 vezes o número de dimensões ou a solução obtida se aproximar do ótimo do problema por menos de 10^{-8}, o que ocorrer primeiro. Caso o algoritmo não obtenha exito na busca, dá-se a ele ERT igual ao número máximo de avaliações permitido.

Para cálculo das medidas ELA, utilizou-se de amostragem aleatória uniformemente distribuída, com número de amostras de 1000 vezes o número de dimensões. Considerou-se na análise apenas os problemas que foram resolvidos por algum dos algoritmos, descartando-se os demais. Isso gera um total de 883 problemas analisáveis (117 descartados).

Para o classificador, considerou-se as distribuições de probabilidade como normais. Associa-se as características ELA medidas de cada problema do conjunto gerado com a classe correspondente ao algoritmo com menor ERT. O conjunto total de dados foi dividido em 25% para teste e 75% para treinamento. Utilizou-se o conjunto de treinamento para selecionar as medidas mais relevantes, visto a baixa tolerância do classificador a atributos irrelevantes e/ou redundantes. O classificador resultante é então aplicado ao conjunto de teste e seu resultado é comparado com os casos de não se fazer escolha e com a melhor escolha possível.

4 RESULTADOS

O primeiro ponto a ser analisado é o resultado dos algoritmos nos problemas gerados, de modo a verificar o potencial da etapa de escolha de algoritmo. A tabela 1 apresenta uma

comparação entre o número de problemas em que cada algoritmo foi o melhor, o número em que cada um não atingiu o valor ótimo e o ERT total, resultado da soma dos ERT's de cada algoritmo em todos os problemas.

Tabela 1: Resultados: Visão geral

	Nelder-Mead	*LSSTEP*
Em quantos foi melhor?	462	421
Quantos não resolvidos?	186	357
ERT total (x10^7)	2.17	1.71

Pode-se observar que ambos os algoritmos tem desempenho semelhante, principalmente no ERT total e número de problemas em que o algoritmo foi melhor. Além disso, pode-se comparar agora esses resultados com o ERT total mínimo do conjunto, dado pela soma do ERT do melhor algoritmo em cada problema, que é de 4,24 x 10^6. Percebe-se que a melhor seleção possível leva cerca de 4 vezes menos tempo para resolver todos os problemas, quando comparado com o menor dos dois ERT totais. Esses pontos indicam que não se teria uma preferência clara por um único método, evidenciando a importância da etapa de seleção de algoritmo para o caso aqui mostrado.

Vale ressaltar aqui que o *LSSTEP* fornece um ERT menor que o *Nelder-Mead*, apesar de resolver menos problemas, pois aqui tem-se uma penalização relativamente baixa para os problemas não resolvidos. Dessa forma, apesar do *Nelder-Mead* ser mais robusto, ele tende a necessitar de mais avaliações para resolver seus problemas do que o *LSSTEP* precisa, quando tem sucesso no processo. Caso se aplica-se uma penalização superior, o *Nelder-Mead* poderia tornar-se o melhor algoritmo.

Outro ponto relevante agora é identificar em quais circunstâncias um algoritmo é melhor que outro, de modo a verificar se o desempenho deles condiz com o esperado. A tabela 2 compara o número de problemas em que cada algoritmo foi o melhor, em função da característica de separabilidade, enquanto a tabela 3 foca no número de dimensões dos problemas.

Tabela 2: Número de problemas em que cada algoritmo foi melhor, em função da separabilidade

	Nelder-Mead	*LSSTEP*
Número de problemas separáveis	57	421
Número de problemas não-separáveis	405	0

Pode-se notar a dependência que o *LSSTEP* tem da separabilidade, enquanto a grande maioria dos problemas que o *Nelder-Mead* foi melhor são não separáveis. As exceções, no caso de *Nelder-Mead*, são para casos de 2 ou 3 dimensões. Percebe-se também que este último tende a ter deteriorado seu desempenho com o aumento do número de dimensões. Essas informações vão de encontro com o já esperado, como visto em (HANSEN et al., 2010).

Parte-se agora para análise do classificador. Um total de 23 características estão disponíveis, organizadas da seguinte forma:

Tabela 3: Número de problemas em que cada algoritmo foi melhor, em função do número de dimensões

Dimensões	*Nelder-Mead*	*LSSTEP*
2	138	89
3	154	92
5	111	111
10	59	129

- Dimensões
- FDC
- 7 relacionadas com regressões
- 2 com o método IcoFis
- 2 com PDF
- 6 com dispersão
- 4 com covariância

O primeiro passo aqui é determinar quais as mais relevantes para o trabalho aqui proposto. Para tal, utilizou-se o erro de classificação no conjunto de treinamento como critério. A tabela 4 apresenta a comparação entre o erro de treinamento de diferentes combinações de variáveis.

Tabela 4: Erro de classificação no conjunto de treinamento

Características usadas	Erro de classificação (%)
Todas	7.2
Melhor combinação	5.2

O melhor conjunto contém 14 das 23 características analisadas, são essas: número de dimensões, CL, CQ, R2R, Hmax, Skw, Cov, MaxEig, Disp1, Disp2, Disp3, Disp12, Disp23, Disp01. Dá-se destaque aqui a importância de 3 medidas em especial, por estarem relacionadas diretamente com as características fundamentais para o desempenho dos dois algoritmos avaliados. O número de dimensões já era esperado, visto ser fundamental para determinar quando o *Nelder-Mead* é mais adequado. Já o R2R e a Cov, são indicadores de separabilidade de um problema, fator crucial para o desempenho do *LSSTEP*.

Feita essa análise, aplica-se tanto o classificador utilizando o melhor grupo de medidas quanto o com todas ao conjunto de testes. A tabela 5 apresenta uma comparação entre os resultados obtidos pelos dois classificadores com os melhores valores possíveis e o obtido caso se optasse por aplicar apenas um dos dois algoritmos em todos os problemas. Para poder ter uma comparação melhor com o valor mínimo possível, desconsiderou-se o custo de amostragem nessa tabela.

Percebe-se que ambos os classificadores apresentam valores de ERT total bem próximos

Tabela 5: Resultados no conjunto de testes

Classificação	Erro de classificação (%)	ERT Total (x10^5)
Todas as características	9.7	7.39
Melhores características	7.9	7.13
Minimo possível	0.0	6.92
Apenas *LSSTEP*	47.7	27.13
Apenas *Nelder-Mead*	52.3	43.84

do mínimo possível. O melhor classificador tem um ERT apenas 3.0% acima do mínimo, enquanto utilizar todas as variáveis chega à 6.8%. Independente de qual dos dois utilizar, ambos estão bem abaixo do ERT de cada algoritmo individualmente. Essa discrepância evidencia a importância da etapa de seleção de algoritmo.

Um ponto interessante a se notar é a proximidade do ERT total mínimo possível, mesmo com um erro de classificação relativamente grande. A tabela 6 ajuda a explicar isso, onde o erro de classificação efetivo indica quando o classificador opta por um método que não é capaz de resolver o problema. Vale ressaltar aqui que o conjunto de testes possui 176 problemas.

Tabela 6: Problemas resolvidos por cada método, no conjunto de testes

Classificador	Número de problemas resolvidos	Erro de classificação efetivo (%)
Todas as características	171	2.8
Melhores características	175	0.6
Apenas *LSSTEP*	116	34.1
Apenas *Nelder-Mead*	138	21.6

Percebe-se que os classificadores utilizados tem um erro de classificação efetivo muito menor do que o erro utilizado, que considera como única escolha certa o melhor algoritmo. Dessa forma, a maior parte dos erros de classificação cometidos são, na verdade, em problemas em que o desempenho dos dois métodos é muito semelhante e o custo de se errar é pequeno. Assim, o erro de classificação utilizado não parece a melhor forma de quantificar o desempenho, mas sim deve-se associar o custo da escolha no método de aprendizagem.

Para justificar plenamente o processo de escolha, é necessário ainda considerar o custo de amostragem ao ERT total. A tabela 7 apresenta os valores de ERT de cada classificador, agora adicionando o custo de amostragem.

Nota-se que o ERT dos classificadores praticamente dobrou, porém ainda estão bem inferiores ao melhor algoritmo. Como os problemas avaliados são consideravelmente simples, espera-se que o custo de amostragem seja ainda menos significativo ao se adicionar mais algoritmos e problemas. Dessa forma, o método de escolha aqui proposto conseguiu desempenhar muito bem na tarefa de seleção de algoritmo, para os problemas aqui analisados.

Tabela 7: Resultados no conjunto de testes

Classificação	ERT Total (x10^5)
Todas as características	15.74
Melhores características	15.15
Minimo possível	6.92
Apenas *LSSTEP*	27.13
Apenas *Nelder-Mead*	43.84

5 CONCLUSÃO

Após o aqui exposto, percebe-se a importância que a etapa de seleção de algoritmo tem para o sucesso de um processo de otimização. Apesar de simples, os exemplos aqui apresentados deixam claro o quão vantajoso é realizar uma etapa de seleção de algoritmo adequada. O método não só reduz o tempo do processo, como também aumenta as garantias que o resultado obtido será satisfatório.

O principal ponto a se exaltar aqui é que o método trás essa redução sem a necessidade de iteração direta do usuário ou inserção de conhecimento prévio sobre características do problema em questão. Todo o trabalho feito parte do pressuposto que as funções são *black-box*, realizando um processo de amostragem aleatório para a medição das características do mesmo.

Além disso, mesmo considerando-se o custo de amostragem ao tempo esperado de execução de cada algoritmo, o resultado ainda se mostrou favorável ao método aqui desenvolvido. Isso é ainda mais relevante quando percebe-se que os problemas aqui analisados são relativamente simples e o ERT dos algoritmos tende a ser baixo.

Apesar disso, ainda há muito trabalho pela frente. Primeiramente, o conjunto de algoritmos aqui considerado é muito limitado, devendo-se incluir mais algoritmos com capacidades complementares, principalmente voltados para problemas mais complexos e de maiores dimensões. Como mais de 10% dos problemas gerados não foram resolvidos por nenhum dos dois métodos, fica uma lacuna em aberto para inserção de novos algoritmos.

Ficou evidente também que o erro de classificação não é a forma mais adequada de se treinar o classificador, mas sim deve-se considerar o custo do erro de classificação na análise. Pode-se pensar em alterar o método aqui utilizado ou buscar outros classificadores, que possam apresentar um desempenho melhor.

Por fim, o conjunto de problemas elaborado também é limitado, devendo o desempenho do método proposto ser avaliado em funções com estrutura distintas das produzidas pelo gerador aqui utilizado. Só assim pode-se verificar se o método é realmente robusto.

Os resultados obtidos são promissores, indicando que a estrutura aqui trabalhada pode ser muito útil para tratar o processo de seleção de algoritmo de forma automatizada. Porém, esta ainda precisa ser complementada.

AGRADECIMENTOS

Os autores cordialmente agradecem o apoio da CAPES e do CNPQ.

Referências Bibliográficas

ABELL, T.; MALITSKY, Y.; TIERNEY, K. Fitness Landscape Based Features for Exploiting Black-Box Optimization Problem Structure. n. December, 2012.

BISCHL, B. et al. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12*, p. 313–320, 2012.

CAAMAÑO, P. et al. Experimental analysis of the relevance of fitness landscape topographical characterization. *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, p. 10–15, 2012.

CARAFFINI, F.; NERI, F.; PICINALI, L. An analysis on separability for memetic computing automatic design. *Information Sciences*, p. 1–22, 2014.

CARCHRAE, T.; BECK, J. C. Applying machine learning to low-knowledge control of optimization algorithms. *Computational Intelligence*, v. 21, n. 4, p. 372–387, 2005. ISSN 08247935.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, n. 2-3, p. 103–130, nov. 1997. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1007413511361>.

FERNáNDEZ-DELGADO, M. et al. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, v. 15, p. 3133–3181, 2014. Disponível em: <http://jmlr.org/papers/v15/delgado14a.html>.

HANSEN, N. *Benchmarking the Nelder-Mead Downhill Simplex Algorithm With Many Local Restarts*. [S.l.], 2009.

HANSEN, N. et al. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2010. (GECCO '10), p. 1689–1696. ISBN 978-1-4503-0073-5. Disponível em: <http://doi.acm.org/10.1145/1830761.1830790>.

HUANG, D. et al. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, v. 34, n. 3, p. 441–466, 2006. ISSN 09255001.

KERSCHKE, P. et al. Detecting Funnel Structures by Means of Exploratory Landscape Analysis. *Proceedings of the 2015 Genetic and Evolutionary Computation Conference - GECCO '15*, p. 265–272, 2015. Disponível em: <http://dl.acm.org/citation.cfm?doid=2739480.2754642>.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 3–24. ISBN 978-1-58603-780-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1566770.1566773>.

LUERSEN, M. A.; RICHE, R. L. L. Globalized nelder-mead method for engineering optimization. *Computer and Structures*, v. 83, n. 23-26, p. 2251âĂŞ2260, aug 2004.

MAHDAVI, S.; SHIRI, M. E.; RAHNAMAYAN, S. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, v. 295, p. 407–428, 2015. ISSN 00200255.

MALAN, K. M.; ENGELBRECHT, A. P. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, Elsevier Inc., v. 241, p. 148–163, 2013. ISSN 00200255. Disponível em: <http://dx.doi.org/10.1016/j.ins.2013.04.015>.

MCCLYMONT, K. Recent advances in problem understanding. n. 24, p. 1071–1077, 2013. Disponível em: <http://dx.doi.org/10.1145/2464576.2482685>.

MCCLYMONT, K.; WALKER, D.; DUPENOIS, M. The Lay of the Land : a Brief Survey of Problem Understanding. *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO '12)*, p. 425–432, 2012. Disponível em: <http://doi.acm.org/10.1145/2330784.2330849>.

MERSMANN, O. et al. Exploratory Landscape Analysis. *Gecco-2011: Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference*, p. 829–836, 2011.

MERSMANN, O.; PREUSS, M.; TRAUTMANN, H. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6238 LNCS, n. PART 1, p. 73–82, 2010. ISSN 03029743.

MORGAN, R.; MEMBER, S.; GALLAGHER, M. Analysis : Limitations and Improvements. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 3, p. 456–461, 2014.

MÜLLER, C. L.; SBALZARINI, I. F. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6624 LNCS, n. PART 1, p. 294–303, 2011. ISSN 03029743.

MUÑOZ, M. a.; KIRLEY, M.; HALGAMUGE, S. K. A meta-learning prediction model of algorithm performance for continuous optimization problems. *Parallel Problem Solving from Nature - PPSN XII*, v. 7491, n. PART 1, p. 226–235, 2012. ISSN 03029743.

MUÑOZ, M. A. et al. Exploratory landscape analysis of continuous space optimization problems using information content. v. 19, n. 1, p. 74–87, 2015.

MUNOZ, M. A.; SMITH-MILES, K. Effects of function translation and dimensionality reduction on landscape analysis. *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, p. 1336–1342, 2015.

MUÑOZ, M. A. et al. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, v. 317, p. 224–245, 2015. ISSN 00200255.

NELDER, J.; MEAD, R. Simplex method for function minimization. *The Computer Journal*, v. 7, n. 4, p. 308–313, jan 1965.

PITZER, E.; AFFENZELLER, M. A comprehensive survey of Fitness Landscape Analysis. *Recent Advances in Intelligent Engineering Systems*, p. 161–190, 2011.

POSIK, P. Bbob-benchmarking two variants of the line-search algorithm. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. New York, NY, USA: ACM, 2009. (GECCO '09), p. 2329–2336. ISBN 978-1-60558-505-5. Disponível em: <http://doi.acm.org/10.1145/1570256.1570325>.

PREUSS, M.; STOEAN, C.; STOEAN, R. Niching foundations. *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, p. 837, 2011.

RICE, J. R. The algorithm selection problem. *Advances in Computers*, v. 15, n. C, p. 65–118, 1976. ISSN 00652458.

SMITH-MILES, K. A. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, v. 41, n. 1, p. 1–25, 2008. ISSN 03600300.

SWARZBERG, S.; SERONT, G.; BERSINI, H. S.t.e.p.: the easiest way to optimize a function. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. [S.l.: s.n.], 1994. p. 519–524 vol.1.

WANG, L.; SHAN, S.; WANG, G. G. Mode-pursuing sampling method for global optimization on expensive black-box functions. *Engineering Optimization*, v. 36, n. 4, p. 419–438, 2004. ISSN 0305-215X.

XU, L. et al. to Portfolio-Based Algorithm Selectors. p. 228–241, 2012.

YUAN, Q.; HE, Z.; LENG, H. A hybrid genetic algorithm for a class of global optimization problems with box constraints. *Applied Mathematics and Computation*, v. 197, n. 2, p. 924–929, 2008. ISSN 00963003.