



## UMA HEURÍSTICA PARA OTIMIZAÇÃO DE META-HEURÍSTICAS POR MEIO DE MÉTODOS ESTATÍSTICOS

**Eduardo Batista de Moraes Barbosa**

Instituto Nacional de Pesquisa Espaciais (INPE)  
Rod. Presidente Dutra, Km. 40 - Cachoeira Paulista, SP - 12630-000, CPTEC/INPE  
[eduardo.barbosa@inpe.br](mailto:eduardo.barbosa@inpe.br)

**Edson Luiz França Senne**

Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP)  
Av. Dr. Ariberto Pereira da Cunha, 333 - Guaratinguetá, SP - 12516-410, FEG/UNESP  
[edson.senne@unesp.br](mailto:edson.senne@unesp.br)

### RESUMO

A sintonização dos parâmetros em algoritmos, especialmente, nas meta-heurísticas, nem sempre é trivial. Embora o uso de métodos robustos para a sintonização de meta-heurísticas tenha a sua importância reconhecida, ainda é possível encontrar o uso de abordagens informais (por exemplo, tentativa e erro) na sintonização desses algoritmos. Conseqüentemente, os resultados são pouco confiáveis, de baixa replicação, propensos a erros de configuração e custosos. O presente artigo apresenta uma abordagem para o problema da sintonização dos parâmetros de meta-heurísticas. A ideia-chave é um método heurístico que combina técnicas de Estatística e Inteligência Artificial para explorar o espaço de busca dos parâmetros procurando uma alternativa promissora. A validade da abordagem proposta é confirmada a partir de um estudo de caso para sintonização do Algoritmo Genético para resolução de um problema clássico de otimização. Os resultados são comparados considerando a mesma meta-heurística sintonizada por um método Corrida. De modo geral, a abordagem proposta mostrou-se eficaz em termos do tempo global do processo de sintonização.

**PALAVRAS CHAVE.** Meta-heurísticas. Sintonização. Otimização Combinatória.

**Tópicos:** Meta-heurísticas, Otimização Combinatória, Estatística.

### ABSTRACT

The fine-tuning of the algorithms parameters, especially in metaheuristics, is not always trivial. Although the use of robust methods for the fine-tuning of metaheuristics has its importance is recognized, it is still possible to find the use of informal approaches (e.g.: trial and error) in the fine-tuning of these algorithms. Consequently, the results are unreliable, of low replication, prone to configuration errors, and costly. This paper presents an approach to the problem of fine-tuning metaheuristics by means of a heuristic method. The key idea is a heuristic method which combines Statistical and Artificial Intelligence techniques to explore the search space of parameters looking for a promising alternative. The validity of the proposed approach is confirmed through a case study to tune a Genetic Algorithm on a classic optimization problem. The results are compared considering the same metaheuristic tuned by means of a Racing method. Broadly, the proposed approach proved effective in terms of the overall time of the fine-tuning process.

**KEYWORDS.** Metaheuristics. Fine-tuning. Combinatorial Optimization.

**Topics:** Metaheuristics, Combinatorial Optimization, Statistics.



## 1. Introdução

A sintonização dos parâmetros em algoritmos, especialmente, nas meta-heurísticas, nem sempre é trivial e, usualmente, é realizada *ad hoc* de acordo com o problema sob análise. Frequentemente, a definição incorreta de parâmetros pode provocar, em casos extremos, um comportamento inesperado do algoritmo e fazê-lo convergir rapidamente para uma solução ótima local. Em outras situações, o comportamento pode ser aleatório, tal que o algoritmo não converge para uma solução no limite de tempo aceitável. Independente da situação, muito esforço computacional é desperdiçado com soluções de baixa qualidade.

Há muitos desafios relacionados a sintonização dos parâmetros das meta-heurísticas (por exemplo, domínio dos parâmetros, estratégia de sintonização, etc.) que, em geral, exigem a utilização de metodologias inovadoras. Esses desafios são de interesse de diferentes comunidades de pesquisa e, por isso, é possível encontrar na literatura contemporânea uma diversidade de trabalhos que descrevem abordagens variadas para solucioná-los [Dobslaw, 2010; Lessman et al., 2011; Neumüller et al., 2011; Ries et al., 2012; Akbaripour e Masehian, 2013; Amoozegar e Rashedi, 2014; Calvet et al., 2016; dentre outros]. Muitas dessas pesquisas empregam técnicas de Estatística a fim de auxiliar a compreensão do processo de sintonização, bem como para encontrar configurações efetivas.

O presente estudo tem o objetivo de contribuir com a literatura apresentando uma metodologia para sintonização dos parâmetros de meta-heurísticas, que combina técnicas de Estatística e de Inteligência Artificial, tais como o Planejamento de Experimentos (em Inglês, *Design of Experiments* - DOE) [Montgomery, 2012] e o conceito de algoritmos de Corrida (em Inglês, *Racing algorithms*) [Maron e Moore, 1994; Birattari et al., 2002]. A ideia-chave desta proposta é considerar que as configurações candidatas pertencem a um espaço de busca e concentrar as buscas sobre as alternativas promissoras neste espaço. Em síntese, a presente abordagem concentra as buscas sobre alternativas dinamicamente criadas em um processo iterativo e emprega o conceito de Corrida para avaliar e descartar as alternativas estatisticamente inferiores, assim que houver evidências contra elas.

Na literatura de meta-heurísticas é possível encontrar diversos métodos de sintonização, dentre os quais se destacam CALIBRA [Adeson-Díaz e Laguna, 2006], que utiliza DOE para definir um espaço de busca para parâmetros, F-Race [Birattari et al., 2002 e 2009] e sua versão iterativa I/F-Race [Balaprakash et al., 2007], que empregam algoritmo de Corrida e estatística não-paramétrica na avaliação das configurações candidatas presentes em um espaço de busca, e ParamILS [Hutter et al., 2009], em que as alternativas são criadas a partir de perturbações nos parâmetros.

As características dos diferentes métodos de sintonização da literatura podem ser identificadas no método heurístico proposto neste estudo. A vantagem de combinar as diferentes características pode ser sumarizada na habilidade de definir o espaço de busca, eficiência para avaliar a qualidade das configurações candidatas e capacidade de concentrar as buscas sobre alternativas promissoras dentro desse espaço de busca.

A validade da abordagem proposta é confirmada a partir de resultados de um estudo de caso com um Algoritmo Genético, e sua efetividade é comparada com outra metodologia de sintonização, isto é, um algoritmo de Corrida inspirado no método F-Race [Birattari et al., 2002]. A qualidade das configurações propostas é avaliada por meio da comparação das meta-heurísticas aplicadas na solução de um problema clássico de otimização combinatória, tal qual o Problema do Caixeiro Viajante.

O restante deste artigo está organizado da seguinte maneira: a Seção 2 apresenta o problema da sintonização dos parâmetros de meta-heurísticas e a abordagem proposta neste estudo, que combina métodos de Estatística e Inteligência Artificial para resolvê-lo. Na Seção 3 há uma breve apresentação do Algoritmo Genético e do problema selecionado para o estudo de caso. A Seção 4 ilustra a aplicação da abordagem proposta na sintonização do Algoritmo Genético. Esta seção também apresenta a análise dos resultados. As considerações finais estão na Seção 5.



## 2. O Problema de Sintonização de Meta-heurísticas

Informalmente, o problema de sintonização de meta-heurísticas consiste em determinar valores dos parâmetros, que permitem um algoritmo atingir o melhor desempenho possível em tempo aceitável. A sintonização de meta-heurísticas é, por si só, um problema de otimização, em que o objetivo é otimizar um algoritmo (por exemplo, melhorar seu desempenho, aumentar a qualidade da solução, etc.) para resolução de diferentes problemas [Blum e Roli, 2003; Talbi, 2009].

Na literatura contemporânea é possível encontrar definições formais para este problema [Birattari et al., 2002 e 2009; Smit e Eiben, 2009]. Neste estudo, o problema é formalizado da seguinte maneira: seja  $M$  uma meta-heurística qualquer, com um conjunto de parâmetros, aplicada na resolução de problemas da classe  $P$ . Os parâmetros  $(\alpha, \beta, \dots, \xi)$  de  $M$  podem assumir um conjunto finito de valores e sua cardinalidade pode variar extensivamente de acordo com  $M$  e  $P$  estudados. Considere  $\Theta$  um conjunto de configurações candidatas, tal que  $\theta$  é uma configuração candidata de  $M$ . Portanto, o problema de sintonização de meta-heurísticas é formalizado como um espaço de estados:

$$S = (\Theta, P), \quad (1)$$

Em síntese, o problema consiste em descobrir qual a melhor  $\theta \in \Theta$  para  $M$  resolver  $P$ . A partir desta formalização pode-se determinar o número total de experimentos que será realizado durante o processo de sintonização, por meio do produto  $(|\alpha| \times |\beta| \times \dots \times |\xi|) \times |P|$ . Por exemplo, seja  $M$  uma meta-heurística com quatro parâmetros:  $A (a_1, a_2, a_3)$ ,  $B (b_1, b_2, b_3, b_4)$ ,  $C (c_1, c_2, c_3)$  e  $D (d_1, d_2, d_3, d_4, d_5)$  e  $P$  uma classe de problemas contendo 50 exemplares, tal que  $|P| = 50$ . Cada possível combinação de parâmetros corresponde a uma configuração diferente de  $M$ , por exemplo,  $(a_1, b_1, c_1, d_1) \neq (a_1, b_1, c_1, d_2)$ , tal que o número de experimentos do processo de sintonização de  $M$  será  $(3 \times 4 \times 3 \times 5) \times 50 = 9000$ . Portanto, a melhor configuração de  $M$  para  $P$  é uma alternativa em (1), tal que sua determinação, na pior hipótese, se dá a partir de uma busca completa em  $S$ .

### 2.1 Heurística Orientada por Algoritmo de Corrida

A abordagem proposta neste estudo visa evitar a busca completa em (1) e ainda assim encontrar uma boa sintonização para os parâmetros de  $M$ . Para isso, esta abordagem combina métodos de DOE e o conceito de Corrida, respectivamente, para encontrar consistentemente boas configurações candidatas com base em avaliações estatísticas de uma ampla gama de problemas da classe  $P$ .

O processo de sintonização é iniciado a partir da seleção arbitrária de  $n$  exemplares ( $n > 1$ ) de uma classe de problemas de otimização, e segue com as definições de valores, isto é, dos intervalos que cada parâmetro da meta-heurística alvo pode assumir. Os exemplares previamente selecionados são tratados como um conjunto de treinamento inicial, sobre o qual são realizados estudos experimentais com a Metodologia de Superfície de Respostas para determinar a sintonização dos parâmetros da meta-heurística alvo. Portanto, os resultados dos estudos experimentais são  $n$  configurações para cada parâmetro, sendo cada uma delas relacionada a um exemplar do conjunto de treinamento. O conjunto de treinamento têm a finalidade de diversificar valores para os parâmetros, e permitir a identificação de um intervalo promissor para cada um deles, denominado espaço de busca de configurações candidatas, cujos limites são o máximo e o mínimo de cada parâmetro após os estudos experimentais. A partir daí, o objetivo é buscar alternativas, que são criadas dinamicamente em um processo iterativo na vizinhança de uma configuração candidata promissora, considerando os limites do espaço de busca. Para cada alternativa criada, a meta-heurística alvo é executada sobre um conjunto expandido de exemplares.



O processo descrito acima é chamado Heurística Orientada por Algoritmo de Corrida (HORA) (Figura 1), devido à maneira de explorar as alternativas no espaço de busca, ou seja, com um método heurístico, e a avaliação das configurações candidatas, isto é, por meio de um algoritmo de Corrida.

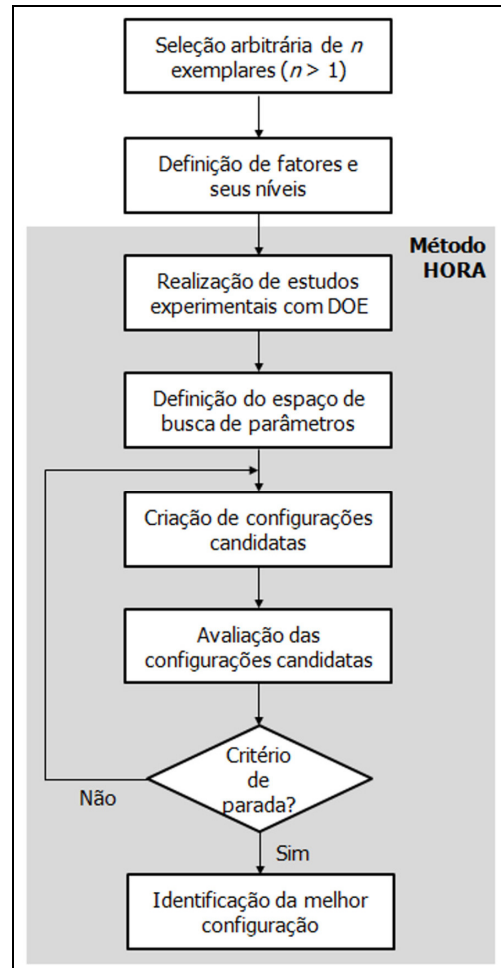


Figura 1. Esquema da abordagem proposta.

## 2.2 Dinâmica do Conjunto de Configurações Candidatas

A estratégia mais intuitiva para resolver o problema de sintonização de meta-heurísticas é a força bruta. Nessa estratégia o poder computacional é atribuído uniformemente para todas as configurações candidatas previamente definidas em (1), uma vez que executa o mesmo número de experimentos para cada alternativa. Entretanto, configurações candidatas de qualidade inferior são testadas como aquelas consideradas boas.

Para evitar este problema, o método de Corrida tradicional emprega uma estatística robusta para avaliar as configurações candidatas durante o processo de sintonização e descartar aquelas consideradas estatisticamente inferiores, assim que coletar evidências suficientes contra elas.

Mesmo considerando a eficiência do método de Corrida na avaliação das alternativas, ambas abordagens (força bruta e Corrida) iniciam o processo de sintonização a partir de um conjunto grande de configurações candidatas. Portanto, de acordo com o tamanho desse conjunto, sua avaliação é inicialmente lenta.

Diferente das abordagens tradicionais, no HORA o conjunto de configurações candidatas é criado dinamicamente durante o processo de sintonização. Isto é, as alternativas são



criadas sob demanda na vizinhança de uma configuração candidata promissora, como uma sequência de conjuntos de configurações candidatas:

$$\Theta_0 \Rightarrow \Theta_1 \Rightarrow \Theta_2 \Rightarrow \dots$$

A partir da iteração  $k$  para  $k+1$ , o conjunto  $\Theta_k$  dá origem a um novo conjunto  $\Theta_{k+1}$  possivelmente descartando as alternativas consideradas estatisticamente inferiores. Dado que algumas alternativas persistem em  $\Theta$ , elas são testadas sobre um número maior exemplares. Portanto, todas as alternativas criadas durante o processo de sintonização devem ser avaliadas sobre os mesmos exemplares previamente utilizados nas avaliações das configurações candidatas persistentes.

A dinâmica do conjunto de configurações candidatas é ilustrada na Figura 2. Seja um espaço de estados qualquer em que a cada iteração  $k$ , são criadas  $m = 3$  configurações candidatas. Ao final de uma iteração  $k$ , todas as alternativas presentes em  $\Theta$  são avaliadas e aquelas com qualidade inferior são descartadas. Por isso, a cardinalidade de  $\Theta$  é dinâmica, isto é, aumenta ou diminui. Esta dinâmica é ilustrada na Figura 2, em que as alternativas incluídas no processo de sintonização são apresentadas em preto e as excluídas, em vermelho. O processo segue em ciclo na busca por novas alternativas até atingir um critério de parada (por exemplo, número de alternativas em  $\Theta$ , tempo de execução, entre outros).

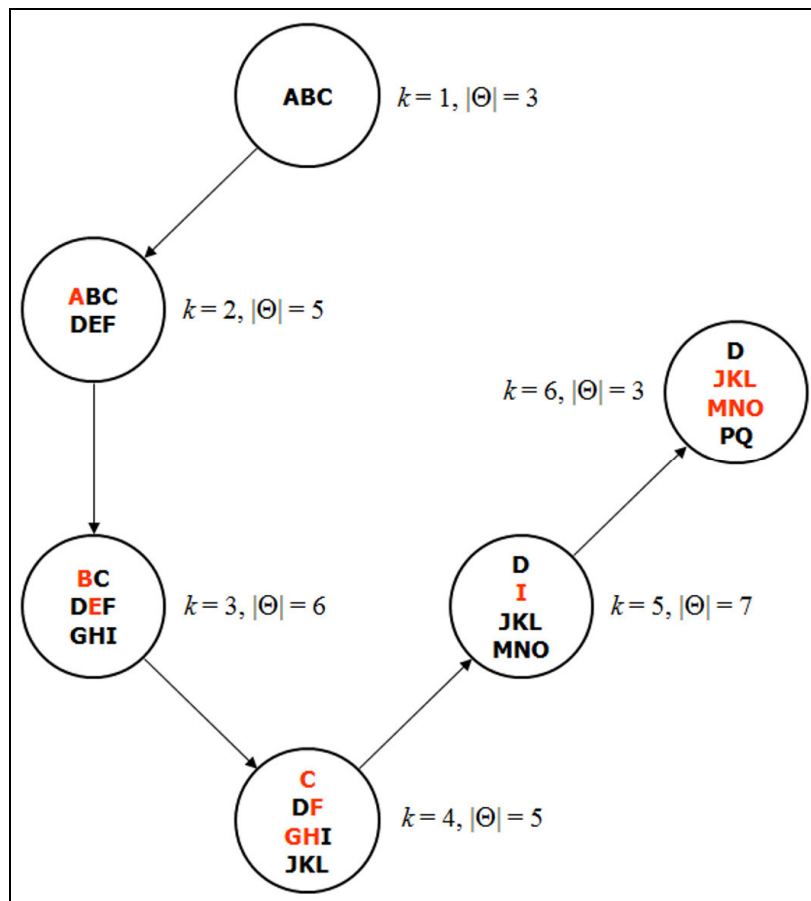


Figura 2. Processo dinâmico de atualização do conjunto de configurações candidatas.

Durante o processo iterativo as alternativas são classificadas separadamente em blocos, isto é, por exemplar, de acordo com o seu desempenho (por exemplo, valor da função objetivo), tal que a configuração com o melhor desempenho tem a classificação 1, a segunda melhor tem a classificação 2, e assim sucessivamente. Em caso de empates entre diferentes configurações



atribui-se o valor médio das classificações a cada um dos candidatos. Uma descrição detalhada sobre o processo de avaliação por meio do método de Corrida utilizando a estatística não paramétrica de Friedman pode ser encontrada em Birattari et al. [2002 e 2009].

### 3. Sintonização de Meta-heurísticas para o Problema do Caixeiro Viajante

Muitas meta-heurísticas são inspiradas em metáforas de diferentes áreas de conhecimento, como a física (por exemplo, otimização de enxame de partículas e recozimento simulado) e biologia (por exemplo, algoritmos genéticos e redes neurais). Geralmente, esses algoritmos se diferenciam em termos do padrão de busca empregado, mas oferecem métodos precisos e equilibrados para promover a diversificação (exploração do espaço de busca) e a intensificação (exploração de uma região promissora), compartilham características como o uso de componentes estocásticos (envolvendo a aleatoriedade das variáveis), e possuem uma diversidade de parâmetros que devem ser sintonizados de acordo com o problema em estudo.

Embora a sintonização de meta-heurísticas tenha sua importância reconhecida, ainda é possível encontrar o uso de abordagens informais, de tentativa e erro, e a utilização de conhecimentos empíricos para a sintonização dos parâmetros desses algoritmos. Essas abordagens, além de extremamente entediadas, são pouco confiáveis, de baixa replicação, propensas a erros de configuração e custosas.

Este estudo irá ilustrar a aplicação do método HORA na sintonização de um Algoritmo Genético [Holland, 1975] para resolução do Problema do Caixeiro Viajante.

#### 3.1 Problema do Caixeiro Viajante

Um dos problemas de rotas mais célebres é o Problema do Caixeiro Viajante (em Inglês, *Traveling Salesman Problem* - TSP). A simplicidade da concepção e sua aparente intratabilidade fazem o TSP desempenhar um papel de referência na literatura. As principais técnicas algorítmicas em otimização combinatória têm o TSP como um dos problemas de *benchmark* motivadores. Exceto para alguns casos especiais, a resolução do TSP é difícil, tal qual o problema de determinar um ciclo Hamiltoniano em um grafo.

A literatura relacionada ao TSP é extensa devido à sua relevância prática [Applegate, 2007; Laporte, 2007; Derigs, 2009; Matali et al., 2010], bem como por ele ocorrer como subproblema de diversos problemas de rotas.

O enunciado genérico do TSP consiste em um conjunto com  $n$  cidades  $C = \{1, 2, \dots, n\}$ , em que a cada par de cidades  $(i, j)$  é associado a uma distância  $d_{ij}$ , tal que  $(i, j) \in C$  com  $i \neq j$ . O objetivo é visitar todas as cidades uma única vez e voltar à cidade de partida, de forma a minimizar a distância total percorrida. O aumento no número de cidades torna a determinação dos percursos mais complexa.

O TSP é classificado como simétrico se  $d_{ij} = d_{ji}$ ,  $\forall i, j \in C$ , ou assimétrico, se  $d_{ij} \neq d_{ji}$  para ao menos um par de cidades. O problema pode ser definido em um grafo não direcionado completo  $G = (C, E)$  se for simétrico ou em um grafo direcionado  $G = (C, A)$  se for assimétrico. Os exemplares do TSP, que formam o conjunto de treinamento do estudo de caso são simétricos com distância Euclidiana entre as cidades, e foram selecionados arbitrariamente do *benchmark* TSP da biblioteca TSPLIB [Reinelt, 1991].

#### 3.2 Algoritmo Genético

O Algoritmo Genético (em Inglês, *Genetic Algorithm* - GA) é uma meta-heurística de otimização numérica inspirada no processo de seleção natural e na genética [Coley, 1999]. Esses algoritmos são de base populacional e caracterizam um modelo computacional do processo evolutivo [Blum e Roli, 2003; Talbi, 2009].

Os GA operam sobre uma população (conjunto de soluções de um problema) e fornecem uma maneira intuitiva para explorar o espaço de soluções [Mitchell, 1998].





Originalmente, o GA proposto em Holland (1975) foi inspirado no princípio de sobrevivência enunciado a partir da teoria da evolução das espécies de Darwin em 1858.

Um típico GA utiliza um conjunto de operadores para (i) selecionar indivíduos de uma população com base em sua aptidão, (ii) recombinar e (iii) modificar esses indivíduos selecionados, para produzir novas gerações com indivíduos mais adaptados. Sucintamente, a ideia dos operadores é transformar uma população através de gerações sucessivas, mantendo as características de adaptação das gerações ascendentes. A estrutura de um GA em alto nível de abstração é ilustrada no pseudocódigo da Figura 3.

```
Entrada:  $\mu, p_c, p_m$  // tamanho da população, taxas de recombinação e mutação  
Saída:  $P^*$  // Melhor solução encontrada  
1.  $P \leftarrow \{\}$   
2.  $P^* \leftarrow ()$   
3. Para  $\mu$  vezes Faça  
4.  $P \leftarrow P \cup \{\text{novo indivíduo}\}$   
5. Repetir  
6. Para cada  $p' \in P$  Faça  
7. Se  $\text{aptidão}(p') > \text{aptidão}(P^*)$  Então  
8.  $P^* \leftarrow p'$   
9.  $Q \leftarrow \{\}$   
10. Para  $\mu$  vezes Faça  
11.  $P_i \leftarrow \text{selecionar}(P)$   
12.  $P_j \leftarrow \text{selecionar}(P)$   
13.  $(C_i, C_j) \leftarrow \text{cruzamento}(P_i, P_j, p_c)$   
14.  $(M_i, M_j) \leftarrow \text{mutação}(C_i, C_j, p_m)$   
15.  $Q \leftarrow Q \cup \{M_i, M_j\}$   
16.  $P \leftarrow Q$   
17. Até encontrar um critério de parada  
18. Retornar  $P^*$ 
```

Figura 3. Pseudocódigo do GA.

Além do conjunto de operadores, o GA também possui parâmetros que influenciam fortemente o desempenho do algoritmo, tais como o tamanho da população e as taxas de recombinação e mutação. O tamanho da população deve ser definido adequadamente, pois se for pequeno pode ser insuficiente para explorar o espaço de soluções, mas se for muito grande, pode demandar tempo computacional excessivo. Em geral, a recombinação é realizada com alta probabilidade, enquanto a mutação ocorre com probabilidade mais baixa. Sem recombinação, a aptidão da população deve convergir até se igualar à aptidão do melhor indivíduo. A partir desse ponto, só poderá haver melhoria por meio de mutação.

#### 4. Estudo de Caso

O GA desenvolvido para este estudo de caso é iniciado a partir de uma população, cujos indivíduos são gerados arbitrariamente. O processo de seleção adotado é conhecido como amostragem da roda da roleta (em Inglês, *roulette wheel*) [Goldberg, 1989], que é equivalente a distribuir os indivíduos em um círculo proporcionalmente a sua aptidão. O procedimento consiste em somar a aptidão de todos indivíduos ( $A$ ) e gerar um número aleatório ( $r$ ) entre  $[0; A]$ . Em seguida, uma soma parcial das aptidões ( $a$ ) é iniciada e o indivíduo para o qual o valor  $a$  exceder  $r$  é selecionado. Este procedimento é repetido até que sejam selecionados todos os indivíduos para compor a população da nova geração.

O operador de recombinação (ou cruzamento) emprega o método de corte em dois pontos (em Inglês, *two-point crossover*) [Mitchell, 1998]. Nesse método, dois pontos são selecionados ao acaso nos indivíduos progenitores e todos os genes entre estes pontos são trocados entre os progenitores, resultando os indivíduos (filhos) da nova geração (Figura 4). No



entanto, em certas ocasiões, o processo de recombinação precisa ser reparado para evitar violações nas restrições do problema (por exemplo, a repetição de cidades em um ciclo no TSP).

<b>Progenitor<sub>1</sub></b>								<b>Progenitor<sub>2</sub></b>							
0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0
<b>Filho<sub>1</sub></b>								<b>Filho<sub>2</sub></b>							
0	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0

Figura 4. Recombinação em dois pontos.

A mutação de indivíduos da população é realizada a partir de uma inversão simples de sequência (em Inglês, *simple inversion mutation*) [Larrañaga et al., 1999] entre dois pontos selecionados ao acaso (Figura 5).

<b>Indivíduo (original)</b>								<b>Indivíduo (novo)</b>							
0	1	1	1	1	0	0	1	0	1	1	0	0	1	1	1

Figura 5. Mutação de inversão simples.

O estudo de sintonização foi realizado com os parâmetros do GA mais frequentemente utilizados na literatura, isto é, taxa de recombinação ( $p_c$ ), taxa de mutação ( $p_m$ ), tamanho da população ( $\mu$ ) e número de gerações ( $g$ ). Os valores (mínimo e máximo) de cada parâmetro (Tabela 1), que no contexto de DOE são chamados de níveis (alto e baixo), foram escolhidos de acordo com exaustivos experimentos com a meta-heurística. A partir desses valores espera-se promover diversidade no espaço de busca e, conseqüentemente, diferenças entre as configurações candidatas que serão avaliadas.

Tabela 1. Configurações de parâmetros para os estudos experimentais.

Parâmetro	Baixo	Alto
$p_c$	0,400	0,900
$p_m$	0,001	0,050
$\mu$	50	500
$g$	500	2500

Para este estudo utilizou-se um conjunto de treinamento com  $m = 5$  exemplares arbitrariamente selecionadas do *benchmark* TSP da TSPLIB. Os estudos experimentais com a Metodologia da Superfície de Respostas produziram 5 resultados diferentes para cada parâmetro, sendo cada resultado relacionado a um exemplar. A partir desses resultados, foram definidos os espaços de busca de parâmetros, limitados pelos valores máximo e mínimo de cada parâmetro no conjunto de treinamento, tal que:

- $p_c \in [0,412; 0,641]$ ;
- $p_m \in [0,008; 0,035]$ ;
- $\mu \in [208; 390]$ ; e
- $g \in [2117; 2500]$ .

A exploração do espaço de busca foi conduzida por meio da heurística HORA, que cria alternativas sob demanda na vizinhança de uma configuração candidata conhecida. Para cada uma das  $m$  alternativas criadas (neste estudo,  $m = 4$ ), a meta-heurística GA foi executada durante 15 segundos sobre um conjunto expandido de exemplares.

Para comparações, considera-se o espaço de busca previamente definido e uma abordagem de sintonização inspirada no método de Corrida F-Race [Birattari et al., 2002 e 2009; Balaprakash et al., 2007], aqui chamado *Racing*. As configurações usadas neste processo foram as seguintes:  $p_c \in \{0.412; 0.526; 0.641\}$ ,  $p_m \in \{0.008; 0.015; 0.022; 0.028; 0.035\}$ ,  $\mu \in \{208;$





299; 390} e  $g \in \{2117; 2245; 2372; 2500\}$ . Essas configurações foram definidas com o número de níveis que parece ser suficiente para levar o GA a um bom resultado. Como cada possível combinação de parâmetros corresponde a uma configuração diferente da meta-heurística, o espaço de pesquisa é composto por  $3 \times 5 \times 3 \times 4 = 180$  configurações candidatas. A ideia é usar o *Racing* para selecionar a melhor configuração possível a partir de um número grande de opções. Para cada uma das alternativas criadas, a meta-heurística GA foi executada durante 15 segundos sobre o mesmo conjunto de exemplares usados anteriormente pelo HORA.

O processo de sintonização do GA com ambos métodos (HORA e *Racing*) foi repetido 15 vezes e as configurações propostas são apresentadas em termos de média e desvio padrão ( $\mu \pm \sigma$ ) na Tabela 2.

**Tabela 2.** Configurações de parâmetros propostas pelos métodos HORA e *Racing*.

(a) HORA		(b) <i>Racing</i>	
Parâmetro	Configuração	Parâmetro	Configuração
$p_c$	0,438 $\pm$ 0,040	$p_c$	0,420 $\pm$ 0,030
$p_m$	0,015 $\pm$ 0,006	$p_m$	0,022 $\pm$ 0,009
$\mu$	218 $\pm$ 14	$\mu$	208 $\pm$ 0
$g$	2236 $\pm$ 89	$g$	2287 $\pm$ 142

Os resultados deste estudo de caso (Tabela 2) revelam semelhanças entre HORA e *Racing* em termos de parametrização. Entretanto, algumas diferenças podem ser identificadas durante o processo de sintonização. Isto é, este processo é mais ligeiro com o HORA (2378 segundos) do que com o *Racing* (9959 segundos). O número de experimentos realizados pelo HORA é menor (146 experimentos) em relação ao *Racing* (666 experimentos). Apesar das diferenças destacadas, o HORA utiliza mais exemplares do que o *Racing*, isto é, 7,5 e 7,1, respectivamente. O número médio de configurações candidatas sobreviventes ao final do processo de sintonização é 1,7 com o HORA, e 1,4 com o *Racing*. Embora existam diferenças de desempenho durante o processo de sintonização, deve-se ressaltar que ambas abordagens utilizam o mesmo método de avaliação das configurações candidatas.

#### 4.1 Análise dos Resultados

Embora não seja o objetivo principal deste trabalho, é interessante verificar se o GA sintonizado pelas diferentes abordagens (HORA e *Racing*) alcança bons resultados para o TSP. Assim, cada uma das configurações propostas por ambas abordagens (Tabela 2) foram executadas 10 vezes sobre 48 exemplares do *benchmark* TSP, considerando-se como critério de parada o tempo máximo de execução do algoritmo (300 segundos). Os resultados foram coletados em um computador Intel® Core i5™ 1.8GHz, 6GB de memória, 1TB de disco rígido e sistema operacional Windows 8 64bit.

Para comparar os resultados utiliza-se:

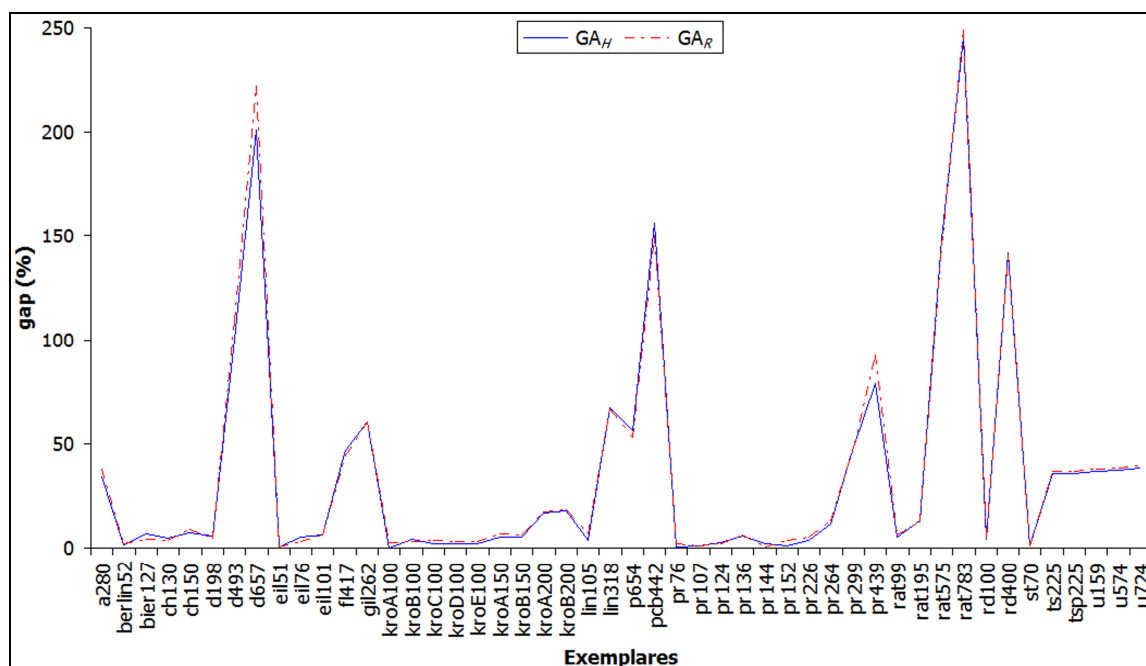
$$gap = \frac{f(s) - f(s^*)}{f(s^*)} \times 100, \quad (2)$$

em que  $f(s)$  é uma solução computada pela meta-heurística GA e  $f(s^*)$  é a melhor solução conhecida para o problema. Portanto, quanto menor o valor de  $gap$ , melhor é o desempenho do algoritmo.

Um resumo dos resultados é apresentado em forma gráfica na Figura 6. No gráfico,  $GA_H$  e  $GA_R$  correspondem ao algoritmo sintonizado pelos métodos HORA e *Racing*, respectivamente. A partir da inspeção visual (Figura 6) observa-se semelhanças nos resultados. Entretanto, análises estatísticas por meio de testes de hipóteses *t-Student* e Wilcoxon (Tabela 3)



permitem diferencia-los e concluir que há diferenças estatisticamente significantes ao nível de significância de 10% ( $\alpha = 0,10$ ), tal que os resultados produzidos pelo  $GA_H$  são melhores.



**Figura 6.** Resultados computacionais do GA para o TSP obtidos a partir de diferentes configurações.

**Tabela 3.** Testes estatísticos.

<i>t-Student</i>		Wilcoxon	
<i>t</i>	valor- <i>p</i>	<i>Z</i>	valor- <i>p</i>
1,859	0,069	-2,074	0,038

## 5. Considerações Finais

Este artigo apresentou uma abordagem para o problema da sintonização dos parâmetros de meta-heurísticas. O problema foi formalizado como um espaço de estados, cuja exploração é conduzida efetivamente por um método heurístico, que combina DOE e método de Corrida.

O método proposto, chamado HORA, aplica estatísticas robustas em um número limitado de exemplares de uma classe de problemas, a fim de delimitar o espaço de busca de parâmetros. Assim, a partir de alternativas criadas dinamicamente na vizinhança de uma configuração candidata conhecida, emprega-se um método de Corrida para encontrar consistentemente as boas configurações.

A partir de um estudo de caso, o HORA foi aplicado na sintonização da meta-heurística GA para solução de diferentes exemplares do TSP e seus resultados foram comparados com a mesma meta-heurística sintonizada pelo método de Corrida. O método HORA provou ser eficaz no processo de sintonização. O melhor desempenho observado pode ser explicado pelo modo como as configurações candidatas são exploradas no espaço busca, isto é, a partir da criação alternativas na vizinhança de uma configuração candidata conhecida, bem como, pela utilização do método de corrida em sua avaliação.

Os resultados apresentados sugerem que o HORA pode ser uma ferramenta promissora e poderosa para auxiliar a sintonização de diferentes algoritmos. Alguns estudos adicionais podem ser conduzidos para verificar sua efetividade considerando outras meta-heurísticas e problemas.



## Referências

- Adeson-Diaz, B., Laguna, M. (2006) Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, Baltimore, v. 54, n. 1, p. 99-114.
- Amoozegar, M.; Rashedi, E. (2014) Parameter tuning of GSA using DOE. In: *4th International Conference on Computer and Knowledge Engineering (ICCKE)*, 4., 2014, p. 431-436.
- Akbaripour, H.; Masehian, E. (2013) Efficient and Robust Parameter Tuning for Heuristic Algorithms. *International Journal of Industrial Engineering & Production Research*, Tehran, v. 24, n. 2, p. 143-150.
- Applegate, D. L.; Bixby, R. E.; Chvátal, V.; Cook, W. J. (2007) *The traveling salesman problem: a computational study*. 2nd ed., Princeton Series in Applied Mathematics, Princeton: Princeton University Press.
- Balaprakash, P., Birattari, M., Stützle, T., Dorigo, M. (2007) Improvement strategies for the F-Race algorithm: sampling design and iterative refinement. In: *4th International Workshop On Hybrid Metaheuristics*, 4., 2007, p. 108-122.
- Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K. (2002) A racing algorithm for configuring metaheuristics. In: *Genetic And Evolutionary Computation Conference*, 2002, New York. p. 11-18.
- Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T. (2009) F-Race and iterated F-Race: an overview. Bruxelles: Iridia Technical Report Series. 21 p.
- Blum, C., Roli, A. (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, v. 35, n. 3, p. 268-308.
- Calvet, L.; Juan, A. A.; Serrat, C.; Ries, J. (2016) A statistical learning based approach for parameter fine-tuning of metaheuristics. *SORT Statistics and Operations Research Transactions*, v. 40, n. 1, p. 201-224
- Coley, D. A. (1999) *Introduction to genetic algorithms for scientists and engineers*. London: Wspc, 244 p.
- Matai, R.; Singh, S. P.; Mittal, M. L. (2010) Traveling salesman problem: an overview of applications, formulations, and solution approaches. In: Davendra, D. *Traveling salesman problem: theory and applications*, Chapter 1, p. 1-24.
- Derigs, U. (2009) *Optimization and operations research*. v. 2, Paris: EOLSS Publishers Co. Ltd. 273 p.
- Dobslaw, F. (2010) A parameter tuning framework for metaheuristics based on design of experiments and artificial neural networks. In: *Sixth International Conference On Natural Computation*, 6., 2010, Cairo, p. 1-4.
- Goldberg, D. E. (1989) *Genetic Algorithms in search, optimization, and machine learning*. Boston: Addison-Wesley, 432 p.
- Holland, J. H. (1975) *Adaptation in natural and artificial systems*. Boston: University of Michigan Press, 211 p.



- Hutter, F., Hoos, H., Leyton-Brown, K., Stützle, T. (2009) ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, v. 36, p. 267-306
- Laporte, G. (2007) What you should know about the vehicle routing problem. *Naval Research Logistics*, New York, v. 54, p. 811-819.
- Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S. (1999) Genetic Algorithms for the travelling salesman problem: a review of representations and operators. *Artificial Intelligence Review*, v. 13, p. 129-170.
- Lessmann, S.; Caserta, M.; Arango, I. M. (2011) Tuning metaheuristics: A data mining based approach for particle swarm optimization. *Expert Systems with Applications*, v. 38, n. 10, New York: Pergamon Press, p. 12826-12838.
- Maron, O., Moore, A. W. (1994) Hoeffding races: accelerating model selection search for classification and function approximation. *Advances in Neural Information Processing Systems*, San Mateo, p. 59-66.
- Mitchell, M. (1998) An introduction to genetic algorithms. Massachusetts: MIT Press. 221 p.
- Montgomery, D. C. (2012) Design and analysis of experiments. 8th ed. New Jersey: John Wiley & Sons Inc.. 699 p.
- Neumüller, C.; Wagner, S.; Kronberger, G.; Affenzeller, M. (2011) Parameter Meta-optimization of Metaheuristic Optimization Algorithms. In: *13th International Conference on Computer Aided Systems Theory (EUROCAST 2011)*, 13., 2011, Las Palmas de Gran Canaria, p. 367-374.
- Reinelt, G. (1991) TSPLIB: A traveling salesman problem library. *ORSA Journal on Computing*, Baltimore, v. 3, n. 4, p. 376-384.
- Ries, J.; Beullens, P.; Salt, D. (2012) Instance-specific multi-objective parameter tuning based on fuzzy logic. *European Journal of Operational Research*, v. 218, p. 305-315.
- Smit, S. K.; Eiben, A. E. (2009) Comparing parameter tuning methods for evolutionary algorithms. In: *2009 Ieee Congress On Evolutionary Computation*, 2009, Trondheim. 2009 IEEE Congress on Evolutionary Computation (CEC'09), 2009. New York: IEEE. p. 399-406
- Talbi, E.G. (2003) Metaheuristics: from design to implementation. New Jersey: John Wiley & Sons Inc., 593 p.