

Configuração Automática de Algoritmos: Métodos e Aplicações

Marcelo de Souza

Universidade do Estado de Santa Catarina
Rua Dr. Getúlio Vargas, 2822, Ibirama, SC, 89140-000
marcelo.desouza@udesc.br

Marcus Ritt

Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500, Porto Alegre, RS, 91501-970
marcus.ritt@inf.ufrgs.br

RESUMO

Métodos automáticos de configuração de algoritmos reduzem o esforço humano e potenciais vieses de abordagens de configuração manuais. A tese apresenta um estudo abrangente sobre configuração automática de algoritmos, cujas contribuições principais são: (i) um conjunto de métodos de poda que reduzem o esforço na configuração de algoritmos de otimização; (ii) a exploração de modelos de regressão de parâmetros para produção de melhores configurações; (iii) uma ferramenta visual para facilitar a análise e o entendimento do processo de configuração; e (iv) um *solver* heurístico que explora técnicas de projeto automático para a produção de algoritmos para a classe geral de problemas binários de otimização. Os resultados experimentais detalhados na tese ressaltam a importância dessas contribuições para essa área de pesquisa.

PALAVRAS CHAVE. Ajuste de parâmetros. Métodos de poda. Modelos de regressão.

Tópicos: inteligência computacional; outras aplicações em PO.

ABSTRACT

Automatic methods for algorithm configuration reduce the human effort and potential biases involved in manual configuration approaches. The thesis presents a comprehensive study on automatic algorithm configuration, whose main contributions are: (i) a set of capping methods that reduce the effort in configuring optimization algorithms; (ii) the exploration of parameter regression models to produce better-performing configurations; (iii) a visual tool to ease the analysis and understanding of the configuration process; and (iv) a heuristic solver that explores automatic design techniques to produce algorithms for a general class of binary optimization problems. The experimental results detailed in the thesis highlight the importance of such contributions to this research field.

KEYWORDS. Parameter tuning. Capping methods. Regression models.

Paper topics: computational intelligence; other applications in OR.

Este documento apresenta um resumo da tese de doutorado de Marcelo de Souza, intitulada “*Automatic Algorithm Configuration: Methods and Applications*” e orientada pelo professor Dr. Marcus Rolf Peter Ritt. A tese foi desenvolvida no Programa de Pós-Graduação em Computação (PPGC) da Universidade Federal do Rio Grande do Sul (UFRGS) e defendida em fevereiro de 2022. A pesquisa aqui relatada trouxe importantes contribuições ao projeto e configuração automáticos de algoritmos,

sendo aprovada com máxima distinção acadêmica pela banca composta por professores de diferentes instituições do mundo e com vasta experiência no tema do trabalho. Por essas razões, candidata-se a referida tese ao Prêmio de Tese do 54º Simpósio Brasileiro de Pesquisa Operacional.

1. Resumo da Tese e suas Contribuições

Muitos algoritmos possuem parâmetros de entrada, que permitem aos usuários adaptar seu comportamento a diferentes domínios de problema. O conjunto de valores para os parâmetros de um algoritmo, chamado de *configuração*, está diretamente relacionado ao seu desempenho. Neste contexto, dado um conjunto de instâncias de treinamento, a *configuração de algoritmos* busca pelas melhores configurações, i.e. aquelas que otimizam o desempenho do algoritmo nessas instâncias. Esse processo envolve testar várias configurações em diferentes instâncias, tornando-o custoso e exigindo um esforço considerável do pesquisador.

Diversas abordagens foram propostas nos últimos anos com o objetivo de automatizar o processo de configuração de algoritmos. Dentre elas se destacam configuradores baseados em busca heurística, otimização baseada em modelos e algoritmos de *racing*. Além de reduzir o esforço humano envolvido no processo de configuração, esses métodos minimizam potenciais vieses de abordagens manuais, facilitam a adaptação de algoritmos a diferentes problemas, e permitem avaliar e comparar algoritmos de forma mais adequada. Além disso, esses métodos se mostraram úteis na melhoria do desempenho de algoritmos em diferentes domínios, incluindo *solvers* matemáticos, compiladores, algoritmos de decisão e algoritmos de otimização.

A tese apresenta um estudo abrangente sobre configuração automática de algoritmos, cujas contribuições se baseiam em quatro objetivos: (i) aumentar a eficiência da configuração automática de algoritmos de otimização; (ii) melhorar a qualidade das configurações produzidas; (iii) facilitar a análise e o entendimento do processo de configuração; e (iv) aplicar os métodos de configuração automática na construção de *solvers* heurísticos para classes de problemas. As seções a seguir resumem os métodos e ferramentas propostos na tese e os principais resultados alcançados.

1.1. Métodos de Poda para Cenários de Otimização

Ao configurar algoritmos visando a minimização do seu tempo de execução, diversos configuradores da literatura implementam métodos de poda, os quais visam evitar o gasto de tempo na avaliação de configurações não promissoras [Hutter et al., 2009]. Esses métodos determinam um limite de tempo de execução com base em execuções anteriores da melhor configuração encontrada até o momento, e então terminam antecipadamente execuções que excedam esse limite. Essas abordagens não são adequadas para a configuração de algoritmos de otimização, uma vez que nesse caso o objetivo consiste em minimizar o custo da melhor solução encontrada após executar o algoritmo com um critério de terminação predeterminado. Os configuradores atuais, portanto, não implementam nenhum método de poda para cenários de otimização, o que torna o processo de configuração desses algoritmos mais custoso e em alguns casos inviável.

A tese preenche essa lacuna, propondo um conjunto de métodos de poda capaz de reduzir o tempo de configuração de algoritmos de otimização. A Figura 1 apresenta a ideia geral desses métodos. Cada execução é representada pelo seu perfil de desempenho, i.e. o custo da melhor solução encontrada ao longo do tempo de execução (ou ao longo de qualquer outra medida de esforço computacional, como o número de iterações). Antes de iniciar uma nova execução, os perfis de desempenhos observados em execuções anteriores são agregados em um envelope de desempenho, o qual é usado para avaliar a qualidade da nova execução. Caso ela não satisfaça o desempenho mínimo definido pelo envelope, a execução é interrompida antecipadamente. São propostas duas abordagens gerais para a construção dos envelopes de desempenho. As abordagens baseadas em

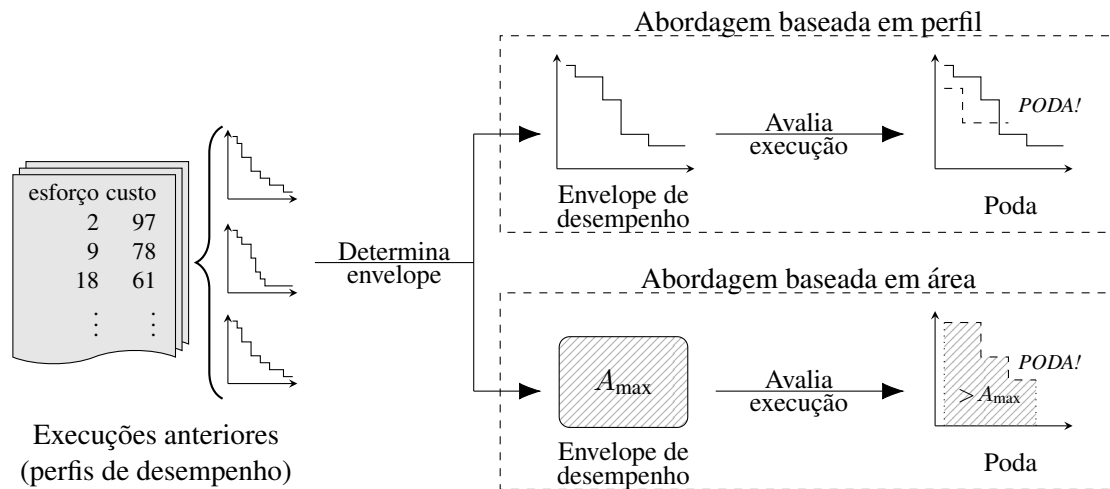


Figura 1: Visão geral dos métodos de poda.

perfil (ilustradas na parte superior direita da Figura 1) definem esse envelope como um perfil de desempenho agregado. Já as abordagens baseadas em área (ilustradas na parte inferior direita da Figura 1) consideram a área usada pelos perfis de desempenho das execuções anteriores, agregando-as em um valor máximo de área. Se em algum momento a nova execução violar as condições definidas pelo envelope, i.e. apresentar soluções piores que o esperado ou uma área maior que o limite por ele definido, a execução é interrompida.

Foram propostas diferentes funções de agregação, tanto para perfis de desempenho quanto para os valores correspondentes de área, dando origem a 18 métodos de poda distintos. Esses métodos foram avaliados em seis cenários de otimização, comparando-os com a configuração sem uso de poda. Ao usar métodos de poda, observou-se uma redução do esforço de configuração entre 5% e 78%, ao mesmo tempo que mantida a qualidade das configurações produzidas. Também foi avaliada a contribuição dos métodos de poda em cenários onde o esforço de configuração é definido em termos do tempo máximo de execução, i.e. um tempo máximo para o processo de configuração. Nesses casos, a economia de tempo obtida por podar execuções não promissoras permite uma exploração mais abrangente do espaço de configurações. Como consequência, o uso de métodos de poda permitiu encontrar configurações de maior qualidade em comparação àquelas obtidas nas mesmas condições sem o uso de poda.

1.2. Modelos Melhorados para Regressão de Parâmetros

Uma característica dos métodos de configuração estudados neste trabalho é que eles determinam configurações *fixas* para todo o conjunto de instâncias de treinamento. Ou seja, os parâmetros assumem valores constantes na solução de diferentes instâncias, não importa as variadas características que elas apresentem. É conhecido, no entanto, que bons valores de parâmetros podem variar em função de características das instâncias (e.g. El Yafrani e Ahiod [2018]), incluindo características específicas do problema, como estatísticas sobre cláusulas e variáveis em problemas de satisfatibilidade booleana, ou independentes de problema, como o tamanho da instância ou propriedades da paisagem de solução. Sob uma perspectiva de aprendizagem de máquina, abordagens tradicionais de configuração de algoritmos correspondem a uma regressão constante no espaço de parâmetros.

A tese explora modelos adicionais de regressão de parâmetros para a configuração de algoritmos. Os parâmetros são representados por modelos que definem seus valores em função do

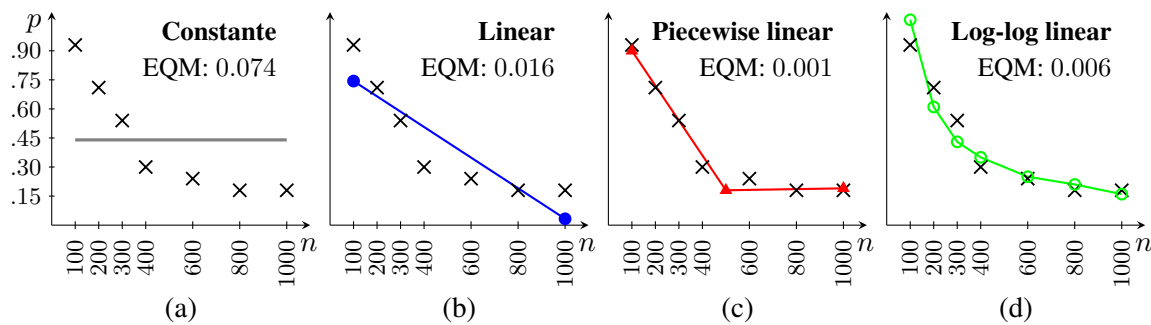


Figura 2: Comparação dos modelos de regressão em um cenário artificial não-linear.

tamanho da instância. Essa característica de instância é independente de problema e obtida sem a necessidade de processamento computacional custoso, o que torna o método simples e pronto para ser usado na configuração de algoritmos para qualquer domínio de problema. Inicialmente, o trabalho explora modelos lineares que, apesar de simples, são capazes de modelar a relação entre tamanho de instância e valores ótimos de parâmetros para vários cenários de configuração. Nos casos em que essa relação é não linear, este trabalho propõe um modelo linear em trechos (ou modelo *piecewise linear*) baseado em múltiplos pontos de apoio. Além disso, o trabalho propõe um modelo *log-log linear* que aplica uma transformação logarítmica aos espaços de parâmetros e tamanhos de instância, e então aplica o modelo linear básico ao cenário transformado. Ambos modelos *piecewise* e *log-log linear* são capazes de representar relações não lineares entre o tamanho da instância e valores ótimos de parâmetros.

A Figura 2 apresenta um cenário artificial com a resposta do valor ótimo de um parâmetro p em função de diferentes tamanhos de instância n . Para cada modelo, é apresentada uma aproximação que minimiza o erro quadrático médio (EQM). É possível verificar que o modelo constante (a) não representa de maneira adequada a relação entre tamanho da instância e valor ótimo do parâmetro, obtendo um EQM de 0.074. A representação é melhorada ao usar o modelo linear (b, com um EQM de 0.016) porém, dada a não linearidade dessa relação, os modelos *piecewise* (c) e *log-log linear* (d) apresentam uma melhor aproximação (EQMs de 0.001 e 0.006, respectivamente).

Esses modelos foram avaliados em quatro cenários reais. Para cada cenário, foi verificada a relação entre o tamanho da instância e os melhores valores dos parâmetros. Em seguida, os algoritmos foram configurados usando a abordagem tradicional (constante) e os modelos de regressão propostos. Finalmente, a qualidade das configurações produzidas foi avaliada em um conjunto independente de instâncias de teste. Em geral, o modelo linear foi capaz de aproximar de maneira satisfatória a relação entre tamanho da instância e valor ótimo dos parâmetros. Em um dos cenários, onde essa relação é não linear, os modelos *piecewise* e *log-log linear* foram capazes de melhorar a aproximação. Como consequência, os algoritmos sob configurações não constantes mostraram desempenho superior, em comparação ao uso de configurações constantes.

1.3. Análise Visual do Processo de Configuração

Diante da facilidade de uso de métodos de configuração automática de algoritmos, é possível e até mesmo comum sua aplicação como métodos de caixa-preta, i.e. sem a necessidade de uma análise e entendimento aprofundados do seu funcionamento. Por outro lado, é importante analisar as execuções do configurador usado e entender seu funcionamento, de modo a obter os melhores resultados do processo de configuração. Os dados gerados durante a execução do configurador fornecem informações úteis sobre o desempenho do algoritmo sob diferentes configurações e sobre as decisões tomadas durante o processo de configuração. No entanto, analisar esses dados (usualmente

encontrados em grandes quantidades) é uma tarefa desafiadora, além de exigir conhecimento sobre o funcionamento interno do configurador, e.g. como ele seleciona configurações para serem avaliadas. Diante disso, a tese propõe uma ferramenta que processa, interpreta e fornece visualizações do processo de configuração de algoritmos, com o objetivo de simplificar sua análise e entendimento.

A ferramenta desenvolvida, chamada *acviz*, fornece duas visualizações principais com base em execuções do configurador *irace* [López-Ibáñez et al., 2016]. A primeira delas apresenta o processo de configuração e sua evolução, mostrando cada avaliação do algoritmo, junto da configuração e instância associadas e o resultado correspondente. Como o *irace* implementa um método de configuração iterativo, o *acviz* mostra ainda o desempenho mediano das configurações de cada iteração, bem como das melhores configurações encontradas até o momento, o que permite identificar a evolução na qualidade das configurações produzidas ao longo do tempo. A segunda visualização mostra o desempenho das melhores configurações encontradas, tanto do processo completo como de cada iteração, em um conjunto de instâncias de teste. O principal benefício dessa visualização é verificar o poder de generalização dessas configurações para novas instâncias e o desempenho esperado do algoritmo em produção.

O uso e benefícios da ferramenta *acviz* foram avaliados em três estudos de caso envolvendo a configuração de algoritmos de decisão e otimização. O primeiro estudo de caso considera cenários onde o conjunto de instâncias de treinamento inclui uma instância muito fácil de ser resolvida e outra muito difícil. As visualizações produzidas permitem identificar essas características das duas instâncias e perceber que ambas não contribuem para a avaliação e comparação de configurações, visto que o desempenho de diferentes configurações são igualmente bons na instância fácil e igualmente ruins na instância difícil. O segundo estudo de caso considera cenários onde o esforço computacional empregado no processo de configuração (número máximo de avaliações ou tempo limite) é demasiado alto. As visualizações mostram a convergência do processo de configuração muito antes do seu término. Novas configurações produzidas são muito similares àquelas já avaliadas, com a consequente estagnação de desempenho. Finalmente, o terceiro estudo de caso considera cenários com *overtuning*, i.e. quando as configurações produzidas se especializam nas instâncias de treinamento e, dada sua baixa representatividade, o desempenho observado em novas instâncias é fraco. As visualizações dos resultados das melhores configurações de cada iteração nos conjuntos de instâncias de treinamento e teste permitem identificar tal efeito, pois mostram a melhoria do desempenho nas instâncias de treinamento ao longo do processo de configuração, ao passo que o desempenho nas instâncias de teste piora simultaneamente. Em resumo, a ferramenta *acviz* permite identificar problemas nos cenários de configuração que não são facilmente detectados de outra forma, como demonstrado nos estudos de caso propostos. Com isso, o projetista pode corrigir tais falhas, como selecionar instâncias adequadas ou reduzir o esforço de configuração, e melhorar a qualidade das configurações produzidas.

1.4. Solver Baseado em Componentes para Otimização Binária

Um uso mais avançado de métodos de configuração consiste em aplicá-los para automatizar o processo completo do projeto de algoritmos. Em vez de focar na configuração dos parâmetros de um algoritmo predeterminado, define-se um conjunto de componentes algorítmicos que podem ser selecionados, combinados e calibrados para a produção de distintos algoritmos. Um configurador é então aplicado para automatizar a exploração desse espaço de componentes e buscar pelos melhores algoritmos para um dado domínio de problema. Essa abordagem é chamada de *projeto automático de algoritmos* e tem o potencial de produzir algoritmos híbridos e competitivos com o estado-da-arte. Stützle e López-Ibáñez [2019] discutem diferentes abordagens para o o projeto automático de algoritmos e apresentam aplicações em variados domínios. Apesar da vasta aplicabilidade e dos

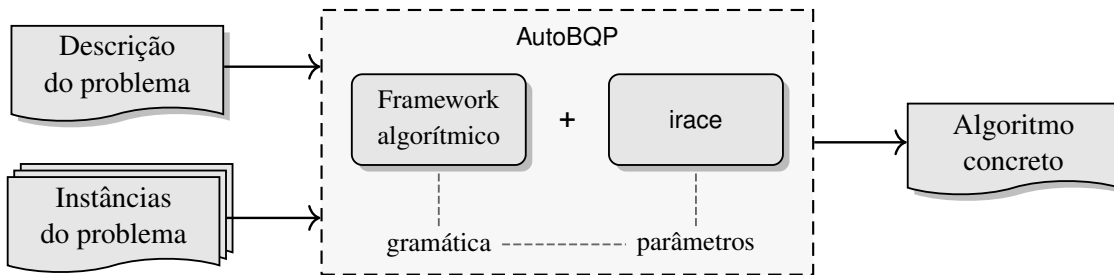


Figura 3: Visão geral do projeto automático de algoritmos com AutoBQP.

resultados positivos reportados na literatura, os algoritmos produzidos usando as técnicas supracitadas são geralmente especializados em um problema específico. Em contrapartida, a tese apresenta um *solver* chamado AutoBQP, que aplica os conceitos de projeto automático de algoritmos para resolver problemas de otimização binários, i.e. a classe geral de problemas de otimização cujas soluções são representadas por um vetor de *bits*.

A Figura 3 apresenta a estrutura e funcionamento do *solver* AutoBQP. Dada uma descrição do problema contendo sua função objetivo e um conjunto de instâncias de treinamento, o *solver* busca automaticamente por algoritmos de alto desempenho. Para isso, o AutoBQP combina um *framework*, que implementa os componentes algorítmicos e permite sua seleção e combinação, com o configurador *irace*, responsável por buscar as melhores combinações de componentes e valores para seus parâmetros de entrada. As regras para combinação de componentes e produção de algoritmos completos são representadas por uma gramática, cujas decisões são mapeadas para parâmetros [Mascia et al., 2014], permitindo a comunicação do *framework* com o *irace*. O *framework* algorítmico foi construído com base em componentes heurísticos para programação quadrática binária irrestrita (ou UBQP, do inglês *unconstrained binary quadratic programming*). Como muitos problemas podem ser reduzidos e resolvidos via UBQP [Kochenberger et al., 2014], o AutoBQP é capaz de resolver toda essa classe de problemas binários.

O *solver* AutoBQP foi usado para produzir algoritmos heurísticos para três problemas distintos: o próprio UBQP; o problema de corte máximo em grafos (MaxCut), que pode ser representado diretamente como um modelo de programação quadrática binária irrestrita; e a alocação de testes, uma variante da coloração de grafos que pode ser reduzida ao UBQP via reformulação. Os algoritmos produzidos se mostraram competitivos, apresentando desempenho superior em comparação com abordagens do estado-da-arte. Além disso, os algoritmos produzidos encontraram novas e melhores soluções para diversas instâncias do MaxCut e da alocação de testes. Esses resultados mostram o potencial do projeto automático de algoritmos, que permite construir soluções de alta qualidade com pouco esforço humano e sem a necessidade de conhecimento especializado nas técnicas de configuração automática e nos problemas explorados.

1.5. Discussão

Os métodos propostos neste trabalho apresentam contribuições relevantes para o projeto e configuração automáticos de algoritmos. Eles permitem reduzir o tempo de configuração e melhorar a qualidade dos resultados, enquanto as visualizações produzidas facilitam a análise e entendimento do processo de configuração. Além disso, o *solver* desenvolvido permite projetar algoritmos no estado-da-arte de maneira automática para uma ampla classe de problemas de otimização.

A tese ainda apresenta uma série de contribuições indiretas. As aplicações apresentadas ao longo do trabalho demonstram a relevância prática e a eficácia do projeto e configuração automáticos de algoritmos, bem como apresentam um conjunto de diretrizes a pesquisadores e projetistas quanto

ao uso desses métodos. Com o *solver* desenvolvido, foram produzidos algoritmos para diferentes problemas e com melhor desempenho em comparação com abordagens da literatura. Finalmente, o trabalho fornece um conjunto amplo de cenários de configuração, cujos algoritmos podem ser adaptados para diferentes domínios de problema. Esses cenários também são úteis no desenvolvimento, avaliação e comparação de técnicas de configuração de algoritmos.

As implementações de todos os métodos e ferramentas propostos na tese estão disponíveis para uso, bem como as implementações dos algoritmos produzidos. Também estão disponíveis materiais suplementares para cada etapa da pesquisa (i.e. capítulo da tese), contendo os algoritmos correspondentes, definição dos cenários de configuração, instâncias de treinamento e teste e detalhes adicionais dos resultados experimentais. Todos os cenários de configuração usados nos experimentos foram também disponibilizados separadamente para facilitar o acesso. Esse material, bem como o texto completo da tese, pode ser acessado em <https://souzamarcelo.github.io/phd>.

2. Publicações

A partir da pesquisa aqui descrita, foram publicados diversos artigos científicos, os quais são listados abaixo. Os resultados dos métodos de poda para cenários de otimização, dos modelos de regressão de parâmetros e da ferramenta de visualização foram publicados nos artigos descritos, respectivamente, nos itens (1 – 3). O *solver* baseado em componentes para otimização binária originou as publicações descritas em (4 – 6).

1. Marcelo de Souza, Marcus Ritt e Manuel López-Ibáñez. **Capping methods for the automatic configuration of optimization algorithms**. Computers & Operations Research, v. 139, p. 1–15, 2022. [DOI: [10.1016/j.cor.2021.105615](https://doi.org/10.1016/j.cor.2021.105615)]
2. Marcelo de Souza e Marcus Ritt. **Improved regression models for algorithm configuration**. The Genetic and Evolutionary Computation Conference (GECCO), p. 222–231, 2022. [DOI: [10.1145/3512290.3528750](https://doi.org/10.1145/3512290.3528750)]
3. Marcelo de Souza, Marcus Ritt, Manuel López-Ibáñez e Leslie Pérez Cáceres. **ACVIZ: A tool for the visual analysis of the configuration of algorithms with irace**. Operations Research Perspectives, v. 8, p. 1–9, 2021. [DOI: [10.1016/j.orp.2021.100186](https://doi.org/10.1016/j.orp.2021.100186)]
4. Marcelo de Souza e Marcus Ritt. **A study of the automatic design of heuristics for binary quadratic programming**. XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), p. 1673–1684, 2016.
5. Marcelo de Souza e Marcus Ritt. **Automatic grammar-based design of heuristic algorithms for unconstrained binary quadratic programming**. 18th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP), p. 67–84, 2018. [DOI: [10.1007/978-3-319-77449-7_5](https://doi.org/10.1007/978-3-319-77449-7_5)]
6. Marcelo de Souza e Marcus Ritt. **An automatically designed recombination heuristic for the test-assignment problem**. 2018 IEEE Congress on Evolutionary Computation (CEC), p. 2411–2418, 2018. [DOI: [10.1109/CEC.2018.8477801](https://doi.org/10.1109/CEC.2018.8477801)]

Além disso, a pesquisa foi selecionada para apresentação em diferentes eventos científicos, os quais estão detalhados abaixo.

1. Marcelo de Souza and Marcus Ritt. **Capping strategies based on performance envelopes for the automatic design of meta-heuristics**. XXIII Latin-American Summer School in Operational Research (ELAVIO), 2019.

2. Marcelo de Souza and Marcus Ritt. **Capping strategies based on performance envelopes for the automatic design of meta-heuristics**. Configuration and Selection of Algorithms Workshop (COSEAL), 2019.
3. Marcelo de Souza. **On the automatic configuration of algorithms**. 1st Doctoral Consortium at the European Conference on Artificial Intelligence (DC-ECAI), p. 15–16, 2020.
4. Marcelo de Souza. **How to speed-up the automated configuration of optimization algorithms**. Doctoral Consortium of the 14th Annual Symposium on Combinatorial Search (SoCS), AAAI, 2021.
5. Marcelo de Souza. **Automatic design of heuristic algorithms for binary optimization problems**. Doctoral Consortium of the 30th International Joint Conference on Artificial Intelligence (IJCAI), 2021. [DOI: [10.24963/ijcai.2021/672](https://doi.org/10.24963/ijcai.2021/672)]
6. Marcelo de Souza, Marcus Ritt e Manuel López-Ibáñez. **Efficient Configuration of Optimization Algorithms**. Hot off the Press, The Genetic and Evolutionary Computation Conference (GECCO), 2022.

A pesquisa aqui descrita foi reconhecida em diferentes oportunidades. Ao ser selecionada para participação na Escola Latinoamericana de Verão em Pesquisa Operacional (ELAVIO 2019), foi premiada com uma bolsa cobrindo os custos de viagem, acomodação e alimentação. O mesmo aconteceu na edição de 2022 do mesmo evento, ocasião em que a pesquisa foi ranqueada em primeiro lugar entre os inscritos do Brasil. Em 2019, o autor foi aceito para desenvolver parte da pesquisa como pesquisador visitante na Universidade de Manchester (Reino Unido), em colaboração com o professor Manuel López-Ibáñez. Finalmente, a tese foi aprovada com máxima distinção acadêmica*.

Referências

- El Yafrani, M. e Ahiod, B. (2018). Efficiently solving the traveling thief problem using hill climbing and simulated annealing. *Information Sciences*, 432:231–244. ISSN 0020-0255.
- Hutter, F., Hoos, H. H., Leyton-Brown, K., e Stützle, T. (2009). ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306.
- Kochenberger, G. A., Hao, J.-K., Glover, F., Lewis, R. M. R., Lü, Z., Wang, H., e Wang, Y. (2014). The unconstrained binary quadratic programming problem: A survey. *Journal of Combinatorial Optimization*, 28(1):58–81.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T., e Birattari, M. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- Mascia, F., López-Ibáñez, M., Dubois-Lacoste, J., e Stützle, T. (2014). Grammar-based generation of stochastic local search heuristics through automatic algorithm configuration tools. *Computers & Operations Research*, 51:190–199.
- Stützle, T. e López-Ibáñez, M. (2019). Automated design of metaheuristic algorithms. In Gendreau, M. e Potvin, J.-Y., editors, *Handbook of Metaheuristics*, volume 272 of *International Series in Operations Research & Management Science*, p. 541–579. Springer International Publishing, New York, NY.

*A banca examinadora foi composta pelos professores Dr. Thomas Stützle (Université Libre de Bruxelles), Dra. Carola Doerr (Sorbonne Université) e Dr. Anderson Rocha Tavares (UFRGS).