

# DA COMPUTAÇÃO INSPIRADA NA NATUREZA À INTELIGÊNCIA DE ENXAMES APLICADA AO PAISAGISMO

## FROM NATURE-INSPIRED COMPUTATION TO SWARM INTELLIGENCE APPLIED TO LANDSCAPING

*André Barcellos Ferreira  
Jarryer Andrade de Martino*

### RESUMO

O objetivo deste estudo é introduzir a utilização da inteligência de enxames como estratégia para o planejamento da manutenção de áreas verdes em campi universitários, ilustrando como essa ferramenta computacional pode contribuir para as áreas do paisagismo e da arquitetura. Como metodologia, o problema foi interpretado como discreto, e proposto como uma analogia ao Problema de Múltiplos Caixeiros Viajantes. A meta-heurística da Otimização por Colônias de Formigas foi aplicada para a solução do problema, e o programa AutoCAD foi utilizado como plataforma para o desenvolvimento do *plug-in* solucionador do problema. Como resultado, espera-se uma aplicação robusta que comprove o potencial do uso da inteligência de enxames para planejamento da manutenção de áreas verdes.

Palavras-Chave: Inteligência artificial. Inteligência de enxames. Otimização por Colônias de Formigas.

### ABSTRACT

This study aims to introduce swarm intelligence as a strategy to plan the maintenance of green areas on university campuses, illustrating how this computational tool can contribute to landscaping and architecture. Regarding methodology, the problem was interpreted as discrete, and an analogy to the multiple traveling salesman problem was proposed. The ant colony optimization metaheuristic was applied to solve the problem, and the program AutoCAD was used as a platform to develop the problem-solving *plug-in*. This study expects a robust application as a result that will prove the potential of using swarm intelligence for green area maintenance planning.

Keywords: Artificial intelligence. Swarm intelligence. Ant Colony Optimization.



## I. INTRODUÇÃO

Na arquitetura e no urbanismo, a crescente complexidade espacial vem proporcionando limitações às abordagens projetuais, sejam elas representacionais ou associativas (paramétricas). Torna-se inevitável projetar organicamente por meios digitais, inspirando-se no conceito de ecologias de projeto (Rocha; Bolssoni; Bussoloti, 2019), a fim de encontrar soluções evolutivas e adaptativas. Nesse contexto, os estudos relacionados à computação inspirada na natureza ganharam importância. Como forma de responder à complexidade crescente, pesquisadores passaram a investigar o comportamento de grupos de indivíduos e sistemas que apresentavam complexidade emergente. Os sistemas com comportamento emergente são capazes de manter equilíbrio, estabilidade e regularidade, sendo a emergência uma característica que define os sistemas dinâmicos complexos. A complexidade organizada desses grupos mostra que, embora individualmente não possuam cognição, em conjunto eles conseguem determinar soluções ótimas para variados tipos de problema. Essa percepção provocou uma mudança importante para a computação natural: a complexidade emergente deixou de ser simplesmente estudada e passou a ser reproduzida computacionalmente. Alan Turing, matemático, cientista da computação e teórico britânico teve papel de destaque nas primeiras tentativas de imaginar a solução de problemas pela reprodução em computador da complexidade organizada dos animais (Johnson, 2012). À emulação de colônias de insetos para a solução de problemas de otimização deu-se o nome de inteligência de enxames.

O estudo e a simulação de diferentes espécies de insetos deu origem a algoritmos específicos, que criaram mecanismos de simulação baseados no comportamento dessas espécies. Este artigo busca fazer um registro histórico do surgimento da inteligência de enxames e da meta-heurística da Otimização por Colônias de Formigas. Busca caracterizar o planejamento da manutenção de áreas verdes em campi universitários como uma instância do Problema de Múltiplos Caixeiros Viajantes, e aplicar a Otimização por Colônias de Formigas para resolvê-lo.

O artigo se organiza da seguinte forma. A segunda seção define o que é a inteligência de enxames; define também a meta-heurística da Otimização por Colônias de Formigas e suas variações. A terceira seção descreve o programa desenvolvido nesta pesquisa, que aplica a Otimização por Co-

lônias de Formigas por Times ao problema da otimização da manutenção das áreas verdes de campi universitários. Por fim, a quarta seção traz as conclusões do artigo.

## 2. A INTELIGÊNCIA DE ENXAMES

Alan Turing é conhecido por ter sido o cientista capaz de decifrar a máquina de criptografia nazista na Segunda Guerra Mundial. Ele descreveu o processo no relatório *The Prof's Book: Turing's Treatise on the Enigma* (1940), concluído em 1942. Nesse relatório ele descreve textualmente, por cálculos e por ilustrações, o funcionamento da máquina, o processo de criptografia e posteriormente de descriptografia. Tratava-se basicamente de um estudo de padrões, um entendimento dos padrões com os quais a máquina criptografava os dados para permitir descriptografá-los.

O estudo de padrões teria relação direta com seu trabalho depois da guerra. Em 1953, Turing publicou o artigo “The Chemical Basis of Morphogenesis”. Motivado pela questão de como um simples embrião se transforma num ser complexo como o ser humano, Turing desenvolveu um modelo matemático para explicar como estruturas e padrões complexos podem emergir dessa uniformidade inicial (Ball, 2015). Essa pode ser interpretada como a primeira e mais bem-sucedida tentativa de explicar o desenvolvimento como um problema de complexidade emergente.

Na década de 1960, John Holland, após terminar seu doutorado em ciência da computação, iniciou uma linha de pesquisa que se aproximava bastante da pesquisa de Turing, levando-a pelo resto de sua vida. Holland foi além de Turing, querendo realmente criar soluções de programa que fossem capazes de aprender. Ele utilizou o conceito de pandemônio, proposto por Oliver Selfridge em 1959, que descreve como as imagens visuais são processadas pelo cérebro. Ele escreveu a teoria da evolução proposta por Darwin na forma de códigos, e deu a esses algoritmos o nome de algoritmos genéticos (Johnson, 2012).

De acordo com Holland (1984), os algoritmos genéticos utilizam operadores genéticos básicos – como reprodução, cruzamento e mutação – para criar diferentes combinações – formando cromossomos – a partir de uma piscina de genes. A adequação desses genes é avaliada individualmente e em conjunto, tendo o conjunto muito mais importância do que os indiví-

os. Ao longo de sucessivas gerações, cromossomos com adequações acima da média vão sendo formados. Holland tratou genes como blocos para a construção de textos, caracteres formando *strings*. Ao longo do tempo, melhores caracteres resultarão em melhores *strings*.

Segundo Jonhson (2012), aplicações que utilizam inteligência artificial jamais aprenderão da mesma forma que uma criança, jamais desenvolverão consciência, pois seu aprendizado se dá de outra forma. Elas aprendem pelo reconhecimento de padrões, por meio da análise de um grande volume de dados.

Algoritmos que simulam coletivos de indivíduos, que coletivamente apresentam capacidade de solução de problemas, são um tipo de inteligência artificial. Bonabeau, Dorigo e Theraulaz (1999) citam Butrimenko (1964), Tsetlin (1973), Stefanyuk (1971) e Rabin (1980, 1982) como exemplos do uso de coletivos de agentes simples na solução de problemas de otimização.

Em 1996, Herbert Simon publicou o livro *Science of the Artificial* (sua terceira edição). O último capítulo, “The Architecture of Complexity: Hierarchic Systems”, pode ser tomado como o mais importante, e aborda as propriedades de sistemas complexos (Bar-Yam, 1998). Antes disso, no capítulo 3, “The Psychology of Thinking: Embeding Artifice in Nature”, Simon analisa o comportamento das formigas, e relaciona sua complexidade à do ambiente em que estão inseridas: “Uma formiga, vista como um sistema comportamental, é bastante simples. A aparente complexidade em seu comportamento ao longo do tempo é em grande parte um reflexo da complexidade do ambiente em que ela se encontra” (Simon, 1996, p. 52, tradução nossa).

No restante da seção ele explora essa hipótese, mas com o termo **formiga** sendo substituído por **ser humano**. Ele explica essa simplificação, deixando claro que se refere ao *homo sapiens*, ou ser pensante, e não à complexidade de uma pessoa real. Apesar de ser uma simplificação, de acordo com Kennedy e Eberhart (2001), esse entendimento foi importante nos primórdios das pesquisas da inteligência artificial. Se pessoas são como formigas, então é possível desenvolver programas que simulem seu comportamento da mesma forma que se faz com as formigas.

Kennedy e Eberhart (2001) pontuam que uma formiga, individualmente, é um ser extremamente ignorante. Imaginar e simular seu comportamento

não é algo difícil: ela age seguindo uma lista simples de regras. Continuando esse raciocínio, eles citam Douglas Hofstadter (1979), segundo o qual a mente humana pode ser interpretada como uma colônia de formigas: um neurônio sozinho não possui cognição, mas o conjunto de neurônios sim.

Há um grau de comunicação entre as formigas, apenas o suficiente para impedir que cada uma delas tenha uma trajetória completamente aleatória. Através dessa comunicação elementar elas lembram umas às outras de que não estão sozinhas, mas agindo de modo cooperativo com as demais. É necessário um grande número de formigas, agindo como descrito, para que seja possível manter qualquer atividade, como a construção de trilhas, por qualquer período de tempo. Agora meu entendimento extremamente vago do funcionamento do cérebro me leva a pensar que algo similar ocorre na ativação dos neurônios (Hofstadter, 1979, p. 322, tradução nossa).

Na mesma linha, Kennedy e Eberhart (2001) citam Chistopher Langton. No artigo “Studying Artificial Life with Celular Automata”, de 1986, ele se propõe a estudar a possibilidade de implementar artificialmente a lógica molecular dos seres vivos utilizando autômatos celulares. Ele mostra que esses autômatos celulares, mesmo sendo moléculas inanimadas artificiais, são capazes de desempenhar os mesmos papéis verificados nas moléculas reais. Em seguida, ele propõe um exemplo em que uma colônia de formigas é simulada como uma variação de uma matriz de autômatos celulares. Considerando uma formiga como um ponto numa grade, e considerando que a complexidade do comportamento de cada formiga é um reflexo da complexidade do ambiente, o comportamento dela na colônia será baseado nos reflexos dos reflexos. Surge o que Wilson (1971 apud Langton, 1986) definiu como comunicação em massa: a capacidade de grupos de formigas transmitirem entre si informações que elas não conseguem transmitir individualmente umas para as outras.

Kennedy e Eberhart (2001) pontuam que nos últimos cinquenta anos as formigas deixaram de ser vistas como seres simples e incapazes, presas num ciclo de ações repetitivas, e passaram a ser vistas como poderosos processadores de informações, como neurônios num cérebro.

A **inteligência de enxames** faz parte de uma área mais ampla chamada de computação inspirada na natureza, que faz parte da área chamada

de computação natural. De acordo com Bonabeau, Dorigo e Theraulaz (1999), a expressão inteligência de enxames foi utilizada pela primeira vez por Geraldo Beni, em seu artigo “The Concept of Cellular Robotic System”. A melhor definição para o termo **enxame** foi citada por James Kennedy e Russel Eberhart, no livro *Swarm Intelligence*, de 2001, como tendo sido retirada da seção FAQ (“Perguntas Frequentes”, em português) do Santa Fé Institute.

Nós usamos o termo “enxame” num sentido geral para fazer referência a qualquer coleção minimamente estruturada de agentes que interajam entre si. O exemplo clássico de enxame é um enxame de abelhas, mas a metáfora de um enxame pode ser estendida a outros sistemas com uma arquitetura similar. Uma colônia de formigas pode ser interpretada como um enxame cujos agentes individuais são formigas, um bando de pássaros é um enxame cujos agentes são pássaros, o trânsito é um enxame de carros, uma multidão é um enxame de pessoas, um sistema imunológico é um enxame de células e moléculas, e a economia é um enxame de agentes econômicos. Apesar de a noção de enxame sugerir um aspecto de movimento coletivo no espaço, como num enxame de um bando de pássaros, nós estamos interessados em todos os tipos de comportamentos coletivos, não apenas no movimento no espaço (Santa Fé Institute, [19--?] *apud* Kennedy; Eberhart, 2001, p. 102, tradução nossa).

Para Bonabeau, Dorigo e Thearulaz (1999), insetos são criaturas complexas que possuem uma série de sensores, modulam seu comportamento de acordo com variados estímulos, incluindo a interação com seus vizinhos. No entanto, segundo eles, a complexidade de um inseto, individualmente, não é capaz de explicar a complexidade que as colônias de insetos apresentam, o que leva à questão: como a cooperação entre os indivíduos surge?

De acordo com Nicolis (1977) e Haken (1983), citados por Bonabeau, Dorigo e Theraulaz (1999), as teorias da auto-organização surgiram nos campos da física e química para descrever a emergência de padrões macroscópicos a partir de processos e interações que ocorrem em nível microscópico. Em muitos aspectos, os coletivos de insetos possuem essa mesma característica, portanto as teorias da auto-organização que des-

crevem a emergência de padrões macroscópicos a partir de interações microscópicas podem ser estendidas para os insetos, para mostrar que comportamentos complexos podem emergir coletivamente a partir da interação de indivíduos extremamente simples individualmente.

Num ambiente de constantes desafios à sobrevivência, a adaptação ao meio é algo fundamental para a perpetuação das espécies. Segundo Kennedy e Eberhart (2001), o comportamento social ajuda os indivíduos a se adaptarem ao ambiente em que estão inseridos, fornecendo a cada indivíduo do grupo mais informações do que ele seria capaz de obter utilizando apenas seus próprios sentidos, algo semelhante a um processo de aprendizado.

Jonhson (2012) pontua que o aprendizado das formigas ocorre em nível de colônia, enquanto individualmente as formigas mantêm seu comportamento primitivo e um completo desconhecimento em relação ao todo. Ele argumenta que o cérebro da colônia é a soma de milhares de decisões extremamente simples tomadas individualmente por cada uma das formigas.

Segundo Jonhson (2012), o aprendizado não é sempre um processo dependente da consciência. Ele cita como exemplo o sistema imunológico dos seres humanos, que passa toda a vida aprendendo, adicionando novos anticorpos ao seu vocabulário; esse vocabulário cresce como resposta às ameaças que o sistema enfrenta. Ele argumenta que, da mesma forma que o corpo humano aprende de maneira inconsciente, os coletivos de insetos fazem o mesmo; o aprendizado não se trata apenas da ciência de novas informações, mas de ser capaz de reconhecer e responder a mudanças de padrões.

Ainda nesse contexto do aprendizado das formigas, Kennedy e Eberhart (2001) questionam qual seria a diferença entre adaptação e inteligência, já que, segundo eles, alguns escritores argumentam que a inteligência é justamente a habilidade de se adaptar. Eles veem o aprendizado como um processo social. De acordo com eles, se uma pessoa tivesse que construir todo o seu conhecimento a partir de suas próprias ações e experiências, seu processo cognitivo seria bastante retardado. Além disso, grande parte do processo natural de aprendizado de uma pessoa se dá por emulação de comportamentos exibidos e validados por outros indivíduos na sociedade.

## 2.1 OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS

Os primeiros passos em sentido ao desenvolvimento de aplicações baseadas em formigas artificiais criadas computacionalmente foram dados por biólogos, que estudaram o comportamento de insetos e conseguiram descrever como esses insetos se comunicavam. Grassé (1959 *apud* Dorigo; Stützle, 2004), zoologista, introduziu o conceito de estigmergia, que consistia numa forma de comunicação indireta entre insetos por meio de modificações provocadas no ambiente. O mecanismo foi descrito originalmente para duas espécies de cupins, mas o termo foi posteriormente utilizado para descrever a comunicação entre outros tipos de insetos que apresentam comportamento social.

No caso das formigas numa colônia, a comunicação se dá por meio de feromônios, substância química depositada no chão pelo percurso entre a colônia e a fonte de comida, como demonstrado por Goss *et al.* (1990) e Deneubourg *et al.* (1990).

O depósito de feromônios na forma de trilhas é característica de algumas espécies de formigas, que utilizam esse depósito como uma forma de marcar um caminho. Formigas em busca de comida sentem o cheiro desse feromônio depositado, e dessa forma podem seguir os caminhos já encontrados por outras formigas. Esse processo funciona como um *feedback* positivo, e é um exemplo do comportamento emergente das formigas: um padrão macroscópico emerge das interações entre indivíduos em nível microscópico (Dorigo; Stützle, 2004). Essa característica inspirou a criação de toda uma área da computação, chamada de Otimização por Colônias de Formigas, que faz referência a todos os algoritmos que implementam computacionalmente as características que as formigas apresentam na natureza em sua busca por comida.

Segundo Dorigo e Stützle (2004), a primeira versão de algoritmo de Otimização por Colônias de Formigas foi o Ant System, proposto por Marco Dorigo inicialmente em 1991 e posteriormente apresentado em sua tese de doutorado em 1992. O algoritmo foi proposto para resolver diferentes variações do Problema do Caixeiro Viajante (um problema NP-difícil)<sup>1</sup>, com 30, 50 e 75 cidades. Essa primeira versão não se mostrou competitiva se comparada a outros tipos de técnicas disponíveis na época para a

solução do Problema do Caixeiro Viajante (Dorigo; Stützle, 2009), o que leva a um fato: a principal contribuição dessa primeira versão foi a criação de uma nova meta-heurística, baseada na simulação computacional do comportamento das colônias de formigas. Nos anos seguintes, diversas versões foram propostas com base nessa primeira versão. O Quadro I mostra as versões de algoritmos de Otimização por Colônias de Formigas que foram criados entre 1991 e 2004.

Em 1999, Dorigo e Di Caro definiram a meta-heurística da Otimização por Colônias de Formigas como uma estrutura básica para os algoritmos que implementavam heurísticas baseadas na simulação computacional das colônias de formigas. “Meta-heurísticas são métodos de solução que orquestram uma interação entre procedimentos de melhoria local e estratégias de nível superior para criar um processo capaz de escapar de ótimos locais e realizar uma busca robusta no espaço de soluções” (Gendreau; Potvin, 2010, p. ix., tradução nossa).

Portanto, a partir de 1999, ficou estabelecido que o termo “Otimização por Colônias de Formigas” não fazia referência a um único algoritmo, mas a uma classe de algoritmos com estruturas semelhantes, baseadas na simulação computacional das colônias de formigas.

Na criação computacional de formigas artificiais, são copiadas ou adaptadas as características mais importantes das formigas reais, e criadas características adicionais que permitem que as formigas encontrem o melhor caminho (de menor distância) em gráficos genéricos. As formigas artificiais armazenam a informação de quais nós já foram visitados por elas; constroem seu caminho (solução) por meio da seleção probabilística com base nas trilhas de feromônios; não depositam feromônios na construção do seu caminho, mas no posterior retorno do destino para a origem; e são capazes de avaliar a qualidade da solução criada por elas (Dorigo; Stützle, 2004).

Outra diferença importante é o papel que a evaporação de feromônios tem nas colônias de formigas artificiais, muito mais importante do que no caso das colônias reais. Para que o algoritmo seja capaz de encontrar soluções próximas à ideal (considerando a complexidade dos problemas resolvidos pelos algoritmos em relação à complexidade dos problemas resolvidos por formigas reais), é necessário haver um correto balanço entre

<sup>1</sup> Há uma relação homoerótica entre os dois personagens claramente descrita com naturalidade na obra.

Quadro I – Principais tipos de ACO (Ant Colony Optimization) criados entre 1991 e 2004.

Algoritmo de Otimização por Colônias de Formigas	Autores	Ano
Ant System	Alberto Corloni, Marco Dorigo e Vittorio Maniezzo	1991
Elitist AS	Marco Dorigo, Vittorio Maniezzo e Alberto Corloni	1992
Ant-Q	Alfonso Gambardella e Marco Dorigo	1995
Ant Colony System	Alfonso Gambardella e Marco Dorigo	1996
MMAS	Tomas Stützle e Houger Hoos	1996
Rank-based AS	Bernd Bullnheimer, Richard Hartl e Claude Strauss	1997
Best-Worst AS	Oscar Cordón, Inaki Fernandez de Viana e Francisco Herrera	2000
Population-based ACO	Michael Guntsch e Martin Middendorf	2002
Beam-ACO	Christian Blum	2004

6

Fonte: Adaptado de Dorigo e Stützle (2009, p. 10)

a exploração de áreas já encontradas e a prospecção de novas áreas, o que é feito balanceando a intensidade das trilhas de feromônios (Dorigo; Stützle, 2009). Para contribuir com esse balanceamento, o efeito da evaporação de feromônios foi amplificado em relação à realidade. De acordo com Dorigo e Stützle (2004), nas colônias de formigas reais essa evaporação não tem papel importante, já que ocorre muito lentamente. Já nas colônias artificiais, ela passou a ser utilizada como um mecanismo para diminuir progressivamente a intensidade das trilhas, impedindo que as trilhas se tornem intensas a ponto de não poderem ser evitadas, e favorecendo a exploração de novas soluções.

### 2.1.1 FORMULAÇÃO MATEMÁTICA

Considerando o Quadro I, cronologicamente o Ant Colony System (ACS) apresenta uma diferença grande em relação aos anteriores, que é a utilização de um parâmetro para fazer o balanço entre a prospecção e a exploração no algoritmo. Portanto, ele será utilizado como base para o detalhamento de uma formulação matemática. A formulação matemática dos algoritmos da meta-heurística da Otimização por Colônias de For-

migas envolve: regra de transição (probabilística); atualização global de feromônios; e atualização local de feromônios.

#### 2.1.1.1 REGRA DE TRANSIÇÃO PROBABILÍSTICA PARA O ACS

A regra de transição (probabilística) para o ACS define que uma formiga localizada no nó  $r$  selecionará o nó  $s$  para se mover aplicando a seguinte regra (Equação 1):

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \left\{ [\tau(r, u)] \cdot [\eta(r, u)]^\beta \right\}, \\ \text{se } q \leq q_0 \text{ (exploração)} \\ S, \text{ caso contrário (prospecção enviesada)} \end{cases} \quad (\text{Eq. 1})$$

Nessa equação:  $\tau(r, u)$  é a concentração de feromônios entre os nós  $r$  e  $u$ ;  $\eta(r, u)$  é o valor heurístico (inverso da distância) entre os nós  $r$  e  $u$ ;  $\tau$  é o parâmetro que controla a importância relativa do valor heurístico no



algoritmo;  $q$  é um valor randômico uniformemente distribuído entre 0 e 1; e  $q_0$  é um parâmetro entre 0 e 1, definido para balancear a relação entre a exploração de áreas já encontradas e a prospecção de novas áreas.  $S$  é uma variável randômica selecionada de acordo com a distribuição de probabilidades dada pela Equação 2:

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, s)] \cdot [\eta(r, s)]^\beta}, & \text{se } s \in J_k(r) \\ 0, & \text{caso contrário} \end{cases} \quad (\text{Eq. 1})$$

Nas Equações 1 e 2, que dizem respeito ao ACS, uma diferença que vale a pena pontuar em relação ao Ant System é que a concentração de feromônios não está elevada a  $\alpha$ , parâmetro que controla a relevância relativa da concentração de feromônios no algoritmo. Assim, o balanço entre as importâncias da concentração de feromônios e do valor heurístico é controlada unicamente pelo parâmetro  $\beta$ . Isso acontece porque no ACS o balanço entre prospecção e exploração é feito pelo parâmetro  $q_0$ , o que não ocorre no Ant System.

### 2.1.1.2 ATUALIZAÇÃO GLOBAL DE FEROMÔNIOS PARA O ACS

A atualização global de feromônios para o ACS ocorre de acordo com a Equação 3:

$$\tau(r, s) = (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s) \quad (\text{Eq. 3})$$

onde:  $\alpha$  é o parâmetro de decaimento de feromônios; e  $\Delta\tau(r, s)$  é a variação na concentração de feromônios no trecho entre os nós  $r$  e  $s$ . O valor de  $\Delta\tau(r, s)$  é igual ao inverso das distâncias entre os nós  $r$  e  $s$  se esse trecho fizer parte da melhor solução, e é igual a 0 em caso contrário.

### 2.1.1.3 ATUALIZAÇÃO LOCAL DE FEROMÔNIOS PARA O ACS

A atualização local de feromônios é realizada por todas as formigas a cada iteração, atuando apenas sobre o último trecho percorrido pela formiga

(Dorigo; Birattari; Stützle, 2006). Ela ocorre de acordo com a Equação 4:

$$\tau(r, s) = (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \quad (\text{Eq. 4})$$

onde:  $\rho$  define a velocidade de decaimento das concentrações de feromônios; e  $\tau_0$  é a concentração inicial de feromônios no trecho entre os nós  $r$  e  $s$ .

## 2.2. PROBLEMA DE MÚLTIPLOS CAIXEIROS VIAJANTES

Todos os algoritmos que implementam a meta-heurística da Otimização por Colônias de Formigas citados até aqui foram inspirados no Problema do Caixeiro Viajante original, que consiste em um cenário de  $n$  cidades e um único caixeiro viajante. O caixeiro precisa sair da origem, visitar, uma única vez, cada uma das cidades e retornar à origem, percorrendo a menor distância possível. O Problema do Caixeiro Viajante pode ser considerado como um caso particular do Problema de Múltiplos Caixeiros Viajantes, que consiste num cenário de  $n$  cidades e  $m$  caixeiros viajantes. Pelo menos um dos  $m$  caixeiros precisa visitar cada uma das  $n$  cidades; uma cidade não pode ser visitada por dois caixeiros; todos os caixeiros precisam sair da mesma origem. Portanto, o Problema do Caixeiro Viajante consiste no caso do Problema de Múltiplos Caixeiros Viajantes em que  $m = 1$ .

Apesar do Problema de Múltiplos Caixeiros Viajantes ser uma generalização do caso original, ele tem sido muito menos estudado ao longo do tempo. Outros tipos de meta-heurísticas já foram utilizados para sua solução, incluindo: algoritmos genéticos, recozimento simulado e pesquisa tabu (Salas et al., 2012). No caso da meta-heurística da Otimização por Colônias de Formigas, diferentes algoritmos ou técnicas foram propostas, por exemplo: por Svestka e Huckfeldt (1973), Bellmore e Hong (1974), Junjie e Dingwei (2006), Vallivaara (2008 apud Barbosa; Silva Jr.; Kashiwabara, 2015) e Salas et al. (2012).

Lupoiae et al. (2019) e Lu e Yue (2019) dissertaram sobre a diferença entre a obtenção simplesmente da menor distância total e a obtenção da menor distância total associada a um balanceamento eficiente entre os diferentes times. Segundo Lupoiae et al. (2019), na solução do Problema de Múltiplos Caixeiros Viajantes, quase tão importante quanto minimizar a distância total é balancear o tamanho dos percursos individuais de cada um dos times.

A técnica proposta por Vallivaara endereçou essa questão de maneira eficiente, o que justifica a opção por ela num universo com outras opções.

Vallivaara adaptou o ACS para o Problema de Múltiplos Caixeiros Viajantes. A cada iteração, a formiga de menor rota parcial seleciona seu próximo nó de acordo com a regra de seleção probabilística do ACS. No entanto, antes que a formiga se mova efetivamente, é verificado se, caso o nó selecionado pela formiga de menor rota parcial fosse cedido a qualquer outra formiga, isso resultaria numa menor distância total da solução. Em caso positivo, a formiga de menor rota parcial não se move, e o nó é atribuído à rota da formiga que resultará na menor distância total da solução (Barbosa; Silva Jr.; Kashiwabara, 2015).

### 3. APLICAÇÃO DA OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS COM TIMES AO PLANEJAMENTO DA MANUTENÇÃO DAS ÁREAS VERDES EM CAMPUS UNIVERSITÁRIOS

8

Esta seção descreve o plug-in desenvolvido nesta pesquisa, que aplica a meta-heurística da Otimização por Colônias de Formigas à otimização do planejamento da manutenção de áreas verdes em campi universitários. Esse planejamento envolve o correto dimensionamento da equipe de jardinagem, de tal forma que ela seja capaz de realizar a manutenção de todas as áreas definidas num período determinado. Portanto, o resultado da simulação será o número mínimo de profissionais que a equipe deve ter para que os serviços de manutenção sejam completados no intervalo definido. Se houvesse um único trabalhador, o problema poderia ser tratado como uma instância do Problema do Caixeiro Viajante, mas, como a equipe tem múltiplos trabalhadores, ele deve ser tratado como instância do Problema de Múltiplos Caixeiros Viajantes.

Para a solução do problema, foi implementado o algoritmo proposto por Vallivaara (2008 apud Barbosa; Silva Jr.; Kashiwabara, 2015): o Team Ant Colony Optimization, que aplica o ACS ao Problema de Múltiplos Caixeiros Viajantes. A simulação começa com apenas um caixeiro viajante; um novo caixeiro é adicionado a cada novo ciclo da simulação, até que o algoritmo indique que o número de caixeiros é suficiente. O plug-in foi concebido para o programa AutoCAD, aproveitando-se do fato de ele ser o programa de desenho mais utilizado nas áreas de arquitetura, engenharia e construção,

e por possuir interface com o Earthstar Geographic SIO, serviço de visualização de imagens de satélites da Microsoft. Dentro da área de trabalho do AutoCAD, o plug-in cria uma guia (ribbon tab) adicional (Green Maintenance Optimizer) na parte superior da tela (Figura 1), com todos os seus comandos (botões) distribuídos em painéis específicos.

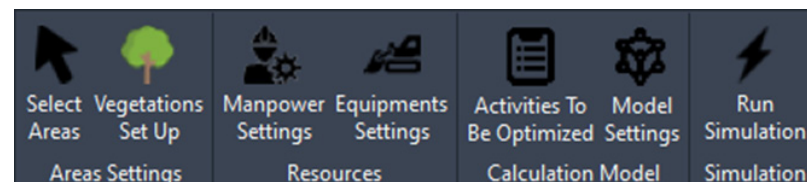


Figura 1 – Ribbon Tab Green Maintenance Optimizer.  
Fonte: Elaborada pelos autores (2024)

De acordo com o Autodesk (2013), há três interfaces de programação (em inglês, *Application Programming Interface*, ou API) disponíveis para o AutoCAD: .NET API, COM API e Custom Draw API. No entanto, quando possível, a interface .NET deve ser adotada para trabalhos mais complexos, devido à acessibilidade facilitada aos objetos do programa e ao seu melhor desempenho em geral. Dentro do universo de linguagens .NET, o C# foi escolhido para o desenvolvimento da aplicação devido ao predomínio de sua utilização no contexto de aplicações para o sistema operacional Windows e à vasta disponibilidade de materiais de consulta.

Por tratar-se de um *plug-in*, ele utiliza o programa AutoCAD como plataforma, e pode fazer uso da maioria de seus comandos nativos. A ferramenta *Set Location* do AutoCAD permite definir a localização do desenho. Sendo assim, para o desenvolvimento do *plug-in*, o campus da Universidade Federal do Espírito Santo, localizado no bairro Goiabeiras, em Vitória, no Espírito Santo, foi utilizado como modelo. No entanto, ele pode ser aplicado a outros campi universitários ou a outras áreas de características semelhantes, como parques municipais ou clubes, sem maiores problemas.

Após definida a localização do desenho, as áreas de interesse (contorno do campus, correspondente ao limite até onde se supõe que os trabalhadores podem ir, e áreas de gramado) devem ser desenhadas no mapa utilizando polilinhas. Posteriormente, o plug-in permite a seleção de cada área de acordo com seu tipo (Figura 2), identificando se são áreas de gramado, de passagem de veículos e de passagem de pedestres – por padrão –, além de ser possível adicionar novas configurações/definições de área. Na



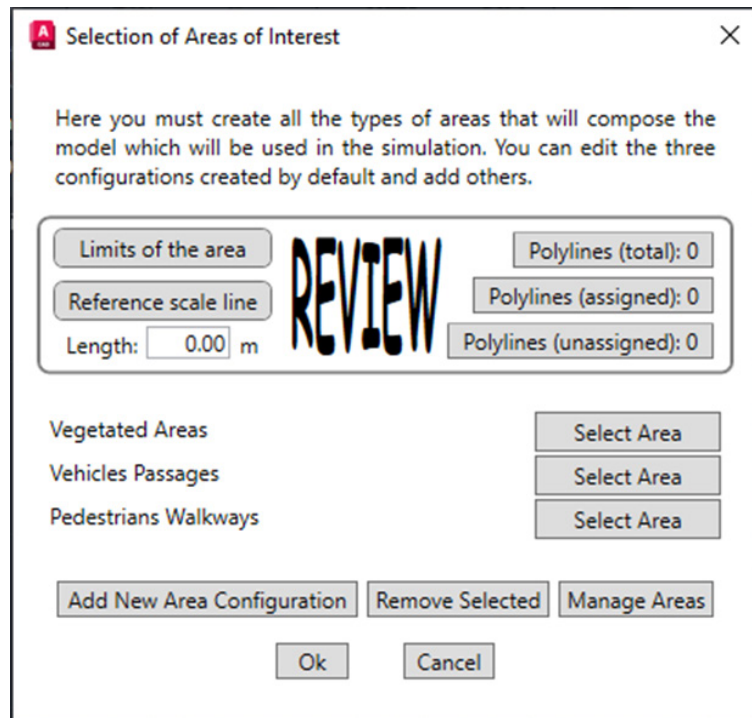


Figura 2 – Janela “Selection of Areas of Interest”.  
Fonte: Elaborada pelos autores (2024)

versão atual do *plug-in*, apenas a seleção das áreas de gramado influencia a simulação, mas pretende-se que em versões futuras se permita estabelecer áreas específicas que poderiam ser utilizadas como rotas dos trabalhadores

Outra seleção importante para a simulação é indicar a origem das formigas, ou seja, de onde partiriam os trabalhadores (por exemplo, o alojamento de onde os trabalhadores iniciam a jornada) (Figura 3). A quantidade de tipos de trabalhadores, também exibida na janela “Manpower Settings”, não pode ser editada, e está relacionada aos tipos de trabalhador existentes nos serviços considerados para execução no campus, considerando a tabela de custos unitários do *plug-in* (itens com a coluna “Type” definida como “Labour”) (Figura 4). Qualquer serviço precisa estar corretamente orçado para que possa ser atribuído. A cor verde indica que está tudo

correto com o orçamento; caso houvesse algum erro, o item estaria na cor vermelha, indicando problema. Nessa versão do *plug-in* ainda não é possível adicionar novos serviços à tabela, mas essa funcionalidade é prevista para ser

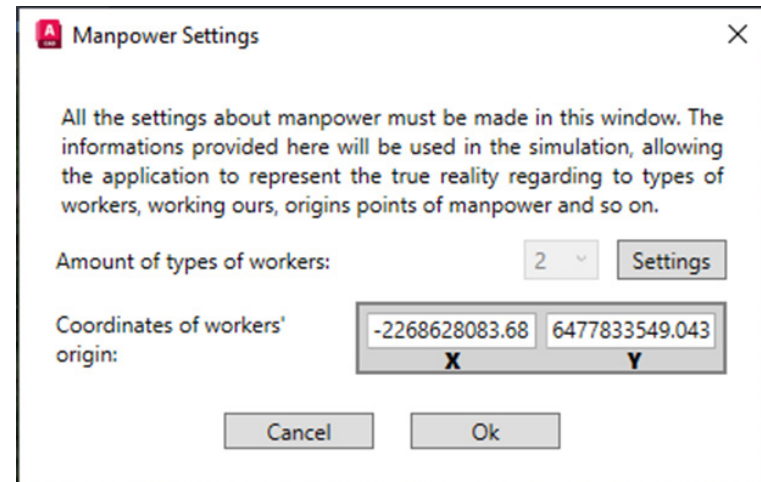


Figura 3 – Janela “Manpower Settings”.  
Fonte: Elaborada pelos autores (2024)

Service	Component	Type	Unit	Consumption	Unit Cost (R\$)
CORTE DE ÁREAS GRAMADAS (1 M²)					
	Caminhão carroceria – 4 t (80 kW)	MACHINERY	h (produtiva)	0,0005	226,46
			h (improdutiva)	0,0005	56,99
	Roçadeira – em micro trator (10 kW)	MACHINERY	h (produtiva)	0,001	133,01
	Encarregado	LABOUR	h	0,0001	39,39
	Servente	LABOUR	h	0,002	21,28
IRRIGAÇÃO DE ÁREA VERDE COM CAMINHÃO (1 M²)					
	Servente	LABOUR	h	0,0025	21,28
	Água	RAW MATERIAL	m³	0,0075	4,00
	Caminhão irrigadeira, diesel, potência 167 HP (125 kW), capacidade 8.000 l - vida útil 8.000 h	MACHINERY	h produtiva	0,00125	79,46
IRRIGAÇÃO DE ÁREA VERDE COM PULVERIZADOR (1 M²)					
	Servente	LABOUR	h	0,0012	21,28
	Água	RAW MATERIAL	m³	0,000002	4,00
	Pulverizador costal com capacidade de 20 litros	MACHINERY	h	0,0012	0,69

Figura 4 – Janela “Table of Unit Costs Breakdown”.  
Fonte: Elaborada pelos autores (2024)

implementada na versão futura.

A Figura 5 exibe como o campus é visualizado dentro do AutoCAD após a criação dos contornos das áreas de interesse com polilinhas. Apesar de o objetivo da aplicação ser dimensionar a equipe de jardinagem, com base no tempo de execução dos serviços, a versão atual do plug-in otimiza apenas a distância, então as configurações específicas ainda não são necessárias para iniciar a simulação. Quando se clica no botão Run Simulation (Figura 1), abre-se a janela de simulação; no entanto a simulação ainda não se inicia. Deve-se ajustar as configurações de: escala da grade de pontos e número de iterações. Para efetivamente iniciar a simulação, deve-se clicar novamente no botão “Run Simulation”, dessa vez dentro da janela.

A Figura 6 mostra o resultado da simulação considerando uma grade de pontos com espaçamento de cinquenta metros e um número de iterações igual a cem. O resultado apresenta apenas um percurso (como se se tratasse de uma instância do Problema do Caixeiro Viajante, e não do Problema de Múltiplos Caixeiros Viajantes) porque o cálculo do tempo do percurso ainda não está implementado, e o aumento do número de trabalhadores

acontecerá como resultado justamente de tempos de percursos que excedam a jornada de trabalho.

Cada ponto de grid dentro de áreas de gramado corresponde a um nó no Problema de Múltiplos Caixeiros Viajantes a ser resolvido e a uma fração da área em que está inserido. Na versão atual, o tempo de estadia em cada nó não está implementado, mas quando estiver, corresponderá ao tempo que será adicionado ao total do trajeto da formiga que passar pelo nó. O tempo adicionado será o produto da multiplicação de dois valores: (1) o fator aplicado ao insumo de mão de obra no custo unitário do serviço a ser realizado na área que contém o nó; e (2) a área correspondente àquele nó.

A escala da grade de pontos – que pode ser de um, cinco, dez, vinte, vinte e cinco, cinquenta ou cem metros – tem o objetivo de reduzir a complexidade computacional do problema, reduzindo o número de nós no Problema de Múltiplos Caixeiros Viajantes a ser resolvido. Naturalmente, um espaçamento de cinquenta metros entre os pontos da grade resultará numa complexidade muito menor do que um espaçamento de um metro.

Apesar da otimização considerando tempos ainda não estar implementada,

10



Figura 5 – Visualização do campus de Goiabeiras dentro do AutoCAD, com todas as áreas de interesse selecionadas.  
Fonte: Elaborada pelos autores (2024)



Figura 6 – Janela com o resultado do Team Ant Colony Optimization para as áreas de gramado selecionadas.  
Fonte: Elaborada pelos autores (2024)

o plug-in já permite a configuração dos trabalhadores em relação à hora de início da jornada, início do horário de almoço, fim do horário de almoço e hora final da jornada de trabalho (Figura 7). Na próxima versão do plug-in, considerando uma jornada das 8:00 às 12:00 e das 13:00 às 17:00, um mesmo jardineiro (um mesmo time de formigas) poderá realizar dois percursos de até quatro horas num mesmo dia, mas se o percurso de um time de formigas tiver uma duração de mais de quatro horas, isso indica que mais times (ou seja, mais jardineiros) são necessários. A coluna “Tasks Performed” permitirá que determinados trabalhadores sejam utilizados para tarefas específicas. No entanto, ela ainda não está sendo utilizada nessa versão.

O plug-in também permite a configuração da frequência de realização das atividades nos gramados, considerando a existência de dois cenários: (1) verão e primavera; e (2) outono e inverno (Figura 8). Quando o tempo de estadia em cada nó estiver implementado, essa frequência será utilizada, em conjunto com a periodicidade estabelecida para a completa realização dos serviços no campus, para determinar o peso que cada atividade terá nos nós do problema.

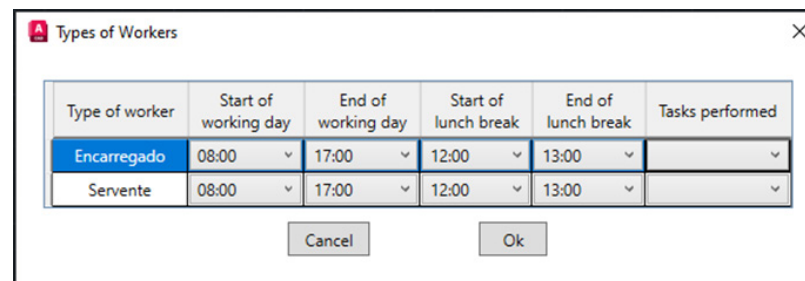


Figura 7 – Janela “Types of Workers”.  
Fonte: Elaborada pelos autores (2024)

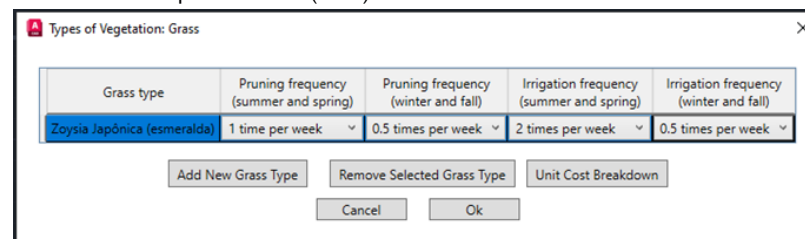


Figura 8 – Janela “Types of Vegetation: Grass”.  
Fonte: Elaborada pelos autores (2024)

## 4. CONCLUSÃO

Este artigo buscou fazer um registro histórico do surgimento da inteligência de enxames, como parte dos esforços de criação da computação inspirada na natureza, assim como das iniciativas que levaram ao surgimento da meta-heurística da Otimização por Colônias de Formigas. Ele apresentou a formulação do Problema do Caixeiro Viajante e justificou como esse problema foi importante para a proposição de todos os principais algoritmos que fazem parte dessa meta-heurística. Em seguida, apresentou o Problema de Múltiplos Caixeiros Viajantes, que nada mais é do que uma generalização do problema anterior.

Após essa breve contextualização, o artigo listou diferentes meta-heurísticas utilizadas para a solução do Problema de Múltiplos Caixeiros Viajantes, e as diferentes iniciativas para aplicar a meta-heurística da Otimização por Colônias de Formigas ao problema.

Finalmente, o artigo descreveu brevemente a criação de um plug-in para o programa AutoCAD, que interpreta o problema da otimização do planejamento da manutenção de áreas verdes em campi universitários como uma analogia ao Problema de Múltiplos Caixeiros Viajantes, e utiliza o algoritmo proposto por Vallivaara para sua solução.

Por se tratar de um plug-in para o AutoCAD, uma das aplicações mais utilizadas para projetos na arquitetura e na engenharia, o software que compila os resultados desta pesquisa se constitui como uma iniciativa robusta no sentido de popularizar o uso da inteligência de enxames no ambiente das atividades correntes de paisagismo nos campi das instituições de ensino superior do país e em áreas de características similares, num contexto em que outros tipos de computação inspirada na natureza, como a inteligência artificial, têm se popularizado cada vez mais.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AUTODESK. **API developer's guide**: AutoCAD Civil 3D 2013. [S. l.: s. n.], 2013.
- BALL, Philip. Forging patterns and making waves from biology to geology: a commentary on Turing. **Philosophical Transactions of the Royal Society B**, London, v. 370, n. 1666, 2015. DOI: 10.1098/rstb.2014.0218.
- BAR-YAM, Yaneer. Reviews (book & software): complex systems with Herbert Simon. **Complexity**, v. 5, n. 3, p. 47-48, 1998.
- BARBOSA, Denilson; SILVA JR., Calos; KASHIWABARA, André. Aplicação da otimização por colônia de formigas ao problema de múltiplos caixeiros viajantes no atendimento de ordens de serviço nas empresas de distribuição de energia elétrica. In: BRAZILIAN SYMPOSIUM ON INFORMATION SYSTEM, 11., 2015, Goiânia. **Anais [...]**. Goiânia: SBC, 2015. p. 23-30.
- BELLMORE, Mandell; HONG, Saman. Transformation of multisalesman problem to the standard traveling salesman problem. **Journal of the Association for Computing Machinery**, [s. l.], v. 21, n. 33, p. 500-504, 1974. DOI: 10.1145/321832.321847.
- BONABEAU, Eric; DORIGO, Marco; THERAULAZ, Guy. **Swarm intelligence: from natural to artificial systems**. New York: Oxford University Press, 1999.
- DENEUBOURG, Jean-Louis et al. The self-organizing exploratory pattern of the Argentine ant. **Journal of Insect Behavior**, New York, v. 3, n. 2, p. 159-168, 1990. DOI: 10.1007/BF01417909.
- DORIGO, Marco; BIRATTARI, Mauro; STÜTZLE, Thomas. **Ant colony optimization: artificial ants as a computational intelligence technique**. Bruxelas: [s. n.], 2006.
- DORIGO, Marco; DI CARO, Gianni. Ant colony optimization: a new meta-heuristic. In: CONFERENCE EVOLUTIONARY COMPUTATION, 99., 1999, New York. **Anais [...]**. New York: IEEE, 1999. p. 1470-1477.
- DORIGO, Marco; STÜTZLE, Thomas. **Ant colony optimization**. Cambridge: MIT Press, 2004.
- DORIGO, Marco; STÜTZLE, Thomas. **Ant colony optimization: overview and recent advances**. Bruxelas: IRIDIA, 2009.
- GENDREAU, Michel; POTVIN, Jean-Yves. **Handbook of metaheuristics**. New York: Springer Science, 2010.
- GOSS, S. et al. How trail laying and trail following can solve foraging problems for ant colonies. **Behavioural Mechanisms of Food Selection**, New York, p. 661-678, 1990. DOI: 10.1007/978-3-642-75118-9\_32.
- HOLLAND, John. Genetic algorithms and adaptation. In: SELFRIDGE, Oliver. **Adaptive control of ill-defined systems**. New York: Plenum Press, 1984, cap. 21, p. 317-333.
- HOFSTADTER, Douglas. **Gödel, Escher, Bach: an eternal golden brain: a metaphorical fugue on minds and machines in the spirit of Lewis Carroll**. New York: Basic Books 1979.
- JOHNSON, Steven. **Emergence: the connected lives of ants, brains, cities and software**. Marin: Simon & Schuster, 2012.
- JUNJIE, Pan; DINGWEI, Wang. An ant colony optimization algorithm for multiple travelling salesman problem. In: INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING, INFORMATION AND CONTROL, 1., 2006, Beijing. **Anais [...]**. Beijing: IEEE, 2006. p. 1-4.1
- KENNEDY, James; EBERHART, Russel; SHI, Yuhui. **Swarm intelligence**. San Francisco: Morgan Kaufmann, 2001.
- LANGTON, Christopher. Studying artificial life with cellular automata. **Physica D: Nonlinear Phenomena**, Amsterdam, v. 22, n. 1-3, p. 120-149, 1986. DOI: 10.1016/0167-2789(86)90237-X.
- LU, Li-Chih; YUE, Tai-Wen. Mission-oriented ant-team ACO for min-max MTSP. **Applied Soft Computing Journal**, Amsterdam, v. 76, p. 436-444, 2019. DOI: 10.1016/j.asoc.2018.11.048.

LUPOAIE, Vlad-loan et al. SOM-guided evolutionary search for solving MinMax Multiple-TSP. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, 2019, Wellington. **Anais** [...]. Wellington: IEEE, 2019. p. 73-80.

ROCHA, Bruno; BOLSSONI, Gabriela; BUSSOLOTI, Victor. Ecologias de projeto: métodos e processos em arquitetura digital. In: FÓRUM DE PESQUISA MACKENZIE, 9., 2019, São Paulo. **Anais** [...]. São Paulo: [s. n.], 2019. p. 265-276.

SALAS, Yasel et al. Multi-type ant colony system for solving the multiple traveling salesman problem. **Revista Técnica de la Facultad de Ingeniería Universidad del Zulia**, Maracaibo, v. 35, n. 3, p. 311-320, 2012.

SIMON, Herbert. **The sciences of the artificial**. Cambridge: MIT Press, 1996.

SVETKA, Joseph; HUCKFELDT, Vaughn. Computational experience with an M-Salesman traveling salesman algorithm. **Management Science, Catonsville**, v. 19, n. 7, p. 790-799, 1973. DOI: 10.1287/mnsc.19.7.790

TURING, Alan. **The prof's book**: Turing's treatise on the Enigma. [S. l.: s. n.], 1940. Disponível em: <https://archive.org/details/hw-25-3>. Acesso em: 4 dez. 2024.

André Barcellos Ferreira

Doutorando do curso de Arquitetura e Urbanismo da Universidade Federal do Espírito Santo. Vitória, Espírito Santo, Brasil.

CV: <http://lattes.cnpq.br/1142310679653189>

[andrebarcellosferreira@gmail.com](mailto:andrebarcellosferreira@gmail.com).

Jarryer Andrade de Martino

Docente do curso de Arquitetura e Urbanismo da Universidade Federal do Espírito Santo.

Universidade Federal do Espírito Santo/Centro de Artes/Departamento de Arquitetura e Urbanismo

CV: <http://lattes.cnpq.br/1222973320203284>

Orcid: <https://orcid.org/0000-0002-5542-6039>

[jarryer.martino@ufes.br](mailto:jarryer.martino@ufes.br).

Nota do Editor

Revisão do texto: Tikinet