

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM ARQUITETURA E
URBANISMO**

ANDRÉ BARCELLOS FERREIRA

**OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS APLICADA AO
DIMENSIONAMENTO DE EQUIPES DE MANUTENÇÃO DE ÁREAS
VERDES EM CAMPI UNIVERSITÁRIOS**

**VITÓRIA
2026**

ANDRÉ BARCELLOS FERREIRA

**OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS APLICADA AO
DIMENSIONAMENTO DE EQUIPES DE MANUTENÇÃO DE ÁREAS
VERDES EM CAMPI UNIVERSITÁRIOS**

Tese apresentada ao Programa de Pós-Graduação em Arquitetura e Urbanismo da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do título de Doutor em Arquitetura e Urbanismo.

Orientador: Prof. Dr. Jarryer Andrade de Martino

VITÓRIA

2026

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

F383o Ferreira, André Barcellos, 1988-
Otimização por Colônias de Formigas aplicada ao dimensionamento de equipes de manutenção de áreas verdes em campi universitários / André Barcellos Ferreira. - 2026.
137 p. : il.

Orientador: Jarryer Andrade de Martino.
Tese (Doutorado em Arquitetura e Urbanismo) -
Universidade Federal do Espírito Santo, Centro de Artes.

1. Otimização combinatória. 2. Programação heurística. 3. Algoritmos. 4. Pesquisa operacional. 5. Planejamento. 6. Manutenção. I. Martino, Jarryer Andrade de. II. Universidade Federal do Espírito Santo. Centro de Artes. III. Título.

CDU: 72


ANDRÉ BARCELLOS FERREIRA

“OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS APLICADA AO
DIMENSIONAMENTO DE EQUIPES DE MANUTENÇÃO DE ÁREAS
VERDES EM CAMPI UNIVERSITÁRIOS”


Tese apresentada ao Programa de Pós-Graduação em Arquitetura e Urbanismo da Universidade Federal do Espírito Santo, como requisito final para a obtenção do grau de Doutor em Arquitetura e Urbanismo.

Aprovada em 18 de março de 2026.


Comissão Examinadora

Documento assinado digitalmente
 **JARRYER ANDRADE DE MARTINO**
Data: 18/03/2026 17:49:01-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Jarryer Andrade de Martino
(orientador – PPGAU/UFES)


Documento assinado digitalmente
 **JORGE LEONID ACHING SAMATELO**
Data: 19/03/2026 10:20:33-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Jorge Leonid Aching Samatelo
(membro externo – UFES)

Documento assinado digitalmente
 **CARLOS EDUARDO VERZOLA VAZ**
Data: 23/03/2026 17:09:39-0300
CPF: ***.352.948-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Dr. Élcio Cassimiro Alves
(membro externo – UFES)

Prof. Dr. Carlos Eduardo Verzola Vaz
(membro externo – UFSC)

Documento assinado digitalmente
 **MAYCON RICARDO SEDREZ**
Data: 23/03/2026 20:52:08-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Maycon Ricardo Sedrez
(membro externo – Deakin University)



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA



O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
ELCIO CASSIMIRO ALVES - SIAPE 1853259
Departamento de Engenharia Civil - DEC/CT
Em 24/03/2026 às 05:59

Para verificar as assinaturas e visualizar o documento original acesse o link: <https://api-lepisma.prod.uks.ufes.br/arquivos-assinados/1303772?tipoArquivo=O>

DEDICATÓRIA

Dedico este trabalho ao meu pai, Dr. Ayala, brilhante advogado, que só esteve presente nos primeiros oito anos da minha vida, mas cujo legado e cuja memória me ajudaram a chegar até aqui, seguem e seguirão me influenciando, me ajudando a seguir em frente.

Dedico-o também à minha avó materna, Zolmira, que nos deixou durante o desenvolvimento desta tese, que sempre acreditou no meu potencial e cuja presença permanece viva em minhas conquistas.

RESUMO

Esta tese apresenta uma abordagem para que a meta-heurística da Otimização por Colônias de Formigas possa ser utilizada como estratégia para o dimensionamento das equipes de manutenção das áreas verdes em campi universitários, detalhando a implementação computacional dessa metodologia. As quatro versões iniciais de algoritmos da meta-heurística (Ant System, Elitist AS, Ant-Q e Ant Colony System) foram analisadas, e as principais características para que um problema possa ser tratado por eles foram reunidas. Foi adotada uma série de simplificações à questão da manutenção de áreas verdes, para que ela pudesse ser descrita como um problema discreto, geograficamente distribuído, e finalmente como uma instância do Problema de Múltiplos Caixeiros Viajantes. Áreas contínuas passaram a ser representadas por grades de pontos, com distâncias conhecidas entre si. A cada um desses pontos foi atribuída uma fração da área total, e um custo de permanência proporcional à integralidade das atividades a serem realizadas nessa fração da área. A integralização das atividades nas áreas foi estimada com base na análise das características de cada um dos serviços considerados, e em valores de frequência e consumo de mão-de-obra por unidade de serviço retirados de manuais. A abordagem proposta foi implementada como um *plug-in* para o AutoCAD. O campus de Goiabeiras da Universidade Federal do Espírito Santo foi utilizado como estudo de caso, e diferentes cenários foram simulados e comparados. Ainda que não houvesse dados reais de comparação para os cenários simulados, a abordagem proposta mostrou consistência, comprovada pelas ordens de grandeza semelhantes dos resultados de diferentes modelos (o modelo proposto para o tratamento das áreas de gramado e o modelo proposto para o tratamento das áreas arborizadas).

Palavras-chave: Manutenção de áreas verdes. Meta-heurística. Otimização por Colônias de Formigas. Problema de Múltiplos Caixeiros Viajantes.

ABSTRACT

This thesis presents an approach for applying the Ant Colony Optimization metaheuristic as a strategy for sizing maintenance teams responsible for green areas in university campuses, detailing the computational implementation of this methodology. The four initial versions of algorithms belonging to the meta-heuristics (Ant System, Elitist AS, Ant-Q and Ant Colony System) were analyzed, and the main characteristics required for a problem to be addressed by these algorithms were compiled. Several simplifications were adopted regarding the issue of green area maintenance, so that it could be described as a discrete, geographically distributed problem, and ultimately as an instance of the Multiple Traveling Salesmen Problem. Continuous areas were represented by grids of points with known pairwise distances. Each of these points was assigned a fraction of the total area and a on-site service cost proportional to the completeness of the activities to be performed in that fraction of the area. The integration of activities within the areas was estimated based on the analysis of the characteristics of each service considered and on frequency and labor consumption values per service unit obtained from technical manuals. The proposed approach was implemented as a plug-in for AutoCAD. The Goiabeiras campus of the Federal University of Espírito Santo was used as a case study, and different scenarios were simulated and compared. Although there were no real data available for comparison among the simulated scenarios, the proposed approach demonstrated consistency, as evidenced by the similar orders of magnitude observed in the results of different models (the proposed model for lawn area management and the proposed model for wooded area management).

Keywords: Maintenance of green areas. Metaheuristic. Ant Colony Optimization. Multiple Traveling Salesman Problem.

LISTA DE FIGURAS

Figura 1 - Ilustração da estigmergia entre as formigas	44
Figura 2 – Mapa de macrozoneamento da UFES	60
Figura 3 – Representação gráfica do TSP	73
Figura 4 – Fluxograma de funcionamento do plug-in	81
Figura 5 – Guia Green Maintenance Optimizer	85
Figura 6 – Visualização do campus de goiabeiras com as áreas desenhadas e a localização das árvores importada	86
Figura 7 – Janela Manpower Settings	87
Figura 8 – Janela Types of Vegetation (aba Grass)	88
Figura 9 – Janela Types of Vegetation (aba Trees)	88
Figura 10 – Janela Activities Setup (aba Grass)	89
Figura 11 – Janela Activities Setup (aba Trees).....	89
Figura 12 – Janela de simulação.....	91
Figura 13 – Tooltip para áreas de gramado	92
Figura 14 – Tooltip para ponto de grade de árvores	92
Figura 15 – Janela de progresso da simulação (1)	93
Figura 16 – Janela de progresso da simulação (2)	93
Figura 17 – Janela de progresso da simulação (4)	94
Figura 18 – Janela de progresso da simulação (5)	94
Figura 19 – Janela de progresso da simulação (5)	95
Figura 20 – Janela <i>Result</i>	96
Figura 21 – Janela <i>Export Result</i>	97
Figura 22 – Conteúdo do arquivo RAR gerado	97
Figura 23 – Conteúdo da pasta PATHS\GRASS\PRISON SYSTEM WORKER	98
Figura 24 – Conteúdo da pasta PATHS\GRASS\OUTSOURCED WORKER	99
Figura 25 – Conteúdo da pasta PATHS\TREES	100
Figura 26 – Conteúdo da pasta SIMULATION STEPS\GRASS\PRISON SYSTEM WORKER\INITIAL SEARCH	101
Figura 27 – Conteúdo da pasta SIMULATION STEPS\GRASS\PRISON SYSTEM WORKER\REFINED SEARCH.....	101
Figura 28 – Conteúdo da pasta SIMULATION STEPS\GRASS\OUTSOURCED WORKER\INITIAL SEARCH	102

Figura 29 – Conteúdo da pasta SIMULATION STEPS\GRASS\OUTSOURCED WORKER\REFINED SEARCH	102
Figura 30 – Conteúdo da PASTA SIMULATION STEPS\TREES\INITIAL SEARCH	102
Figura 31 – Conteúdo da pasta SIMULATION STEPS\TREES\REFINED SEARCH	103
Figura 32 – Primeira página do relatório gerado pelo software	104
Figura 33 – Apresentação do resultado da simulação no relatório (1).....	105
Figura 34 – Imagem do arquivo ANT 20.png, localizado na pasta PATHS\GRASS\OUTSOURCED WORKER.....	106
Figura 35 – Imagem do arquivo ANT 20.png, localizado na pasta PATHS\GRASS\OUTSOURCED WORKER.....	107
Figura 36 – Apresentação do resultado da simulação no relatório (2).....	109

LISTA DE TABELAS

Tabela 1 – Altura e frequência de corte de diferentes espécies de grama.....	62
Tabela 2 – Alturas de corte recomendadas.....	62
Tabela 3 – Altura ideal e frequência de corte.....	63
Tabela 4 – Faixas ótimas de altura de corte.....	63
Tabela 5 – Composição de custo unitário, corte de áreas gramadas – 1.000 m ²	64
Tabela 6 – Irrigação de área verde com caminhão – unidade: m ²	65
Tabela 7 – Irrigação de área verde com pulverizador – unidade: ha.....	65
Tabela 8 – Remoção de espécies invasoras (manual) – unidade: ha.....	66
Tabela 9 – Remoção de espécies invasoras (mecanizada) – unidade: ha.....	66
Tabela 10 – Remoção de espécies invasoras (semi-mecanizada) – unidade: ha.....	66
Tabela 11 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,20 m e menor que 0,40 m.....	67
Tabela 12 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,40 m e menor que 0,60 m.....	67
Tabela 13 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,60 m.....	68
Tabela 14 – Poda em altura de árvore com diâmetro de tronco menor que 0,20 m..	68
Tabela 15 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,20 m e menor que 0,40 m.....	68
Tabela 16 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,40 m e menor que 0,60 m.....	68
Tabela 17 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,60 m.....	69
Tabela 18 – Resultados da busca inicial para diferentes valores da constante N.....	77

LISTA DE QUADROS

Quadro 1 – Principais tipos de ACO criados entre 1991 e 2004	45
--	----

LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i> (Otimização por Colônias de Formigas)
AEC	<i>Architecture, Engineering and Construction</i> (Arquitetura, Engenharia e Construção)
API	<i>Application Programming Interface</i> (Interface de Programação da Aplicação)
FAQ	<i>Frequently Asked Questions</i> (Perguntas Frequentes)
MTSP	<i>Multiple Traveling Salesman Problem</i> (Problema de Múltiplos Caixeiros Viajantes)
TACO	Team Ant Colony Optimization
TSP	<i>Traveling Salesman Problem</i> (Problema do Caixeiro Viajante)
UFES	Universidade Federal do Espírito Santo
UI	<i>User Interface</i> (Interface do Usuário)

SUMÁRIO

DEDICATÓRIA	7
RESUMO	9
ABSTRACT	11
LISTA DE FIGURAS	13
LISTA DE TABELAS	15
LISTA DE QUADROS	17
LISTA DE SIGLAS	19
SUMÁRIO	21
1 INTRODUÇÃO	23
1.1 PROBLEMA E JUSTIFICATIVA DA PESQUISA.....	23
1.2 HIPÓTESE	26
1.3 OBJETIVOS	26
1.4 METODOLOGIA.....	27
1.4.1 Revisão Bibliográfica	28
1.4.1.1 Bases de dados e fontes consultadas	29
1.4.1.2 Descritores e palavras-chave	29
1.4.1.3 Critérios de seleção e descarte	30
1.4.1.4 Percurso bibliográfico e justificativa da ênfase do trabalho	30
1.4.2 Proposição e Modelagem do Problema	31
1.4.3 Seleção e Análise de Algoritmos	32
1.4.4 Implementação e Testes	32
1.4.5 Desenvolvimento do plug-in para O AutoCAD	32
1.5 ESTRUTURA DA TESE	35
2 A INTELIGÊNCIA DE ENXAMES	37
2.1 OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS (PARA O PROBLEMA DO CAIXEIRO VIAJANTE).....	43
2.1.1 Formulação matemática	46
2.1.1.1 Ant System	46
2.1.1.2 Elitist Ant System.....	47
2.1.1.3 Ant Colony System	48
2.1.1.4 Ant-Q.....	49
2.2 OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS (PARA O PROBLEMA DE MÚLTIPLOS CAIXEIROS VIAJANTES).....	49

2.3	APLICAÇÕES DA OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS NOS CAMPOS DE AEC	50
3	PAISAGISMO E MANUTENÇÃO DE ÁREAS VERDES EM CAMPUS UNIVERSITÁRIOS	53
3.1	PAISAGISMO E ÁREAS VERDES URBANAS	53
3.1.1	ÁREAS VERDES NO CONTEXTO DE CAMPUS UNIVERSITÁRIOS.....	54
3.1.2	GESTÃO DE ÁREAS VERDES.....	55
3.1.3	MANUTENÇÃO DE ÁREAS VERDES	56
3.2	OTIMIZAÇÃO DO PLANEJAMENTO DA MANUTENÇÃO DE ÁREAS VERDES (DIMENSIONAMENTO DE EQUIPE).....	58
3.2.1	Elenco das atividades (serviços).....	58
3.2.1.1	Levantamento das frequências das atividades de manutenção em gramados	61
3.2.1.2	Levantamento das frequências das atividades de manutenção em áreas arborizadas	66
4	O SOFTWARE PROPOSTO	71
4.1	LINGUAGENS UTILIZADAS.....	71
4.2	DEFINIÇÃO DO PROBLEMA COMPUTACIONAL	73
4.2.1	Team Ant Colony Optimization.....	75
4.2.2	Modificação implementada no TACO para aprimorar o balanceamento de cargas	76
4.3	CARACTERÍSTICAS DO SOFTWARE.....	77
4.3.1	SIMPLIFICAÇÕES E CONSOLIDAÇÃO DOS DADOS	80
4.4	APRESENTAÇÃO DO SOFTWARE	84
4.4.1	APLICAÇÃO AO CAMPUS DE GOIABEIRAS (CENÁRIO 1)	85
4.4.2	RESULTADOS GERADOS	92
4.4.3	COMPARAÇÃO ENTRE CENÁRIOS	107
5	CONSIDERAÇÕES FINAIS.....	111
6	REFERÊNCIAS BIBLIOGRÁFICAS	115
7	APÊNDICES	125

1 INTRODUÇÃO

Nas últimas décadas, os desafios enfrentados pelas sociedades contemporâneas tornaram-se cada vez mais complexos, impulsionados por fatores como as mudanças climáticas, a crescente escassez de recursos naturais, a pressão por sustentabilidade, a instabilidade econômica global e o avanço acelerado da urbanização. No campo da arquitetura, engenharia e construção (AEC), essa complexidade se manifesta de forma particularmente aguda na gestão eficiente de recursos, no planejamento de equipes operacionais e na otimização de processos técnicos e logísticos. Problemas como a alocação dinâmica de equipes em grandes áreas urbanas, o balanceamento entre eficiência operacional e consumo energético, a previsão de demandas em ambientes com múltiplas variáveis interdependentes e a integração de dados em tempo real para decisões estratégicas são apenas alguns exemplos que ilustram a complexidade envolvida. Tais problemas envolvem volumes massivos de dados, múltiplos critérios conflitantes e variáveis em constante mudança, tornando inviável sua resolução por métodos convencionais. Nesse contexto, a inteligência artificial se destaca como uma alternativa eficaz, por sua capacidade de aprender com dados, identificar padrões complexos e propor soluções adaptativas, oferecendo respostas mais precisas, ágeis e compatíveis com os desafios da contemporaneidade.

Esta tese é mais uma iniciativa que busca aplicar a inteligência artificial no campo da AEC. A proposta inicial foi a aplicação da computação inspirada na natureza à arquitetura. Como componente da computação inspirada na natureza, optou-se por trabalhar com a meta-heurística da Otimização por Colônias de Formigas (ACO, do termo em inglês *Ant Colony Optimization*), e, como componente da arquitetura, a escolha foi por trabalhar com o paisagismo, ou, mais especificamente, a manutenção de áreas verdes em campi universitários. Essas diretrizes definiram as bases do trabalho, e por esse motivo outras abordagens computacionais não foram consideradas.

1.1 PROBLEMA E JUSTIFICATIVA DA PESQUISA

As universidades públicas são parte do Estado, mas são consideradas autarquias, segundo a Lei de Diretrizes e Bases. Como autarquias, elas se autorregulam, com

relação a investimentos, manutenção e gestão de um modo geral. Os recursos financeiros dessas universidades são provenientes da esfera correspondente, sejam da União, no caso de universidades federais, sejam dos Estados, no caso de universidades estaduais; eles estão divididos em dois grupos: recursos de capital (para investimentos) e recursos de custeio (para arcar com as despesas correntes de operação da universidade). Os recursos de custeio compreendem todos os gastos nas mais variadas áreas da universidade, e em cada área esse dinheiro é empregado ao longo do ano por meio de contratações de serviços terceirizados, por meio de licitações ou outras modalidades de contratação admitidas pela lei.

Essas contratações, de acordo com a sistemática legal, podem ser realizadas por meio de disputa (licitação), nas modalidades convite, tomada de preços, concorrência e pregão; ou ser realizadas diretamente, mediante inexigibilidade ou dispensa de licitação (Cox, 2017, p. 2).

Isso também acontece com a manutenção das áreas verdes dos campi universitários: o serviço é licitado por um período determinado, e a empresa terceirizada que ganhar a licitação ficará responsável pela execução do serviço conforme o período determinado na contratação. Seja qual for a modalidade de contratação, existe a necessidade da elaboração de orçamentos (estimativas antecipadas dos custos). Cada um dos serviços desse orçamento contará com item de mão-de-obra, cuja quantificação se baseia em dados históricos de produtividade. Contudo, esses dados de produtividade podem estar equivocados, ou terem sido coletados em áreas com características diferentes, ou em momentos históricos que já não deveriam ser utilizados como referência.

Seja qual for o percurso para a celebração de contrato, com ou sem disputa, o paradigma de contratação engendrado pelas leis citadas exige a elaboração de orçamento estimativo prévio como requisito de legalidade dos negócios jurídicos a serem firmados pela Administração Pública.

Além disso, a qualidade e precisão do orçamento impacta diretamente na economicidade e eficiência das contratações, representando, por isso, uma das etapas mais sensíveis do planejamento dos processos de licitação, de dispensa e de inexigibilidade (Cox, 2017, p. 2).

Tem sido uma realidade recorrente, em anos recentes, a dificuldade das universidades públicas de conseguirem manter seu pleno funcionamento durante todo o ano, em razão de orçamentos que não crescem ano a ano acompanhando a inflação, e se mostram cada vez menores em comparação com as despesas. Nesse cenário, uma ferramenta que permita uma melhor estimativa da força de trabalho necessária para a execução dos serviços de manutenção pode, certamente, contribuir para a realização de licitações de menores valores, resultando em economia para as universidades, aliviando essa situação de escassez de recursos.

Os últimos anos têm sido marcados por uma diminuição do aporte de recursos financeiros às instituições federais de ensino superior pelo Governo Federal, por meio de cortes e contingenciamentos orçamentários, com queda acentuada nos investimentos desde 2015 e reforçada pela aprovação da Emenda Constitucional nº 95/2016.

(...) os últimos dez anos foram os que mais instituíram cortes orçamentários às universidades federais, principalmente no período 2015/2016 e 2018/2022, afetando principalmente as despesas discricionárias, sobre as quais o gestor público teria maior liberdade na tomada de decisão sobre a aplicação do recurso. Considerando a classificação quanto ao grupo de natureza, as despesas de pessoal, custeio e investimentos revelaram quedas em seus montantes medianos, sendo os gastos com investimentos os mais impactados negativamente (Rodrigues *et al.*, 2023, p. 1).

Verificou-se que, principalmente após o ano de 2015, um percentual relevante de universidades federais teve o orçamento anual reduzido, com destaque ao período 2018-2022. Além dos efeitos da crise econômica de 2014, determinou-se que a Emenda Constitucional nº 95/2016 impactou fortemente o orçamento das universidades públicas (Rodrigues *et al.*, 2023, p. 17).

Considerando os *campi* universitários como pequenas cidades, é realidade na maioria, se não em todos, a existência de órgãos análogos a prefeituras, responsáveis pela administração dos *campi*. Esses órgãos possuem o setor de engenharia, responsável pela elaboração de projetos de novos empreendimentos nos *campi* e pela manutenção da estrutura existente. Assim como nas empresas privadas de engenharia, nos setores de engenharia desses órgãos o software de projeto e desenho mais utilizado continua sendo o AutoCAD (Costa; Figueiredo; Ribeiro, 2015).

Esta pesquisa buscou aliar essa realidade de restrição orçamentária, a característica espacialmente distribuída dos serviços de manutenção de áreas verdes, e a utilização

predominante do AutoCAD pelas equipes de engenharia, para propor um *plug-in* para o AutoCAD que aplicasse a meta-heurística da ACO ao dimensionamento das equipes de trabalho que atuam nessa manutenção.

1.2 HIPÓTESE

Diante da natureza espacialmente distribuída das atividades de manutenção de áreas verdes em campi universitários, parte-se da hipótese de que esse problema pode ser formalizado por meio de um modelo matemático capaz de representá-lo como uma instância do Problema de Múltiplos Caixeiros Viajantes (MTSP, do termo em inglês *Multiple Traveling Salesman Problem*).

Admite-se ainda que a aplicação da meta-heurística da ACO a esse modelo, por meio de uma implementação computacional adequada, é capaz de gerar soluções consistentes para o dimensionamento das equipes responsáveis por essas atividades.

Por fim, pressupõe-se que a incorporação dessa abordagem em um *plug-in* desenvolvido para o ambiente do AutoCAD permite a aplicação prática do modelo em contextos reais de planejamento e gestão da manutenção de áreas verdes em campi universitários.

1.3 OBJETIVOS

O objetivo geral desta pesquisa foi o de criar um software que aplique a meta-heurística da ACO ao dimensionamento das equipes de trabalho responsáveis pela manutenção de áreas verdes em *campi* universitários. O software foi concebido como um *plug-in* para o AutoCAD, objetivando facilitar o seu uso por parte das equipes de arquitetura e engenharia das prefeituras dos campi universitários.

Para que esse objetivo geral pudesse ser alcançado, alguns objetivos específicos foram propostos:

- analisar as versões de algoritmos que fazem parte da meta-heurística da ACO e que antecederam o Ant Colony System (incluindo o próprio)¹. Essa análise tem por objetivo detalhar como um problema de otimização necessita ser interpretado para que a ACO possa ser aplicada a ele;
- propor o problema como uma analogia ao MTSP, cujos dados sejam consolidados a partir da adoção de convenções e simplificações dos dados reais do problema;
- analisar os algoritmos pertencentes à meta-heurística da ACO criados para resolver o MTSP, e selecionar o mais adequado para o contexto desta pesquisa;
- desenvolver o *plug-in*;
- aplicar o *plug-in* ao *campus* de Goiabeiras da UFES – etapa essa que ainda integra o desenvolvimento da aplicação, pois engloba validação e ajustes no modelo.

1.4 METODOLOGIA

A metodologia adotada para o desenvolvimento do software que aplica a ACO ao dimensionamento das equipes de jardinagem em campi universitários foi estruturada em cinco etapas principais: (i) revisão bibliográfica; (ii) proposição e modelagem do problema; (iii) seleção e análise de algoritmos; (iv) implementação e testes; e (v) desenvolvimento do *plug-in* para o AutoCAD.

Tais etapas foram encadeadas de forma a permitir uma construção gradual e fundamentada da solução proposta, assegurando coerência entre os fundamentos teóricos, a estruturação do problema, a escolha da abordagem computacional e a entrega final do produto desenvolvido. Essa organização segue o princípio metodológico de que as fases da pesquisa devem ser logicamente articuladas, com objetivos e procedimentos claramente definidos (Lakatos; Marconi, 2003).

¹ A limitação das versões encontra justificativa mais à frente na pesquisa, já que o algoritmo que aplica o ACO ao Problema de Múltiplos Caixeiros Viajantes (MTSP, do termo em inglês *Multiple Traveling Salesman Problem*) adotado se baseia no Ant Colony System.

1.4.1 Revisão Bibliográfica

A revisão bibliográfica teve como objetivo consolidar os fundamentos teóricos necessários à formulação e desenvolvimento da pesquisa. Ela foi organizada em três vertentes principais: (1) entendimento da técnica computacional empregada; (2) entendimento da manutenção de áreas verdes como atividade; (3) e desenvolvimento da aplicação computacional. A primeira vertente se deu com ênfase nos seguintes eixos: (i) a lógica e o funcionamento da ACO; (ii) e as aplicações da ACO no campo da AEC. A segunda vertente se deu com ênfase no levantamento de dados gerais sobre as vegetações consideradas, sobre as atividades realizadas, e sobre os consumos de mão de obra para sua execução. A terceira vertente se deu com ênfase na análise dos requisitos computacionais para o desenvolvimento de aplicações utilizando a API (sigla para Interface de Programação da Aplicação, do termo em inglês *Application Programming Interface*) do AutoCAD e na análise de aplicações semelhantes.

Em relação à primeira vertente, a busca e seleção do material teórico foram orientadas pela afinidade temática com o problema proposto, abrangendo produções científicas nacionais e internacionais. O conteúdo levantado orientou as decisões metodológicas adotadas nas etapas seguintes, como a definição das variáveis do problema, a escolha da estrutura algorítmica e o desenvolvimento da aplicação computacional. A revisão bibliográfica, nesse sentido, assumiu função estruturante ao fornecer os referenciais teóricos que sustentaram o planejamento técnico e a organização da pesquisa (Severino, 2007; Santos, 2018).

Em relação à segunda vertente, a busca e a seleção de materiais se iniciaram por dados que tratassem da manutenção nas vegetações consideradas: gramadas, arbustivas, arbóreas. Durante essa etapa, foi verificado que o nível de exigência e a finalidade eram determinantes, e que dados de manutenção de vegetações com finalidades específicas não poderiam ser utilizados como referências. Foram utilizados livros sobre construção civil, livros sobre elaboração de orçamentos, manuais produzidos por universidades e associações, artigos, além normativas de jardinagem de órgãos públicos e pregões eletrônicos de universidades disponíveis online.

Esgotadas essas fontes, os dados ainda faltantes foram retirados de diferentes versões de Projetos Paisagísticos da UFES (Universidade Federal do Espírito Santo).

Em relação à terceira vertente, a busca e a seleção de materiais se concentraram em publicações específicas sobre o desenvolvimento de *plug-ins* para o AutoCAD, desde os requisitos básicos até a apresentação final.

1.4.1.1 Bases de dados e fontes consultadas

Para garantir abrangência e qualidade científica, foram utilizadas as seguintes bases e bibliotecas digitais: Scopus, Web of Science, ScienceDirect, IEEE Xplore, ASCE Library. A escolha dessas bases se deu por sua relevância acadêmica e pela ampla cobertura de publicações internacionais revisadas por pares nas áreas de arquitetura, engenharia, computação aplicada e construção civil.

O Google Scholar foi utilizado como ferramenta complementar para localizar artigos não indexados ou versões abertas, e o mecanismo de buscas do Google foi utilizado para a busca dos livros, manuais e demais documentos utilizados, sobretudo, no levantamento dos dados relacionados à manutenção de áreas verdes e no entendimento dos requisitos para o desenvolvimento de *plug-ins* para o AutoCAD.

1.4.1.2 Descritores e palavras-chave

Os descritores e palavras-chave a seguir dizem respeito à primeira vertente da revisão bibliográfica. As buscas foram realizadas em português e inglês, considerando plural, singular e abreviações consagradas. Foram utilizadas três categorias principais de termos:

- Termos gerais sobre a técnica: ant colony optimization; ACO; ant system; metaheuristic AND ant colony optimization; metaheurística AND otimização por colônias de formigas.
- Termos relativos à área de aplicação (AEC): architecture, engineering and construction; AEC; arquitetura, engenharia e construção; construção civil; building design optimization; planejamento de obras; project scheduling; construction site layout; facility layout planning.

- Combinações específicas para foco aplicado: "ant colony optimization" AND "construction site layout"; "ant colony optimization" AND "project scheduling"; "ant colony optimization" AND "urban planning"; "ant colony optimization" AND "resource allocation"; "otimização por colônias de formigas" AND "layout de canteiro"; "ACO" AND "routing" AND "construction"; "ACO" AND "spatial optimization"

1.4.1.3 Critérios de seleção e descarte

Os critérios de seleção e descarte a seguir dizem respeito à primeira vertente da revisão bibliográfica. Foram mantidos apenas os artigos revisados por pares que apresentassem aplicação clara da ACO a problemas de planejamento, alocação de recursos, roteamento ou otimização espacial na área AEC. Foram descartados trabalhos sem relação direta com o escopo, incluindo aqueles cujo foco estava exclusivamente no desenvolvimento teórico da ACO ou em aplicações de natureza distinta (como bioinformática ou otimização de redes puramente digitais).

A afinidade temática foi confirmada quando o problema abordado possuía natureza espacial ou logística comparável ao Problema do Caixeiro Viajante (TSP, do termo em inglês *Traveling Salesman Problem*) ou uma de suas variações, pois essa analogia é fortemente associada à essência do algoritmo utilizado na pesquisa.

1.4.1.4 Percurso bibliográfico e justificativa da ênfase do trabalho

A análise dos trabalhos selecionados revelou uma predominância marcante de aplicações voltadas a problemas geograficamente distribuídos, como roteamento de veículos, definição de sequências de tarefas e planejamento de percursos. Essa tendência decorre do próprio fundamento biológico da ACO — inspirado no comportamento das formigas na busca por alimento —, o que a torna especialmente eficiente para problemas de otimização de trajetos e logística espacial. No contexto da AEC, esses desafios frequentemente se materializam como analogias diretas ao TSP, nas quais se busca determinar a sequência ótima de visitas a um conjunto de pontos, minimizando tempo ou distância.

O percurso bibliográfico iniciou-se por artigos recentes (publicados a partir de 2004) que aplicavam a ACO ao planejamento de obras, organização de layouts e alocação de tarefas. A leitura desses trabalhos levou à identificação de referências clássicas que fundamentavam as implementações, apontando para os primeiros estudos formais de Marco Dorigo, que apresentou o algoritmo Ant System em sua tese de doutorado (em 1992) e posteriormente desenvolveu variações amplamente adotadas, como o Elitist Ant System (Dorigo; Maniezzo; Coloni, 1996) e o Ant Colony System (Dorigo; Gambardella, 1997).

A investigação paralela sobre aplicações específicas em logística e alocação de tarefas apontou para o trabalho de Barbosa, Silva Jr. e Kashwabara (2015)²; eles citam Vallivaara (2015), que explorou a otimização do dimensionamento de equipes de manutenção em contextos geograficamente distribuídos.

Assim, o caminho da revisão permitiu uma dupla ancoragem: por um lado, nas bases teóricas e fundadoras da ACO (Dorigo e colaboradores), que garantem o rigor algorítmico; por outro, em aplicações práticas de natureza próxima ao problema desta pesquisa (Vallivaara e outros), que oferecem referência direta para a adaptação metodológica ao caso estudado.

1.4.2 Proposição e Modelagem do Problema

A etapa de modelagem teve como objetivo formalizar, com base em observações de campo e dados reais, a estrutura do problema enfrentado pelas equipes responsáveis pela manutenção paisagística em campi universitários.

O modelo considerou diferentes tipologias de áreas verdes, suas respectivas frequências de manutenção, o tempo necessário para a execução dos serviços e a disponibilidade de trabalhadores. Esses parâmetros foram utilizados para simular cenários e identificar padrões que pudessem ser tratados por meio da abordagem

² Os autores foram alcançados a partir de artigos que tratavam da atribuição de serviços geograficamente distribuídos a múltiplos trabalhadores, nos quais a ACO era adaptada para tratar restrições reais de jornada, deslocamento e frequência de serviços — aspectos centrais também no presente trabalho.

algorítmica adotada. O problema foi descrito como uma instância do MTSP, por sua adequação ao contexto de alocação de equipes a múltiplos destinos com retorno à base.

1.4.3 Seleção e Análise de Algoritmos

A intenção da análise dos algoritmos pertencentes à ACO não foi abranger exaustivamente todas as variações disponíveis, mas sim compreender a formulação original e as principais modificações implementadas em cada novo algoritmo. A análise foi realizada em ordem cronológica, respeitando a sequência histórica das principais versões, desde a formulação original até a versão que serviu de base para o algoritmo implementado nesta pesquisa, especificamente voltado para o MTSP.

1.4.4 Implementação e Testes

A etapa de implementação consistiu na construção do algoritmo em ambiente computacional integrado ao AutoCAD, através da utilização de sua API. A estrutura do código foi organizada com foco em clareza e modularidade, facilitando seu uso futuro em novas pesquisas decorrentes desta.

Foram realizados testes com diferentes cenários baseados em dados reais do campus de Goiabeiras da Universidade Federal do Espírito Santo (UFES), permitindo verificar a estabilidade do algoritmo, a qualidade das soluções geradas e a coerência entre os resultados obtidos e os critérios definidos no modelo. Essa fase foi essencial para a validação da estratégia proposta e para o refinamento do comportamento da aplicação.

1.4.5 Desenvolvimento do plug-in para O AutoCAD

Esta subseção descreve o desenvolvimento do *plug-in* criado para integrar o modelo computacional proposto ao ambiente do AutoCAD, detalhando sua estrutura, funcionamento e etapas de implementação, desde o levantamento dos requisitos técnicos até a geração dos resultados finais. Os dados relacionados ao problema incluem:

- dados geográficos do campus;
- dados das áreas de gramado (localização, limites);
- tipos de grama utilizados;
- localização das árvores;
- distribuição das árvores por faixa de diâmetro médio de tronco;
- frequências das atividades de manutenção de acordo com a estação do ano;
- consumo de mão de obra por unidade dos serviços;
- estação do ano considerada na simulação;
- dados relacionados à mão-de-obra (número e duração dos turnos de trabalho, velocidade de deslocamento, tempo de descanso por turno).

Inicialmente esses dados são utilizados pelo *plug-in* para definir a localização e o custo de permanência dos nós no MTSP proposto como analogia ao problema original. O custo de permanência foi uma grandeza criada para refletir os dados relacionados às atividades realizadas em cada nó do problema. A velocidade de deslocamento é utilizada para calcular os tempos gastos entre os nós (Equação 1):

$$T_{ij} = \frac{d_{ij}}{v} \quad (\text{Eq. 1})$$

onde: T_{ij} = tempo de deslocamento entre os nós i e j ; d_{ij} = distância entre os nós i e j ; v = velocidade de deslocamento dos trabalhadores. O time se inicia sempre com um único trabalhador. Em cada ciclo o dado do tamanho do time é utilizado como número de formigas e uma nova instância do MTSP é resolvida.

A solução do MTSP é a ordem dos nós percorridos por cada formiga do time. Cada uma dessas ordens de nós define o percurso da formiga. O *plug-in*, então, utiliza o dado de velocidade e os custos de permanência dos nós para determinar a duração do percurso de cada formiga (Equação 2):

$$D_k = \sum_{j=1}^m \left(\frac{d_{j,j+1}}{v} + C_j \right) \quad (\text{Eq. 2})$$

onde: D_k = duração total do percurso da formiga k ; $d_{j,j+1}$ = distância entre os nós j e $j+1$ percorridos pela formiga k ; v = velocidade de deslocamento dos trabalhadores; C_j = custo de permanência no nó j ; m = número total de nós percorridos pela formiga k . A maior duração entre os percursos de todas as formigas é, então, comparada com a duração do turno de trabalho (Equação 3).

$$\max(D_k \leq T_t) \quad (\text{Eq. 3})$$

onde: $\max(D_k)$ = maior duração entre os percursos de todas as formigas; T_t = duração do turno de trabalho. Quando a desigualdade não é satisfeita, adiciona-se uma nova formiga ao time e o cálculo é repetido.

O processo então se repete, até que a maior duração entre os percursos das formigas seja menor do que a duração do turno de trabalho. Dentro desse sistema, o algoritmo que implementa a ACO para a solução do MTSP funciona como o motor do *plug-in*, sendo acionado a cada novo ciclo da aplicação

De uma forma geral, entre as atividades para o desenvolvimento do *plug-in*, destacaram-se:

- entendimento e satisfação dos requisitos técnicos para o início do desenvolvimento;
- desenvolvimento da interface do usuário (UI) do *plug-in* dentro do ambiente do AutoCAD;
- implementação do modelo computacional proposto para o problema;
- integração do algoritmo que implementa a ACO ao modelo computacional proposto;
- análise dos resultados gerados;
- ajustes finais e documentação do *plug-in*.

Em relação a esse desenvolvimento, cabe ressaltar que, por se tratar de um software científico, o usuário final é frequentemente o próprio desenvolvedor – pesquisador –; a maioria dos requisitos é emergente, surgindo ao longo do desenvolvimento; o

aparecimento de novas exigências está fortemente associado à avaliação do software, e os testes realizados são muitas vezes informais (KELLY, 2007). Soluções de software científico podem ser caracterizadas como veículos empregados para produzir resultados científicos em diferentes áreas de aplicação; elas constituem um meio – e não um fim –, diferentemente do que ocorre com soluções de software comerciais. Nesse contexto, o foco principal dos desenvolvedores deve estar na execução da ciência, e não no desenvolvimento do código em si (WILSON et al., 2014).

Nesse contexto, o foco da presente pesquisa concentrou-se no desenvolvimento integral do *plug-in*, até sua versão final utilizável (disponibilizada de maneira pública, nas versões código-fonte e executável, no Github). Foram realizadas a avaliação de seu desempenho em diferentes cenários e a análise de sensibilidade dos parâmetros do modelo, conforme sugerido na etapa de qualificação. No entanto, a realização de testes extensivos voltados à usabilidade e à otimização da interface, embora relevante na etapa de finalização comercial de um software, não foi abarcada, por extrapolar o escopo definido para esta pesquisa, cujo objetivo central foi demonstrar a viabilidade científica do modelo e sua implementação em um *plug-in* funcional.

1.5 ESTRUTURA DA TESE

A estrutura desta tese foi organizada de modo a acompanhar o desenvolvimento lógico da pesquisa, articulando os fundamentos teóricos, os procedimentos metodológicos e os resultados obtidos. Buscou-se construir uma narrativa progressiva, em que cada capítulo aprofunda e complementa os aspectos abordados nos anteriores, permitindo ao leitor compreender não apenas o percurso investigativo, mas também as decisões técnicas e conceituais adotadas ao longo do trabalho. A seguir, apresenta-se um breve resumo do conteúdo de cada capítulo.

No capítulo 1, ora apresentado, fez-se a introdução ao tema da pesquisa; justificou-se por que ele é relevante no contexto atual de escassez de recursos para as universidades públicas; abordou-se os objetivos da pesquisa – o geral e os específicos –; e abordou-se a metodologia da pesquisa (como o tema foi dividido para ser abordado em cada uma das suas vertentes, e o método utilizado em cada uma delas).

O capítulo 2 aborda em detalhes a inteligência de enxames e a meta-heurística da ACO. É feita uma análise dos principais eventos, desde o surgimento da inteligência artificial até a proposição do algoritmo que aplica a ACO ao MTSP, situando a ACO e a inteligência de enxames dentro da computação inspirada na natureza.

O capítulo 3 aborda o paisagismo, com foco na manutenção de áreas verdes, fazendo o levantamento na bibliografia consultada das taxas e valores de referência necessários para a construção do modelo matemático que foi implementado computacionalmente.

O capítulo 4 aborda o software criado, detalhando: procedimento metodológico utilizado para a produção do *plug-in* (incluída aí a abordagem adicional dos algoritmos de ACO voltados ao MTSP); limitação do software; simplificações adotadas; abordagem computacional; e as características do software desenvolvido.

O capítulo de Considerações finais aborda os resultados obtidos; retoma as limitações que precisaram ser impostas ao escopo do software para que ele pudesse ser desenvolvido inteiramente no tempo de uma pesquisa de doutorado; apresenta recomendações sobre como o trabalho pode ser continuado em novas pesquisas, incluindo instruções para a utilização do código fonte produzido na pesquisa. Esta seção precede as referências bibliográficas do trabalho, que listam todas as fontes consultadas para o seu desenvolvimento.

2 A INTELIGÊNCIA DE ENXAMES

Alan Turing teve grande importância para a criação dos computadores e para os primórdios da inteligência artificial³. Ele é conhecido por muitos por ter sido o responsável por decifrar a máquina de criptografia nazista na Segunda Guerra Mundial, feito descrito no manuscrito *The Prof's Book: Turing's Treatise on the Enigma*, do final da década de 1940. Contudo, suas contribuições para a ciência vão muito além desse feito. Turing participou dos esforços para a criação dos computadores, e um dos últimos artigos publicados antes de sua morte, em 1954, contribuiu enormemente para o surgimento de toda a área da computação inspirada na natureza. No artigo *The chemical basis of morphogenesis*, publicado em 1952, Turing descreve um modelo matemático através do qual agentes simples, seguindo regras simples, geram resultados incrivelmente complexos. Nas palavras dele: “O propósito deste artigo é discutir um possível mecanismo pelo qual os genes de um zigoto podem determinar a estrutura anatômica do organismo resultante” (Turing, 1952, p. 1).

No início do artigo, Turing refere-se ao processo utilizado para descrever o modelo matemático do crescimento dos embriões como uma simplificação, uma idealização, uma falsificação. Ao final do artigo, ele diz que devido às simplificações feitas dificilmente muitos fenômenos biológicos observados na realidade estariam cobertos pelo artigo, mas que o sistema biológico imaginário que foi proposto por ele e os princípios abordados provavelmente seriam de grande valor na interpretação de formas reais da biologia.

Algumas décadas depois, em 1970, a bióloga Evelyn Fox Keller e o estudante de matemática Lee Segel publicaram o artigo *Initiation of Slime Mold Aggregation Viewed as an Instability*, em que utilizaram os conceitos abordados no artigo de Turing para descrever como o bolor limoso era capaz de se comportar ora como uma criatura única, e ora como um enxame. De acordo com eles, as células de bolor conseguiam dar início ao processo de agregação através do controle da quantidade de acrasina

³ O termo inteligência artificial faz referência a toda a área da computação inspirada na natureza, que inclui: computação evolutiva, inteligência de enxames e sistemas imunológicos artificiais (Tavares, 2018).

(um tipo de feromônio) que eles liberavam individualmente. Um número maior de células seguia por caminhos onde havia maior quantidade de acrasina, criando um processo de *feedback* positivo que acabava por resultar em verdadeiros aglomerados. As células de bolor conseguiam criar e passar a seguir trilhas de acrasina que respondiam a mudanças globais no ambiente.

Esse breve relato mostra o caminho desde o surgimento da computação inspirada na natureza até o surgimento de uma de suas subáreas: a inteligência de enxames. Como Keller e Segel mostraram, já em 1970, os indivíduos num enxame (naquele caso, células de bolor) eram capazes de responder a mudanças globais no ambiente. Em 1992, Marco Dorigo propôs a primeira versão de algoritmo que simulava computacionalmente outro tipo de enxame: as colônias de formigas. Surgia ali uma nova meta-heurística: a ACO, em que o mecanismo de comunicação entre as formigas em sua busca por comida era simulado computacionalmente para resolver problemas de otimização.

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 do autor).

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).

Como vimos no último capítulo, em tais sistemas, o todo é mais do que a soma das partes no sentido pragmático fraco, mas importante, de que, dadas as propriedades das partes e as leis de sua interação, não é uma questão trivial inferir as propriedades do todo (SIMON, 1996, p. 184, tradução do autor).

O grau de complexidade ou simplicidade de uma estrutura depende fundamentalmente da maneira como a descrevemos. A maioria das estruturas complexas encontradas no mundo é extremamente redundante, e podemos usar essa redundância para simplificar sua descrição. Mas para usá-la, para obter a simplificação, precisamos encontrar a representação correta (Simon, 1996, p. 215, tradução do autor).

O último capítulo é também uma revisão de conceitos já tratados anteriormente no livro. No capítulo 3 – *The psychology of thinking: embedding artifice in nature* –, Simon utiliza as formigas como exemplo para a definição do grau de complexidade ao analisar seu comportamento, e relaciona sua complexidade ao 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 do autor).

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.

Os seres humanos, vistos como sistemas de comportamento, são bastante simples. A aparente complexidade de nosso comportamento ao longo do tempo é, em grande parte, um reflexo da complexidade do ambiente em que nos encontramos. Agora, gostaria de arriscar um pouco minhas apostas. Em vez de tentar considerar a “pessoa inteira”, totalmente equipada com glândulas e vísceras, gostaria de limitar a discussão ao Homo sapiens, a “pessoa que pensa”. Eu mesmo acredito que a hipótese se aplica até mesmo à pessoa como um todo mas talvez seja mais prudente dividir as dificuldades desde o início e analisar apenas a cognição em vez do comportamento em geral. (Simon, 1996, p. 53, tradução do autor).

Apesar de ser uma simplificação, de acordo com Kennedy, Eberhart e Shi (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, Eberhart e Shi (2001) pontuam que uma formiga, individualmente, é um ser extremamente inepto. 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 do autor).

Na mesma linha, Kennedy, Eberhart e Shi (2001) citam Christopher 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, Eberhart e Shi (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 expressão inteligência de enxames foi proposta por Gerardo Beni e Jing Wang, no artigo *Swarm Intelligence in Cellular Robotic Systems*, de 1989, com a seguinte definição:

Sistemas de robôs não inteligentes que exibem um comportamento coletivamente inteligente, evidente na

capacidade de produzir, de forma imprevisível (#), padrões ordenados específicos (=não em um sentido estatístico) de matéria no ambiente externo (Beni; Wang, 1993, p. 708, tradução do autor).

Sobre o sentido de imprevisibilidade adotado nessa definição, os autores fazem as seguintes considerações:

Isso pode ser realizado de duas maneiras. Na primeira, o sistema global é, a rigor, previsível, mas não em tempo real. (Aqui, "em tempo real" significa "durante o tempo que o sistema leva para produzir um novo padrão"). Em outras palavras, o problema da previsão é "intratável".

No segundo caso, o sistema global não é previsível (independentemente da velocidade de processamento), pois é essencialmente não-representável externamente (Beni; Wang, 1993, p. 707, tradução do autor).

A expressão foi proposta no contexto de sistemas robóticos celulares, mas passou a representar toda uma área, que engloba uma série de inteligências artificiais, que simulam diferentes coletivos. Nesse contexto, uma definição menos específica, que abranja de maneira direta as inteligências artificiais da área, seria bem-vinda. Em 2001, no livro *Swarm Intelligence*, James Kennedy, Russel Eberhart e Yuhui Shi citaram a que seria a melhor definição do termo, como tendo sido retirada da seção FAQ (sigla em inglês para o termo Perguntas Frequentes) 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; Shi, 2001, p.102, tradução do autor).

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 Prigogine e Nicolis (1977) e Haken (1983), 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 descrevem 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, Eberhart e Shi (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 a nível de colônia, enquanto individualmente as formigas mantém 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, fazem os coletivos de insetos; 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, Eberhart e Shi (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, que 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 (PARA O PROBLEMA DO CAIXEIRO VIAJANTE)

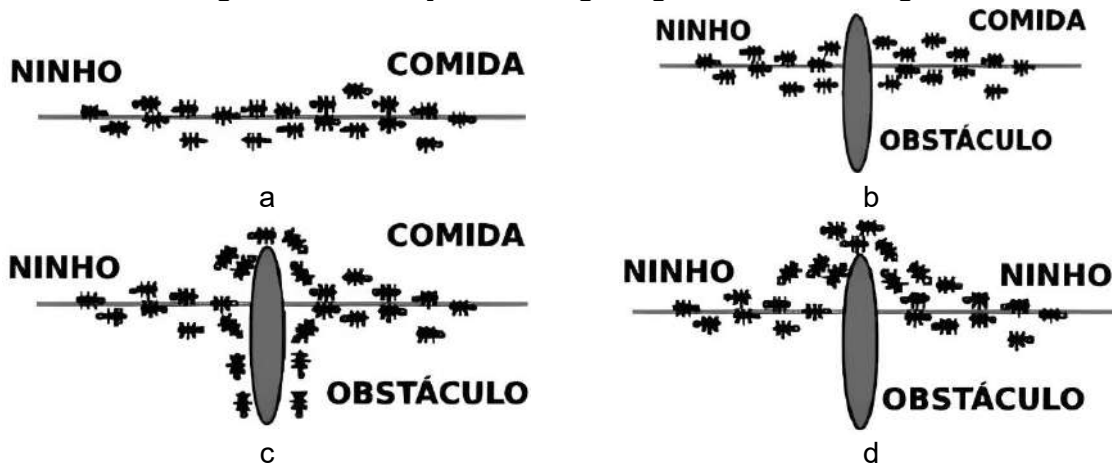
Os primeiros passos em sentido ao desenvolvimento de aplicações baseadas em formigas artificiais criadas computacionalmente foram dados por biólogos, que estudaram o comportamento de insetos e conseguiram descrever como esses insetos se comunicavam. Pierre-Paul Grassé (1959), zoologista, introduziu o conceito de estigmergia, que consistia numa forma de comunicação indireta entre insetos através 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 ocorre por meio de feromônios, substâncias químicas depositadas no chão pelo percurso entre a colônia e a fonte de comida, como demonstrado por Deneubourg, Pasteels e Verhaeghe (1983), 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). A Figura 1 mostra como as formigas, inicialmente seguindo em linha reta (situação a), conseguem rapidamente se adaptar a uma mudança no ambiente – interposição de um obstáculo (situação b) – através da seleção probabilística de rotas (situação c), determinando o melhor percurso considerando a existência do obstáculo (situação d).

Figura 1 - Ilustração da estigmergia entre as formigas



Fonte: adaptado de Kavenh; Talatahari (2010, p. 159)

Essa característica inspirou a criação de uma nova meta-heurística, chamada de ACO, 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 ACO 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 TSP, 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 TSP (Dorigo; Stützle, 2009), o que leva a um fato: a principal contribuição dessa primeira versão foi a criação de uma nova meta-heurística (da ACO). Nos anos seguintes, diversas versões foram propostas baseadas nessa

primeira versão. O Quadro 1 mostra as versões de algoritmos de Otimização por Colônias de Formigas que foram criados entre 1991 e 2004.

Quadro 1 – Principais tipos de ACO 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 Cordon, Inaki Fernandez de Viana e Francisco Herrera	2000
Population-based ACO	Michael Guntsch e Martin Middendorf	2002
Beam-ACO	Christian Blum	2004

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

Em 1999, Dorigo e Di Caro definiram a meta-heurística da ACO como: uma estrutura básica para os algoritmos que implementavam heurísticas baseadas na simulação computacional das colônias de formigas. Portanto, a partir de então, ficou estabelecido que o termo ACO 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 vértices já foram visitados por elas; constroem seu caminho (solução) através 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 maior complexidade dos problemas resolvidos pelos algoritmos em relação aos problemas resolvidos por formigas reais) é necessário haver um correto balanço entre 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 1, a seguir são detalhadas as formulações matemáticas das versões que antecederam o Ant Colony System, incluindo o próprio. O Team Ant Colony Optimization (TACO), algoritmo utilizado de maneira extensiva ao longo do trabalho, aplica o Ant Colony System ao MTSP, sendo esse o motivo deste recorte incluir apenas esses quatro algoritmos.

2.1.1.1 Ant System

O Ant System foi o algoritmo inaugural da meta-heurística da ACO. Matematicamente, ele é descrito por: construção de trajetos e atualização das trilhas de feromônios. A construção de trajetos envolve a regra de transição probabilística, através da qual cada formiga decidirá para qual nó irá na próxima iteração do algoritmo. A regra de transição probabilística do Ant System é dada pela Equação 4:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta}, \text{ se } j \in N_i^k \quad (\text{Eq. 4})$$

onde: p_{ij}^k é a probabilidade de que a formiga k localizada no nó i se mova para o nó j ; α e β são os parâmetros que determinam as influências relativas da intensidade da

trilha de feromônios e do valor heurístico, respectivamente; τ_{ij} é a intensidade da trilha de feromônios entre os nós i e j ; η_{ij} é o valor heurístico entre os nós i e j , e é igual ao inverso da distância entre os nós; N_i^k é a potencial vizinhança para a formiga k localizada no nó i (composta pelos nós não visitados), sendo i cada nó dessa potencial vizinhança. Cada formiga k mantém uma memória M^k que contém os nós já visitados por ela; essa memória é utilizada para definir a potencial vizinhança da formiga k a cada iteração, já que cidades já visitadas não podem ser visitadas novamente.

Depois que todas as formigas construíram seus trajetos, as intensidades das trilhas de feromônios são atualizadas. A taxa de evaporação de feromônios é aplicada em todas as arestas⁴ (de acordo com a Equação 5), reduzindo a intensidade das trilhas. Posteriormente, todas as formigas depositam feromônios nas arestas que foram percorridas por elas, de acordo com a Equação 6.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij}, \forall (i, j) \in L \quad (\text{Eq. 5})$$

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k, \forall (i, j) \in L \quad (\text{Eq. 6})$$

ρ é a taxa de evaporação de feromônios, e L é o conjunto de arestas que conectam os nós. $\Delta \tau_{ij}^k$ é a quantidade de feromônios que a formiga k deposita nos arcos visitados por ela, e vale $\Delta \tau_{ij}^k = \frac{1}{C_k}$, onde C_k é o comprimento total do percurso da formiga.

2.1.1.2 Elitist Ant System

O Elitist Ant System foi uma evolução do Ant System, que permitiu que a melhor formiga da colônia até então, a cada iteração, adicionasse uma camada a mais de feromônios. A Equação 7 define a atualização das intensidades das trilhas de feromônios:

⁴ As arestas são as linhas que ligam os nós entre si.

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k + e \cdot \Delta \tau_{ij}^{MA} = \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k + e \cdot \frac{1}{C^{MA}} \quad (\text{Eq. 7})$$

onde: e é um parâmetro que define o peso dado à melhor formiga até o momento, e C^{MA} é o comprimento do trajeto T^{MA} construído pela melhor formiga até então. O termo MA faz referência a *melhor até agora*.

2.1.1.3 Ant Colony System

No Ant Colony System, o balanço entre a exploração de áreas já encontradas e a prospecção de novas áreas é feito pelo parâmetro q_0 , que controla se a escolha será feita pela melhor opção conhecida ou de forma probabilística. Sendo q um valor randômico uniformemente distribuído contido em $[0, 1]$, se $q \leq q_0$, j será dado pela Equação 8 (exploração), caso contrário j será dado pela Equação 4, com $\alpha = 1$ (prospecção).

$$j = \operatorname{argmax}_{l \in N_i^k} \left\{ \tau_{il} [\eta_{il}]^\beta \right\} = \operatorname{argmax}_{l \in N_i^k} \left\{ \tau_{il} \left[\frac{1}{d_{il}} \right]^\beta \right\} \quad (\text{Eq. 8})$$

A cada vértice selecionado, cada formiga corrige a concentração de feromônios do último arco percorrido de acordo com a Equação 9, onde φ é o coeficiente de decaimento ($0 < \varphi < 1$) e τ_0 é a concentração inicial de feromônios.

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (\text{Eq. 9})$$

Ao final de cada iteração, a melhor formiga corrige a concentração de feromônios dos arcos que fazem parte de seu trajeto, de acordo com a Equação 10.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \frac{1}{C^{MA}} \quad (\text{Eq. 10})$$

2.1.1.4 Ant-Q

O Ant Colony System foi baseado no Ant-Q, sendo a única diferença entre os dois a definição do termo τ_0 , que no Ant-Q é dado pela Equação 11 (Dorigo; Stützle, 2004). De acordo com Dorigo e Stützle (2004), como percebeu-se que a atribuição de constantes de pequeno valor para τ_0 resultava num algoritmo mais simples e com desempenho semelhante, o Ant-Q foi abandonado.

$$\tau_0 = \gamma \max_{j \in N_i^k} \{ \tau_{ij} \} \quad (\text{Eq. 11})$$

2.2 OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS (PARA O PROBLEMA DE MÚLTIPLOS CAIXEIROS VIAJANTES)

Todos os algoritmos que implementam a meta-heurística da ACO citados até aqui foram inspirados no TSP 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 TSP 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 TSP consiste no caso do MTSP em que $m = 1$. Apesar do MTSP com múltiplas origens também ter sido estudado (Ghafurian; Javadian, 2011; Ramadhani; Hertono; Handari, 2017; Farisi; Setiyono; Danandjojo, 2021), todas as menções ao MTSP referem-se ao MTSP com uma única origem.

Apesar do MTSP 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 (Costa *et al.*, 2012). No caso da meta-heurística da ACO, diferentes algoritmos ou técnicas foram propostas, como por exemplo: Junjie e Dingwei (2006), Vallivaara (2008, *apud* Barbosa; Silva Jr.; Kashiwabara, 2015); Ghafurian e Javadian (2011); Costa *et al.* (2012); Yousefikhoshbakht, Didehvar e Rahmati (2013); Hingrajiya, Gupta

e Chandel (2015); Ramadhani; Hertono e Handari (2017); Srikakulapu e Vinatha (2018); Lu e Yue (2019); Wang *et al.* (2020); Sun *et al.* (2021); Changdar *et al.* (2023); Pereira, Pires e Oliveira (2023); Sun *et al.* (2023); e Xu *et al.* (2023).

Lupoaie *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 Lupoaie *et al.* (2019), na solução do Problema de Múltiplos Caixeiros Viajantes, quase tão importante quando minimizar a distância total é balancear o tamanho dos percursos individuais de cada um dos times. A técnica proposta por Valliavaara endereçou essa questão de maneira eficiente, o que justifica a opção por ela num universo com tantas outras opções.

Valliavaara (2008, *apud* Barbosa; Silva Jr.; Kashiwabara, 2015) adaptou o Ant Colony System para o MTSP. A cada iteração, a formiga de menor rota parcial seleciona seu próximo nó de acordo com a regra de seleção probabilística do Ant Colony System. 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.

2.3 APLICAÇÕES DA OTIMIZAÇÃO POR COLÔNIAS DE FORMIGAS NOS CAMPOS DE AEC

A ACO já foi utilizada para a determinação dos parâmetros ótimos (pesos, número de nós) de redes neurais (Ashena; Moghadasi, 2011), para a estimativa da quantidade de resíduos de construção (Lee; Kim; Kim, 2016). Também foi utilizado para a solução de problemas de otimização combinatória (Dorigo; Gambardella, 1997) e problemas de definição de layout (Solimanpur *et al.*, 2005) e de *flowshop scheduling*⁵ (Rajendran;

⁵ *Flowshop scheduling problems* são problemas que envolvem a realização de N trabalhos em M máquinas; o tempo de realização de cada trabalho é conhecido. Todos os trabalhos têm que ser feitos sequencialmente nas máquinas e na mesma ordem. O objetivo é determinar a ordem ideal para otimizar algum parâmetro, normalmente o tempo.

Ziegler, 2004). Lee (2011) utilizou o ACO para otimizar o cronograma das obras de uma casa de veraneio de grandes dimensões, num processo que envolvia simultaneamente construção e operação; ele utilizou o ACO para a determinação do menor montante possível a ser adquirido pela empreiteira na forma de empréstimo com bancos para a realização de uma obra, considerando os custos de cada uma das etapas, e as receitas geradas por elas depois de prontas, ainda durante o período da construção.

Outros estudos utilizaram o ACO para a otimização de projetos de estruturas metálicas espaciais (Rajasekaran; Chitra, 2009 *apud* Lee, 2011), traçados de tubulações (Christodoulou; Ellinas, 2010), traçado de terrenos (Ning; Lam; Lam, 2010), problemas de otimização tempo-custo (Ng; Zhang, 2008; Xiong; Kuang, 2008), distribuição de força de trabalho (Lee *et al.*, 2010) e otimização de cronogramas de obras (Christodoulou, 2009; Lee, 2009; Christodoulou, 2010).

O ACO foi utilizado também para a otimização de sistemas de distribuição de água (Maier *et al.*, 2003), otimização de projetos de canais abertos, otimização de propriedades hidráulicas do solo (Abbaspour; Schulin; Genuchten, 2001), otimização de sistemas de água subterrânea (Li; Hilton, 2005), e otimização de problemas estruturais (Kaveh; Shojaee, 2007; Kaveh; Shahrouzi, 2008; Kaveh *et al.*, 2008; Kaveh; Talatahari, 2008; Kaveh; Jahanshahi, 2008).

3 PAISAGISMO E MANUTENÇÃO DE ÁREAS VERDES EM CAMPUS UNIVERSITÁRIOS

Esta seção discute os conceitos de paisagismo adotados nesta tese, o enquadramento das áreas verdes no contexto urbano e a formulação do dimensionamento das equipes de manutenção como um problema de otimização.

3.1 PAISAGISMO E ÁREAS VERDES URBANAS

Araújo, Onofre e Macedo (2016) definem a arborização urbana como toda cobertura vegetal de porte arbóreo existente dentro da cidade. Paiva, Luz e Silva (2022) utilizam a terminologia “áreas verdes urbanas” para o mesmo conceito. Para eles, as áreas verdes urbanas são um subgrupo da categoria “espaço livre de edificações”, que apresentam cobertura vegetal, arbórea, arbustiva ou rasteira, distribuídos em: canteiros centrais; praças; parques; florestas e áreas de conservação urbanas; jardins; e terrenos não edificadas. Essa definição também se aproxima da realidade dos campi universitários brasileiros, onde o espaço verde é composto por áreas arborizadas em conjunto com grandes áreas de gramado.

Dentro das universidades, a notória carência de setores específicos e políticas voltadas à promoção da preservação e da qualidade das áreas verdes faz com que seus benefícios à população de um modo geral, e principalmente à população que frequenta seus campi, fiquem muito aquém do que poderiam, sendo vista muitas vezes como um elemento puramente estético. Segundo Araújo *et al.* (2015), essas áreas podem desempenhar diversas funções, de acordo com sua finalidade e o grau de conservação. Esses dois fatores definirão o nível de investimentos necessários para a manutenção de tais áreas.

É certo que as áreas verdes dentro de campi universitários exercem importantes funções ambientais, que vão muito além do aspecto paisagístico e embelezamento estético (Araújo; Onofre; Macedo, 2016). Se a criação dessas áreas for feita de maneira estratégica e bem planejada, diversos benefícios podem ser alcançados (Paiva; Luz; Silva, 2022). Esses benefícios incluem: conforto térmico (melhora do microclima local); absorção de parte dos raios solares e da poluição atmosférica;

aumento do sombreamento; diminuição da poluição sonora (mitigação de ruídos) e visual; manutenção da biodiversidade em ambientes urbanos, favorecendo o equilíbrio ecológico no espaço, proporcionando abrigo e recursos à fauna local; manutenção da umidade do solo (e recarga de aquíferos), favorecendo uma melhor infiltração, uma menor evapotranspiração da água precipitada, e evitando inundações; estabilização do solo, impedindo erosões; organização dos espaços para atividades humanas, de recreação e esportivas; melhoria da qualidade de vida (bem-estar psicológico) dos frequentadores (Araújo; Onofre; Macedo, 2016).

3.1.1 ÁREAS VERDES NO CONTEXTO DE CAMPI UNIVERSITÁRIOS

De acordo com Macêdo, Lisboa e Carvalho (2012), um campus universitário, pelo grande espaço geográfico que ocupa, a diversidade de serviços e infraestruturas que comporta, e pela elevada população, se assemelha estrutural e administrativamente a uma pequena cidade, inclusive quando se trata de questões ambientais.

Segundo eles, a seleção de espécies para a arborização dos *campi* envolve informações relacionadas à adaptabilidade dessas espécies ao ambiente, como facilidade de cultivo, velocidade de crescimento, suscetibilidade a pragas e doenças, conformação de copas e peso dos frutos. O formato das copas irá determinar o sombreamento proporcionado por cada árvore quando adulta; isso pode ser usado para definir o espaçamento ideal entre as árvores próximo a uma passagem de pedestres.

Rodrigues, Carvalho e Mori (2014) citam o caso do campus de São Carlos da Universidade de São Paulo, onde foi proposta a criação de áreas verdes a no máximo 100 metros de qualquer edificação do campus, viabilizando o fácil acesso dos estudantes a essas áreas. Além disso, foi incentivada a criação de trilhas, corredores verdes e calçamentos cobertos por árvores, a fim de incentivar a utilização de ciclovias e passeios de pedestres para o deslocamento no campus. Dessa forma, a existência dessas áreas verdes passou a ser vista como um ativo da população universitária e da comunidade do entorno.

Araújo, Onofre e Macedo (2016) manifestam que, idealmente, a arborização do campus deve integrar todos os espaços externos às edificações, incluindo as ruas, os passeios, as ciclovias, as praças, os jardins e as zonas especiais. Macêdo, Lisboa e Carvalho (2012) pontuam que, diferentemente de um bairro numa cidade, que normalmente é subdividido em lotes, num campus universitário não existem esses limites, todos os espaços são transitáveis por pedestres, e assim todo o espaço existente entre as construções é susceptível à gestão de áreas verdes.

3.1.2 GESTÃO DE ÁREAS VERDES

A gestão das áreas verdes dentro de um campus universitário se assemelha muito à gestão das áreas verdes dentro de um bairro, ou de um município. Da mesma forma que um município necessita de um Plano Diretor Urbano para orientar a construção de imóveis dentro de seus limites, faz-se necessária a existência de um plano que defina como as florestas serão tratadas; o mesmo se aplica às universidades. Um plano de gestão de áreas verdes apresenta orientações e diretrizes para a preservação da cobertura vegetal do campus, direcionadas ao plantio e à manutenção das áreas verdes, com foco na manutenção do conforto, da estética e da biodiversidade (Araújo; Onofre; Macedo, 2016).

O levantamento da arborização de um campus universitário proporciona o conhecimento do patrimônio botânico do campus, embasando a implantação de ações corretivas que permitam ampliar a conservação biológica na universidade. A vegetação pode ser classificada quanto à origem, porte arbóreo, condições das árvores, injúrias/danos mecânicos, problemas ocasionados por raízes, geração de sombras etc. A análise qualitativa das árvores permitirá avaliar a vegetação com relação a aspectos florísticos, fitossanitários, de plantio e manejo (Souza *et al.*, 2019).

As áreas verdes de um campus universitário são compostas por toda a cobertura vegetal do campus e normalmente estão divididas em: canteiros centrais, bosques, parques, praças, jardins, áreas de proteção ambiental, áreas não edificantes e áreas abertas de gramado.

3.1.3 MANUTENÇÃO DE ÁREAS VERDES

De acordo com a norma ABNT (Associação Brasileira de Normas Técnicas) NBR 5462:1994 (Confiabilidade e Manutenibilidade), a manutenção pode ser de três tipos: preventiva, corretiva e preditiva (controlada). A manutenção preventiva é a efetuada em intervalos predeterminados ou com base em algum outro critério previamente definido, com o objetivo de reduzir a ocorrência de falhas ou a degradação prematura de um bem. A manutenção corretiva é efetuada após a ocorrência de eventos atípicos que requeiram medidas corretivas a fim de interromper os efeitos do evento e recolocar o bem em condições de exercer novamente sua função original. A manutenção preditiva (controlada) é um tipo de manutenção voltada à garantia da qualidade desejada para o bem, com o objetivo de reduzir ao mínimo as necessidades tanto de manutenção preventiva quanto de manutenção corretiva.

A manutenção de áreas verdes é simplesmente o conceito de manutenção trazido pela norma NBR 5462 aplicado às áreas verdes. Dessa forma, as ações planejadas de manutenção de áreas verdes num campus universitário podem ser sintetizadas como o conjunto de ações programadas executadas nas áreas verdes localizadas no interior do campus, com o objetivo de “assegurar plena capacidade e condições de funcionamento contínuo, seguro e confiável dos ambientes de trabalho administrativo ou acadêmico” (UNIVERSIDADE DE SÃO PAULO, 2018, p. 4).

As ações voltadas à gestão das áreas verdes nas universidades são recentes, quando comparadas à existência das próprias universidades. Na USP (Universidade de São Paulo), por exemplo, fundada em 1934, as primeiras iniciativas voltadas à promoção do meio ambiente datam da década de 1990, com a criação do Programa USP Recicla, em 1994, e dos Programas Permanentes para o Uso Eficiente dos Recursos Hídricos e Energéticos, em 1997 (energia) e 1998 (água). Em 2009 foi criado um grupo de trabalho para discutir a implantação de uma gestão ambiental na universidade, que culminou, em 2012, na criação da Superintendência de Gestão Ambiental (SGA).

Ou seja, na maior parte de sua história, a USP foi mantida sem um setor específico para cuidar da preservação dos ativos ambientais da universidade. A partir do ano de 2012, quando a SGA foi criada, iniciativas começaram a ser tomadas no sentido de

identificar áreas de mata atlântica e de cerrado, e de criar áreas de reservas ambientais e áreas de proteção permanente. A distinção dessas áreas é importante porque definirá as ações que serão consideradas ou não para cada área. No caso de áreas com mata nativa, as ações se limitam a monitoramento e preservação. Essas áreas são mantidas, dentro do possível, em seu estado natural, e intervenções diretas – como poda e plantio de novas árvores – são ao máximo evitadas. Para essas áreas, “é necessária (...) sua identificação e delimitação, (...) considerando, além da conservação das áreas existentes, sua recuperação, restauração e renaturalização” (Lemos *et al.*, 2018, p. 94).

No caso da Universidade de Brasília, a manutenção de áreas verdes está a cargo da Coordenadoria de Parques e Jardins, e inclui as seguintes atividades: poda de gramado, poda de árvores, rastelagem, roçagem, recolhimento de resíduos e aplicação de inseticidas (UNIVERSIDADE DE BRASÍLIA, 2017). Já a Universidade Federal Fluminense (2018) incluiu uma lista maior de serviços, num pregão realizado em 2018: roçada mecânica e/ou manual; manutenção de gramados; manutenção de jardins; podas de árvores e arbustos; limpeza e manutenção de áreas pavimentadas sem vegetação; limpeza e manutenção de áreas não pavimentadas sem vegetação; limpeza e manutenção de áreas não pavimentadas com vegetação; limpeza de canaletas, bocas de lobo, bueiros e ralos, de calhas de drenagem e de caixas coletoras de águas pluviais; limpeza de coberturas, telhados e calhas; remoção de entulhos; plantio de espécies de vegetação (árvores, arbustos e ervas); compostagem de resíduos vegetais; e produção de mudas.

De acordo com a Universidade Federal Fluminense (2018), a manutenção das áreas verdes:

(...) visa executar um conjunto de atividades, para assegurar plena capacidade e condições de funcionamento contínuo, seguro e confiável dos ambientes de trabalho administrativo, acadêmico e de suas instalações, preservando as características e desempenhos (Universidade Federal Fluminense, 2018, p. 6).

De acordo com Araújo, Onofre e Macedo (2016), as ações de poda visam a adequação da arborização à realidade do campus, minimizando possíveis

interferências com a infraestrutura, as edificações e o mobiliário, com o trânsito de veículos e de pedestres, a acessibilidade e o bem-estar da comunidade. Eles indicam que o monitoramento contínuo das ações de manutenção é necessário para aumentar o sucesso das ações.

3.2 OTIMIZAÇÃO DO PLANEJAMENTO DA MANUTENÇÃO DE ÁREAS VERDES (DIMENSIONAMENTO DE EQUIPE)

Otimizar o dimensionamento de equipes de manutenção de áreas verdes significa determinar qual o número mínimo de profissionais necessários para executar todos os serviços a serem executados em determinado período. Isso significa organizar os tempos de deslocamento até os locais das atividades e os tempos de realização das atividades propriamente ditas dentro dos turnos que compõem a jornada de trabalho de cada trabalhador, alocando um novo trabalhador quando as jornadas dos trabalhadores já alocados estiverem preenchidas.

Para o cálculo dos tempos de realização das atividades, as composições unitárias dos serviços de manutenção são fundamentais. A composição descreve todos os custos associados a uma unidade do serviço considerado, e se apresenta na forma de uma tabela com cinco colunas: insumo, unidade, índice, custo unitário e custo total (Mattos, 2006). Na definição do índice de mão-de-obra por unidade de serviço entra o conceito de produtividade, que pode variar bastante em função de fatores diversos, como tamanho da obra e instabilidade do tempo, por exemplo. Esse índice é muitas vezes de difícil obtenção, mas é o valor mais importante da composição quando se trata do dimensionamento de equipes.

3.2.1 Elenco das atividades (serviços)

A manutenção das áreas verdes será, em última análise, o conjunto de uma série de atividades (serviços), executadas com periodicidades específicas. Os serviços que fazem parte da manutenção podem ser retirados de cadernos de encargos, de manuais de paisagismo, ou mesmo de projetos de paisagismo específicos.

O elenco das atividades não é algo simples e direto. As atividades envolvendo paisagismo e jardinagem recebem pouco destaque nas publicações das áreas de arquitetura, engenharia e construção. Por exemplo, o livro *A Técnica de Edificar*, da editora PINI, em sua 10ª edição, de 2009, traz apenas um capítulo, de uma única página, sobre o assunto. Conseqüentemente, a consulta de projetos específicos torna-se bastante importante para a descrição das atividades envolvidas na manutenção. Aproveitando a relação direta do trabalho com a UFES, pela seleção de seu campus como estudo de caso, diferentes versões (de 1997 e de 2010) de seus projetos paisagísticos foram utilizados como referências para preencher essas lacunas causadas pela falta de dados.

O projeto paisagístico de 1997 faz o seccionamento do campus em áreas específicas, e apresenta como objetivo a compatibilização das áreas de interesse paisagístico com os demais sistemas existentes no campus, procurando oferecer maior nível de conforto à comunidade como um todo.

Em relação ao corte de árvores, o projeto faz a seguinte consideração: “O corte de árvores deverá ser efetuado por pessoas conhecedoras do assunto, utilizando instrumental apropriado (motosserra, escada, esporão etc.)” (Fernandes, 1997, p. 10). Recomenda também como se deve proceder após o corte:

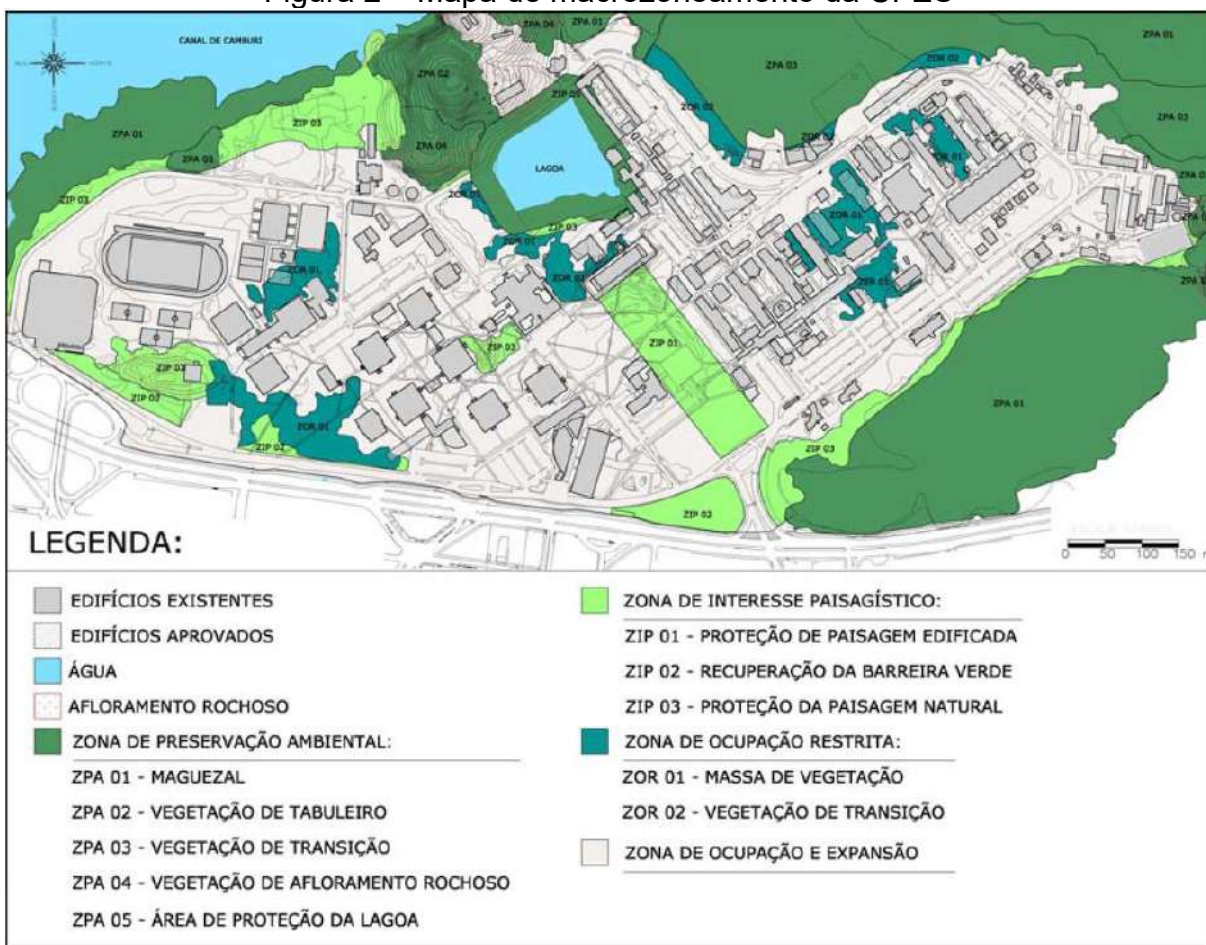
Após o corte o mais rente possível do solo e picagem do material lenhoso, este deverá ser retirado da área do campus universitário, através do uso de veículo de dois eixos, possuindo carroceria longa e baixa (Fernades, 1997, p. 10).

Em relação ao plantio de grama, o projeto especifica qual o tipo de grama utilizado: “Wild zoysia (grama esmeralda) em tapetes de 1,25 x 0,40m” (Fernades, 1997, p. 10); ele também faz recomendações em relação à manutenção: “A empresa executora deverá dar manutenção às plantas (limpeza, controle de formigas e irrigação) durante 30 (trinta) dias contados a partir do término do plantio” (Fernades, 1997, p. 16). E à irrigação: “A irrigação também deverá ser realizada durante o período de plantio, sempre que necessário” (Fernades, 1997, p. 16).

Em relação à mão de obra, o projeto lista as atividades envolvidas: aterro; corte de árvores e picagem; marcação/abertura de covas; plantio/replanteio; reconstituição de áreas ajardinadas; manutenção (irrigação, limpeza e controle de formigas); recolhimento de material lenhoso/residual (Fernades, 1997).

O Projeto Paisagístico de 2010 traz o mapa de macrozoneamento do campus, fazendo referência ao Plano Diretor Físico da UFES de 2008 (Figura 2).

Figura 2 – Mapa de macrozoneamento da UFES



Fonte: Universidade Federal do Espírito Santo (2008, p. 53)

A partir da análise dessas diferentes versões de Projetos Paisagísticos, foi possível obter dados a respeito das espécies de grama utilizadas no campus, e delimitar quais áreas serão consideradas no planejamento de manutenção, através das especificações das áreas de interesse paisagístico e de interesse ambiental.

A fim de quantificar os serviços envolvidos na manutenção das áreas verdes, as atividades foram divididas em vegetações gramíneas e vegetações arbóreas.

3.2.1.1 Levantamento das frequências das atividades de manutenção em gramados

Os serviços de manutenção em gramados foram divididos em: poda; irrigação; e remoção de espécies invasoras.

3.2.1.1.1 Poda

A frequência com que o gramado necessita ser podado depende de alguns fatores, sendo os principais: tipo de grama, época do ano (temperatura e regime de chuvas), frequência de irrigação, adubação (Raats, 2005; Nagel; Byrd Jr., 2015), tipo e fertilidade do solo (Associação Nacional Grama Legal, 2017b). A frequência de podas dependerá da velocidade de crescimento do gramado, e fatores como excesso de fertilidade do solo, calor e abundância de água aumentarão essa velocidade.

Em relação à época do ano, existem quatro cenários (um para cada estação do ano): verão, outono, inverno e primavera. No verão, as chuvas constantes estimulam o crescimento do gramado, aumentando a necessidade e a frequência de podas. O outono é um período de transição, em que o gramado começa a se adaptar para o inverno, e já não cresce tanto. No inverno, o gramado entra em dormência vegetativa, e não exige muita manutenção. Já a primavera favorece o crescimento das gramas, de forma semelhante ao verão (CGP, 2022).

Gonçalves (2020) faz uma estratificação diferente, fazendo referência a apenas duas condições: primavera-verão e outono-inverno. Na primeira, de acordo com ele, as gramas crescem em maior velocidade, devendo ser podadas num intervalo que pode variar entre 7 e 15 dias. Na segunda as gramas crescem menos, podendo seu intervalo entre podas variar entre 20 e 30 dias.

Após a formação de uma nova área de gramado, a primeira poda deve ser realizada de 30 a 60 dias após o plantio. Depois disso, a frequência será definida pela altura

máxima definida para a grama (Associação Nacional Grama Legal, 2017b). A Normativa de Jardinagem para o Poder Judiciário De Santa Catarina (Poder Judiciário de Santa Catarina, 2020) recomenda uma periodicidade de uma poda a cada 30 dias nos gramados.

A Tabela 1 traz recomendações de altura e frequência de corte ideais de algumas espécies de grama, estratificando os dados para verão e inverno. A Tabela 3 apresenta dados de frequências de cortes, estratificados em épocas quentes e épocas frias. Uma estratificação simplificada que pode ser utilizada em ambos os casos é: (1) verão e primavera; e (2) inverno e outono. As Tabelas 2 e 4 apresentam recomendações de alturas de corte por espécie de grama.

Tabela 1 – Altura e frequência de corte de diferentes espécies de grama

Características gerais		Esmeralda e Mini Zoysia	São Carlos	Bermuda Tifway
Altura de corte		2,5 a 5 cm	2,5 a 5 cm	1,5 a 3 cm
Frequência de corte	Verão	2 vezes por semana	1 vez por semana	2 vezes por semana
	Inverno	1 vez por semana	2 vezes por mês	2 vezes por semana

Fonte: Becker (2012, p. 21)

Tabela 2 – Alturas de corte recomendadas

Espécie da grama	Polegadas
Bermudas	0,5 a 1,5
Santo Agostinho	2,5 a 3
Centipede	1,5 a 2
Grama esmeralda	1 a 1,5
Festuca arundinacea	2 a 3 (no inverno); 3 a 4 (no verão)
Poa pratensis	1,5 a 2 (no inverno); 2 a 3 (no verão)
São Carlos	1 a 2
Festuca Rubra	2 a 2,5 (no verão)

Fonte: Nagel; Byrd Jr. (2015, p. 19, tradução do autor)

Tabela 3 – Altura ideal e frequência de corte

Grama (espécie/variedade)	Altura de corte ideal (cm)	Frequência de corte	
		Épocas quentes	Épocas frias
Esmeralda	2 a 4	1 a 2 vezes/semana	2 a 4 vezes/mês
São Carlos	3 a 5	1 a 3 vezes/semana	2 a 6 vezes/mês
Batatais	3 a 6	1 a 2 vezes/semana	2 a 4 vezes/mês
Santo Agostinho	5 a 6	1 a 2 vezes/semana	4 a 6 vezes/mês
Bermudas (Tifway e Celebration)	1,5 a 3	4 a 5 vezes/semana	2 a 3 vezes/semana

Fonte: adaptado de Associação Nacional Grama Legal (2023, p. 9)

Tabela 4 – Faixas ótimas de altura de corte

Espécie da grama	Faixa ótima (cm)
Bluegrass do Kentucky	5,1 a 8,9
Azevém perene	5,1 a 8,9
Festuca de folhas finas	5,1 a 8,9
Festuca alta	6,4 a 10,2
Esmeralda	2,5 a 5,1

Fonte: adaptado de Reicher *et al.* (2006)

De acordo com a Associação Nacional Grama Legal (2017b), a primeira poda deve ser feita entre 30 e 60 dias após o plantio, e tem o objetivo de estimular a formação completa do gramado. Já de acordo com Raats (2005), várias podas são necessárias durante a formação do gramado, com a finalidade de induzir a emissão de novas ramificações e acelerar o fechamento do gramado. Após esse período de formação do gramado, a frequência das podas passa a ser determinada pela altura máxima definida para o gramado.

Ainda de acordo com Raats (2005), um gramado aparado corretamente torna-se muito mais resistente a pragas, doenças e plantas invasoras. Quando a grama é aparada corretamente, e na altura adequada, as raízes crescem mais profundamente, dando ao gramado um aspecto saudável (Salvador; Minami, 2002). Becker (2012) indica a poda do gramado como a operação de manutenção mais importante, sendo capaz de proporcionar uniformidade, densidade e crescimento horizontal.

A variação na frequência das podas com as estações do ano está ligada à influência do clima e da temperatura sobre seu crescimento. De forma equivalente, o clima e a temperatura afetam a frequência com que a grama necessita ser irrigada. Segundo Lorenzi e Souza (1999, *apud* Tapparo, 2008), a grama perde menos água durante o inverno; por esse motivo, a frequência de regas deve diminuir. Além da menor

necessidade, o excesso pode favorecer a ocorrência de doenças. Outra recomendação frequente é de que a irrigação de áreas gramadas seja sempre feita pela manhã (Gonçalves, 2020; Raats, 2005; Associação Nacional Grama Legal, 2023).

Para a estimativa do tempo de execução da atividade de poda de gramados, foi considerada a composição de custos da Tabela 5:

Tabela 5 – Composição de custo unitário, corte de áreas gramadas – 1.000 m²

COMPONENTES	UNID.	CONSUMO S
Caminhão carroceria – 4 t (80 kW)	h (produtiva)	0,50
	h (improdutiva)	0,50
Roçadeira – em micro trator (10 kW)	h (produtiva)	1,00
Encarregado	h	0,10
Servente	h	2,00

Fonte: Departamento Nacional de Infraestrutura de Transportes (2003, p. 178)

3.2.1.1.2 Irrigação

A irrigação de gramados se divide em irrigação relacionada ao plantio e irrigação pós plantio. De acordo com a Associação Nacional Grama Legal (2017a), a irrigação no período após o plantio é condição para que o gramado consiga se estabelecer na área de plantio, devendo ser feita em dias alternados nos primeiros 15 dias após o plantio. A necessidade de irrigação depende da espécie da grama, do tipo de solo (Giacoia Neto, 2004), do clima, da temperatura, da velocidade do vento e das práticas utilizadas na manutenção do gramado (Raats, 2005). De acordo com Giacoia Neto (2004), gramas com baixa resistência à seca podem necessitar de três a quatro irrigações por semana nos meses mais quentes, enquanto gramas com alta resistência à seca necessitam de apenas uma.

Para a estimativa do tempo de execução da atividade de irrigação de gramados, foram consideradas as composições de custos das Tabelas 6 e 7. Qual composição utilizar será função de qual tipo de irrigação está sendo considerada.

Tabela 6 – Irrigação de área verde com caminhão – unidade: m²

COMPONENTES	UNID.	CONSUMOS
Servente	h	0,0025
Água	m ³	0,0075
Caminhão irrigadeira, diesel, potência 167 HP (125 kW), capacidade 8.000 l - vida útil 8.000 h	h prod.	0,0013

Fonte: adaptado de PINI (2010, p. 115)

Tabela 7 – Irrigação de área verde com pulverizador – unidade: ha

COMPONENTES	UNID.	CONSUMOS
Servente	h	12,00
Água	l	3,50
Pulverizador costal com capacidade de 20 litros	h	12,00

Fonte: adaptado de Embrapa (2012, p. 19)

3.2.1.1.3 Remoção de espécies invasoras

Planta invasora é qualquer planta que cresça onde não é desejada. Elas surgem no meio da grama e naturalmente se multiplicam, comprometendo o gramado de variadas formas: competição por água, luz, CO₂ e nutrientes; possibilidade de inibição química do desenvolvimento do gramado; e prejuízo à aparência geral do gramado (BECKER, 2012).

Segundo Becker (2012), no caso de gramados, existem duas formas de combate às espécies invasoras: mecânico e químico. De acordo com a Associação Nacional Grama Legal (2017b), as principais práticas para esse controle são: arranquio, capina, roçada (combate mecânico); e aplicação de herbicidas (combate químico). Matos e Renault (200-?) esclarecem que o uso de herbicidas só é recomendado em grandes áreas de gramado em casos sérios de ataque de espécies invasoras, pelo fato de os herbicidas serem extremamente tóxicos e poderem contaminar o solo, os cursos d'água e o lençol freático. Seu uso fica definitivamente impossibilitado num campus universitário já que, segundo a Associação Nacional Grama Legal (2017b), em áreas urbanas não é permitido o uso de herbicidas agrícolas; o único uso permitido é de produtos domissanitários, que são de baixa eficiência.

De acordo com a Associação Nacional Grama Legal (2017b), o foco deve ser o combate às plantas invasoras de crescimento vertical. Plantas invasoras de

crescimento rasteiro não precisam ser eliminadas, ficando a cargo do gestor do gramado removê-las ou não.

Para a estimativa do tempo de execução da atividade de remoção de espécies invasoras, foram consideradas as composições de custos das Tabelas 8, 9 e 10. A composição utilizada será função de qual tipo de remoção está sendo considerada.

Tabela 8 – Remoção de espécies invasoras (manual) – unidade: ha

COMPONENTES	UNID.	CONSUMOS
Servente	h	40

Fonte: adaptado de Embrapa (2012, p. 6)

Tabela 9 – Remoção de espécies invasoras (mecanizada) – unidade: ha

COMPONENTES	UNID.	CONSUMOS
Servente	h	1
Trator 50 HP	h prod.	1

Fonte: adaptado de Embrapa (2012, p. 6)

Tabela 10 – Remoção de espécies invasoras (semi-mecanizada) – unidade: ha

COMPONENTES	UNID.	CONSUMOS
Servente	h	20
Moto-roçadeira costal	h	20

Fonte: adaptado de Embrapa (2012, p. 6)

Em relação à frequência da atividade, abordando o caso de arbustos, Giulietti *et al.* (2018), tratando do caso da Floresta Nacional de Carajás, recomendou ações mensais durante o período de chuvas (de janeiro a maio) e a cada dois meses no restante do ano. Já Venner (2006) relatou a obtenção de bons resultados no controle dessas espécies com ações realizadas entre duas e quatro vezes por ano.

3.2.1.2 Levantamento das frequências das atividades de manutenção em áreas arborizadas

Em relação à atividade de poda em árvores, a consolidação de dados é complexa. As podas dependem de fatores como: idade da árvore, tamanho, modelo arquitetônico da espécie (porte, forma da copa, disposição das folhas), entre outros. A correta

realização das podas durante o crescimento inicial da árvore também reduz a necessidade de podas na idade adulta. De acordo com a finalidade, há também diferentes tipos de poda (ENEL, 2020):

- poda de formação;
- poda de condução;
- poda de limpeza;
- poda de controle;
- poda de adequação;
- poda de emergência;
- poda de raízes.

No entanto, com o objetivo de uniformizar o tratamento do problema e simplificar o modelo, considerou-se apenas um tipo de poda. Pela complexidade que representaria tratar individualmente cada árvore, foram adotados dados que estratificam todas as árvores por seu diâmetro médio de tronco: menores do que 20 cm, de 20 cm a 40 cm, de 40 cm a 60 cm e maiores do que 60 cm. Para a estimativa do tempo de execução da atividade de poda em árvores foram consideradas as seguintes composições de custos das Tabelas 11, 12, 13, 14, 15, 16 e 17.

Tabela 11 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,20 m e menor que 0,40 m

COMPONENTE	UNIDADE	CONSUMO
Servente	h	1,54440
Jardineiro	h	1,54440

Fonte: Caixa Econômica Federal, 2023, p. 60

Tabela 12 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,40 m e menor que 0,60 m

COMPONENTE	UNIDADE	CONSUMO
Servente	h	2,75110
Jardineiro	h	2,75110

Fonte: Caixa Econômica Federal, 2023, p. 62

Tabela 13 – Corte raso e recorte de árvore com diâmetro de tronco maior ou igual a 0,60 m

COMPONENTES	UNID.	CONSUMO S
Servente	h	3,67750
Jardineiro	h	3,67750
Guindaste hidráulico autopropelido, com lança telescópica	h prod.	0,19290
Guindaste hidráulico autopropelido, com lança telescópica	h improd.	0,78810

Fonte: Caixa Econômica Federal, 2023, p. 64

Tabela 14 – Poda em altura de árvore com diâmetro de tronco menor que 0,20 m

COMPONENTES	UNID.	CONSUMO S
Guindauto hidráulico	h prod.	0,20480
Guindauto hidráulico	h improd.	0,48730
Servente	h	0,69210
Jardineiro	h	0,69210

Fonte: Caixa Econômica Federal, 2023, p. 66

Tabela 15 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,20 m e menor que 0,40 m

COMPONENTES	UNID.	CONSUMO S
Guindauto hidráulico	h prod.	0,40690
Guindauto hidráulico	h improd.	1,75020
Servente	h	2,15980
Jardineiro	h	2,15980

Fonte: Caixa Econômica Federal, 2023, p. 68

Tabela 16 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,40 m e menor que 0,60 m

COMPONENTES	UNID.	CONSUMO S
Guindauto hidráulico	h prod.	1,22880
Guindauto hidráulico	h improd.	3,99020
Servente	h	5,21900
Jardineiro	h	5,21900

Fonte: Caixa Econômica Federal, 2023, p. 70

Tabela 17 – Poda em altura de árvore com diâmetro de tronco maior ou igual a 0,60 m

COMPONENTES	UNID.	CONSUMO S
Guindauto hidráulico	h prod.	1,63840
Guindauto hidráulico	h improd.	7,15810
Servente	h	8,79650
Jardineiro	h	8,79650

Fonte: Caixa Econômica Federal, 2023, p. 72

Em relação à frequência com que os serviços de poda devem ser realizados, todas as recomendações encontradas dizem respeito à espécie específica tratada na publicação. No estudo de Pezzutti *et al.* (2016), realizado com pinus, ele relatou que as podas foram realizadas anualmente. Menezes (2021), tratando de oliveiras, indicou que a frequência de podas pode ser anual, bianual ou polianual. Souza e Velasco (2022), relataram a percepção de que na maioria das cidades brasileiras as podas em árvores urbanas ocorrem numa frequência acima do necessário, com o objetivo de prevenir problemas causados pela vegetação arbórea.

4 O SOFTWARE PROPOSTO

O objetivo desta pesquisa foi desenvolver um software que aplicasse a meta-heurística da ACO ao dimensionamento das equipes de jardinagem em campi universitários. Optou-se pela implementação do software como um *plug-in* para o AutoCAD em função de dois fatores principais: (1) a predominância da utilização do AutoCAD nos setores de engenharia das prefeituras universitárias, e (2) a existência da ferramenta *AutoCAD Geolocation*, presente pela primeira vez na versão 2014 (Walmsley, 2013), que permite a geolocalização e visualização num estilo de vista de satélite de desenhos feitos no software.

4.1 LINGUAGENS UTILIZADAS

Um *plug-in* é um software secundário, que funciona dentro de um software principal, adicionando novos comandos a seus comandos nativos, de modo a aumentar sua funcionalidade. A própria Autodesk esclarece a respeito da criação de soluções baseadas no AutoCAD:

O software Autodesk AutoCAD® é uma plataforma para o desenvolvimento de aplicações especializadas de projeto e desenho. Sua arquitetura aberta e APIs permitem uma personalização que se estende a *aplicações* de software de desenho específicas para diferentes indústrias, desenvolvidos pela própria Autodesk, e a uma vasta gama de aplicações adicionais criadas com o auxílio da plataforma Autodesk Developer Network (ADN). O AutoCAD fornece ambientes de programação, incluindo o ObjectARX® com interfaces C++ orientadas a objetos, a API .NET gerenciada, o ambiente de desenvolvimento Visual LISP® completo e a interface ActiveX® (Autodesk, 2023, tradução do autor).

De acordo com a Autodesk (2013), há três interfaces de programação 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.

O *.NET Framework* é um ambiente de execução gerenciado para o Windows que oferece uma variedade de serviços aos aplicativos em execução. Ele consiste em dois componentes principais: o CLR (*Common Language Runtime*), o mecanismo de execução que manipula aplicativos em execução, e a biblioteca de classes *.NET Framework*, que oferece uma biblioteca de códigos testados e reutilizáveis que os desenvolvedores podem chamar de seus próprios aplicativos (Microsoft, 2023a, tradução do autor).

Aplicações *.NET* são escritas nas linguagens C#, F# ou Visual Basic. O código é compilado num código intermediário chamado de CIL (*Common Intermediate Language*), que independe da linguagem de programação utilizada. O código compilado é armazenado em arquivos binários – arquivos com extensões DLL ou EXE (Microsoft, 2023b, tradução do autor).

A linguagem C# foi definida como linguagem principal para a produção do software, mas não foi a única: também foram utilizados Python e LaTeX⁶. Dentro da solução proposta, o código que consolida os dados de entrada, propõe o problema como uma instância do MTSP, e o envia para ser resolvido aplicando a ACO para cada tamanho de time – crescendo progressivamente de 1 até o resultado – foi implementado em C#. O código que aplica a meta-heurística da ACO para resolver o MTSP proposto foi implementado em Python. O relatório com o resultado da simulação foi feito utilizando LaTeX.

O algoritmo que aplica a meta-heurística da ACO ao MTSP é o TACO. O código principal em C# fornece para o ele todos os dados (os dados consolidados e a localização dos nós) e importa dele o resultado (as sequências de nós que constituem as soluções individuais, que juntas representam a solução do MTSP). O código do TACO, especificamente, utilizou como base a versão implementada em Python por Gabriel Lima, disponível no link: <https://github.com/limaagabriel/taco>. Para a geração do relatório, criou-se um arquivo TXT com a estrutura básica do arquivo (seguindo a sintaxe da linguagem LaTeX). Esse arquivo é modificado pelo código principal, com a inserção dos dados reais, e convertido de TXT para TEX. O código principal executa o compilador de LaTeX, que lê o arquivo TEX e cria o relatório em PDF. Esse relatório

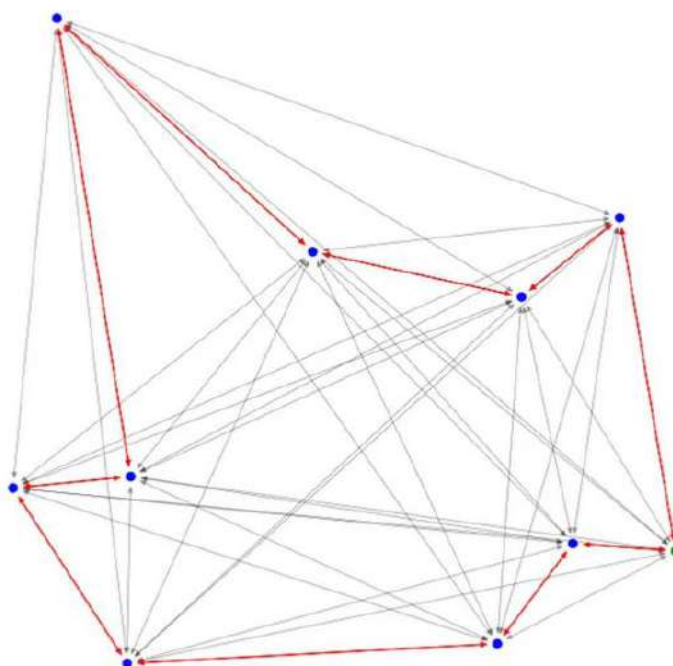
⁶ “LaTeX é um sistema para composição tipográfica de documentos (...). Ele se tornou uma língua franca da produção científica: (...) artigos (...) na forma de dados de entrada em LaTeX.” (LAMPOR, 1994, p. 2, tradução do autor).

é então inserido dentro do arquivo RAR com o resultado da simulação gerado pelo *plug-in*.

4.2 DEFINIÇÃO DO PROBLEMA COMPUTACIONAL

O problema a ser resolvido é a determinação do número de trabalhadores que, saindo da origem (local de início da jornada), consigam realizar em conjunto todas as atividades de manutenção existentes no campus e definidas para serem executadas e retornar à origem sem exceder o tempo de seu turno de trabalho. Isso será atendido quando o maior tempo de percurso entre todos os trabalhadores for menor ou igual à duração do turno. Cada trabalhador (em um turno de trabalho) constitui uma formiga numa instância do MTSP.

Figura 3 – Representação gráfica do TSP



Fonte: Isik; Atkin (2023)

Matematicamente, o TSP pode ser entendido como um conjunto de vértices (V) onde todos os pontos estão conectados entre si por arestas (A), como representado pela Figura 3. Cada elemento de A possui um custo associado (ao deslocamento por ela), normalmente dado pelo comprimento da aresta. Esses custos associados podem ser organizados numa matriz C ; nessa matriz C , cada elemento c_{ij} é o custo do

deslocamento do vértice $i \in V$ até o vértice $j \in V$. A matriz C é simétrica se $c_{ij} = c_{ji}$, e assimétrica se $c_{ij} \neq c_{ji}$. A variável x_{ij} informa se a aresta entre os vértices i e j faz ou não parte da solução:

- $x_{ij} = 0$, se a aresta entre os vértices i e j não pertence à solução;
- $x_{ij} = 1$, se a aresta entre os vértices i e j pertence à solução.

No TSP, existe apenas um caixeiro; no MTSP, existem $m > 1$ caixeiros. A função a ser minimizada (representação matemática do problema) é dada pela Equação 12, e as restrições são dadas pelas Equações 13, 14, 15, 16, 17 e 18. Nessas formulações, x_{ij} indica se o arco entre os nós i e j é ou não percorrido.

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (\text{Eq. 12})$$

$$\sum_{j=2}^n x_{1j} = m \quad (\text{Eq. 13})$$

$$\sum_{j=2}^n x_{1j} = m \quad (\text{Eq. 14})$$

$$\sum_{i=1}^n x_{ij} = 1, j = 2, \dots, n \quad (\text{Eq. 15})$$

$$\sum_{j=1}^n x_{ij} = 1, i = 2, \dots, n \quad (\text{Eq. 16})$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V \setminus \{1\}, S \neq \emptyset \quad (\text{Eq. 17})$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V \quad (\text{Eq. 18})$$

A Equação 12 descreve a função objetiva a ser minimizada, como sendo a soma dos custos c_{ij} associados a cada uma das arestas entre os vértices i e j que pertencem à solução. As restrições das Equações 13 e 14 garantem que todos os m caixeiros saiam da origem e retornem a ela. As restrições das Equações 15 e 16 garantem que cada vértice seja visitado uma única vez, ao restringir a 1 o número de destinos que aquele vértice pode ter enquanto origem.

A restrição da Equação 17 estabelece que, sendo S qualquer conjunto de vértices pertencentes a V que não inclua a origem, o número de arestas entre os vértices de S será menor ou igual ao número de vértices menos 1. Isso implica que cada par de vértices poderá ser conectado uma única vez, ou seja, não poderá haver mais de um caminho levando ao mesmo vértice, garantindo que não sejam criados caminhos entre nós que já estejam conectados. A restrição da Equação 18 especifica que o valor de x_{ij} será igual a 1 ou 0 para todas as arestas que ligam todos os vértices i e j pertencentes a V .

4.2.1 Team Ant Colony Optimization

Para resolver o MTSP, foi utilizado o Team Ant Colony Optimization, algoritmo proposto por Vallivaara (2008, *apud* Barbosa, Silva Jr., Kashiwabara, 2015), que adapta o Ant Colony System (desenvolvido originalmente para resolver o TSP) para o MTSP. Ao invés de um único caixeiro, o algoritmo considera a existência de m caixeiros viajantes. Considerando um problema com N vértices, cada caixeiro guarda a informação (memória) dos vértices visitados, e o time guarda a informação dos vértices visitados por todos os caixeiros.

No início da simulação, todos as formigas (caixeiros) estão na origem, e a memória de todas está vazia (assim como a lista de vértices visitados pelo time). Enquanto o quantitativo de nós N do problema for maior do que a soma dos nós visitados pelos caixeiros o processo de atribuição de nós continua. Em cada passo da simulação, a formiga com menor rota parcial escolhe probabilisticamente (de acordo com a regra do Ant Colony System) seu próximo nó. No entanto, antes que a formiga efetivamente se mova, é verificado se, caso esse nó fosse cedido a qualquer outra formiga do time, isso resultaria num tamanho total menor para a solução do time. Se isso for verdadeiro, a formiga inicial não se move, mas sim a formiga cujo movimento resultou numa menor distância total para a solução. Essa formiga também ganha o direito de escolher seu próximo nó.

4.2.2 Modificação implementada no TACO para aprimorar o balanceamento de cargas

Segundo Vallivaara, a imposição de que a formiga com menor rota parcial seja sempre a que escolhe seu próximo nó tem por objetivo proporcionar uma distribuição eficiente de comprimentos de percurso pelas formigas do time. Contudo, outra característica do algoritmo acaba comprometendo o efeito desse mecanismo: o fato de o nó ser oferecido a todas as outras formigas do time – antes que o movimento se dê efetivamente – pode acarretar numa distribuição pouco eficiente.

No presente estudo, com o TACO sendo empregado para a solução de um MTSP com mais de cem vértices espalhados pelo campus da UFES, em Goiabeiras, notou-se que com times de até 5 formigas o método proposto por Vallivaara resultava numa distribuição eficiente dos comprimentos dos trajetos, mas com times de 6 ou mais formigas a distribuição se tornava irregular, chegando a diferenças de mais de 90% entre o maior e o menor comprimentos com 20 formigas no time.

Para evitar que a simulação resultasse numa distribuição tão desigual, foi estabelecido um limite para o número de vértices da solução de cada formiga, dado pelo produto de uma constante N pela razão entre o número total de vértices do problema e o número de formigas no time. Novos vértices continuarão sendo oferecidos inicialmente à formiga com menor distância total de trajeto, mas deixarão de ter suas cessões às formigas que já tenham atingido o limite de vértices testadas. Ao final ainda poderá haver formigas com soluções maiores que o valor definido, mas o estabelecimento desse limite será capaz de proporcionar uma solução com melhor distribuição de vértices entre as formigas do time.

Foram comparadas as eficácias de diferentes valores para a constante N : 1,05; 1,10; 1,15; 1,2; 1,25; e 1,30. Para verificação das eficácias, foi utilizado como referência um tipo de trabalhador realizando apenas o serviço de poda em gramados (com a frequência de uma vez por semana), e mantidas as demais configurações. Foram comparados os resultados para a busca inicial (Tabela 18). O valor que apresentou um menor resultado para o tamanho do time ao final da busca inicial foi 1,2, que, portanto, foi adotado.

Tabela 18 – Resultados da busca inicial para diferentes valores da constante N

Constante	Resultado da busca inicial	Duração do maior percurso
1,05	TS = 51	3:50:42
1,10	TS = 56	3:37:58
1,15	TS = 51	3:37:58
1,20	TS = 46	3:59:15
1,25	TS = 51	3:50:42
1,30	TS = 56	3:37:58

Fonte: elaborado pelo autor (2025)

4.3 CARACTERÍSTICAS DO SOFTWARE

Como um *plug-in* para o AutoCAD, o software produzido é carregado junto com o AutoCAD, quando este é aberto. O *plug-in* adiciona uma nova guia às existentes em sua área de trabalho: a guia *Green Maintenance Optimizer*. Nessa guia estão todos os comandos do *plug-in*, organizados em painéis.

O fluxo de trabalho no *plug-in* se inicia pela geolocalização do campus, utilizando o comando *Set Location* (nativo do AutoCAD). Selecionando a opção *From Map*, é possível especificar a localização geográfica exata do campus, e a partir de então tudo o que for desenhado estará referenciado à localização definida. Feito isso, a visualização do mapa da região selecionada estará habilitada dentro do AutoCAD, em escala real. O passo seguinte é o desenho das áreas. Todas as áreas de gramado e de interesse paisagístico a serem consideradas na simulação devem ser desenhadas utilizando a ferramenta *Polyline*, na forma de polilinhas fechadas. Todas as áreas de gramado devem ser colocadas no *layer Grass*.

A localização das árvores necessita ser importada para o desenho, e colocadas em sua localização real, no *layer Trees*. Isso pode ser feito através da utilização de arquivos GIS, com inventários da vegetação arbórea. Muitas cidades, parques e campi universitários possuem esse inventário atualizado, onde normalmente consta a localização e a espécie da árvore. A origem deve ser representada por um círculo, desenhado no *layer Origin*. As configurações das

Deve-se ajustar as configurações das vegetações. As configurações dos gramados incluem as frequências das ações de poda e irrigação (variando de 2 vezes por semana a uma vez a cada 2 meses) e o intervalo entre as ações de remoção de espécies invasoras (variando de 1 a 12 meses). As configurações das árvores incluem sua estimativa de distribuição por faixa de diâmetro médio de tronco: menores que 20 cm, entre 20 cm e 40 cm, entre 40 cm e 60 cm, e maiores que 60 cm; incluem também a frequência de ações de poda (variando de uma vez a cada ano a uma vez a cada 5 anos).

Em seguida, deve-se fazer as configurações relacionadas aos trabalhadores. Por padrão, um tipo de trabalhador vem configurado: *Outsourced worker*⁷ (trabalhador terceirizado), mas é possível criar novos tipos de trabalhador. Também estão disponíveis configurações de: velocidade de deslocamento; número de turnos de trabalho por dia; duração do turno de trabalho; e tempo de descanso durante o turno. A velocidade de deslocamento pode ser configurada no intervalo de 1 km/h a 10 km/h. O número de turnos de trabalho por dia pode ser configurado no intervalo de 1 a 2. Jornadas de trabalho compostas por um único turno terão turno com duração de 6 horas, e jornadas de trabalho compostas por dois turnos terão turnos com duração de 4 horas. O tempo de descanso durante o turno de trabalho pode ser configurado no intervalo de 0 a 60 minutos. O tempo de descanso será subtraído da duração do turno de trabalho na simulação, significando que nesse período o trabalhador está em descanso e não cumprindo o seu percurso de trabalho.

Feitas as configurações relacionadas aos trabalhadores, pode-se passar às configurações das atividades de manutenção propriamente ditas. Essas configurações estão separadas em gramados e árvores. Para gramados, é possível considerar ou desconsiderar na simulação as atividades de irrigação, poda e remoção de espécies invasoras. Para cada uma delas deve-se definir o tempo gasto de encarregado e ajudante para a execução de 1000 m² de serviço. Também deve-se especificar qual trabalhador irá executar cada um dos serviços. Para árvores, as configurações são de tempo gasto por encarregado e ajudante na execução de uma

⁷ A criação desse tipo de trabalhador como padrão buscou refletir a realidade dos *campi* de universidades públicas em que esses serviços ficam a cargo de empresas e trabalhadores terceirizados, e não de trabalhadores das próprias universidades.

unidade do serviço, de acordo com a seguinte estratificação, por tamanho de tronco: menores do que 20 cm; de 20 cm a 40 cm; de 40 cm a 60 cm; e maiores do que 60 cm. Também deve-se especificar qual trabalhador irá executar o serviço.

Feitas essas configurações, deve-se clicar no botão *Run Simulation*. Ele abre a janela de simulação, onde as demais configurações são feitas. Essas configurações incluem: escala da grade de pontos (para gramados e para árvores); número de iterações (para busca inicial e para busca refinada); número de times (para busca inicial e para busca refinada); e estação do ano a ser considerada na simulação.

Todas as configurações citadas vêm preenchidas por padrão no *plug-in*, mas podem ser modificadas a fim de alterar as características da simulação. As configurações que vêm carregadas por padrão retratam um caso genérico, que poderia ser simulado e representaria um quadro plausível. Do mesmo modo, todas as possibilidades de configuração foram propostas com o objetivo de simular cenários possíveis, sem se afastar da realidade.

A simulação não tem por objetivo definir o número de trabalhadores para que os serviços sejam concluídos, mas sim para que os serviços sigam sendo realizados com a periodicidade desejada. Ou seja, a questão da manutenção é tomada como um problema contínuo, e admitido constante (enquanto o cenário definido pelos dados de entrada for o mesmo).

A simulação é dividida em duas etapas: busca inicial e busca refinada. Para cada uma é possível definir o número de times que serão utilizados e o número de iterações. O número de times no MTSP funciona como o número de formigas no TSP: a simulação será repetida para cada time, e o resultado sobre o ambiente (depósito de feromônios) será somado. Um maior número de times aumenta o custo computacional da simulação, e a precisão do resultado. O número de iterações define quantas vezes o processo descrito na definição do TACO será repetido.

A busca inicial foi projetada para utilizar menos formigas e menos iterações, resultando em menor custo computacional em comparação com a busca refinada, que emprega configurações mais exigentes. Nessa fase inicial, o tamanho do time é

aumentado progressivamente até que se atinja o critério de parada; a duração da maior jornada entre todos os trabalhadores deve ser inferior à duração do turno de trabalho predefinida. Atingido esse ponto, inicia-se a busca refinada, em que o processo é repetido, desta vez com a remoção de trabalhadores, a fim de verificar se, com um custo computacional mais elevado, é possível satisfazer o critério de parada com um número menor de agentes.

A variação no tamanho do time de formigas foi prevista inicialmente para ocorrer em uma unidade por vez, tanto para mais quanto para menos. No entanto, durante o desenvolvimento ficou claro que dessa forma a simulação acabava sendo repetida muitas vezes desnecessariamente, considerando que mesmo resultados da busca inicial com imprecisões maiores seriam igualmente otimizados pela busca refinada. Dessa forma, foi definida uma estratégia para permitir uma progressão mais rápida da simulação, em relação ao tamanho do time:

- a simulação sempre se inicia com 1 formiga no time;
- na busca inicial, enquanto a duração da maior jornada for maior ou igual a 150% da duração do turno de trabalho, o tamanho do time é acrescido em 10 formigas;
- na busca inicial, a partir do momento em que a duração da maior jornada for menor do que 150% da duração do turno de trabalho, o tamanho do time é acrescido em 5 formigas;
- na busca refinada, o tamanho do time é reduzido em 1 formiga a cada iteração.

4.3.1 SIMPLIFICAÇÕES E CONSOLIDAÇÃO DOS DADOS

A partir das configurações de entrada, o *plug-in* consolida os dados que serão utilizados na simulação a partir de uma série de simplificações e convenções adotadas. A manutenção das áreas verdes é considerada um problema discreto⁸. As semanas são consideradas de 5 dias (os fins de semana não são considerados na correção das frequências); os meses são considerados de 4 semanas e 22 dias

⁸ Um problema discreto é aquele que envolve um conjunto finito ou infinito enumerável de possibilidades, em contraste com problemas contínuos, que lidam com quantidades variáveis em um domínio infinito não enumerável. Problemas discretos frequentemente surgem em contextos de matemática discreta, ciência da computação e otimização combinatória, onde as soluções são representadas por estruturas como grafos, sequências ou arranjos (Rosen, 2019, p. 3).

(removendo os fins de semana de um período de 30 dias); os anos são considerados de 365 dias. Essas convenções foram feitas para padronizar valores que variam mês a mês, pela quantidade diferente de dias de cada um.

As simplificações, convenções e decisões de modelagem adotadas nesta pesquisa constituem o núcleo da contribuição metodológica da tese. Embora a Otimização por Colônias de Formigas (ACO) seja empregada como mecanismo de busca para a definição de rotas — em consonância com sua aplicação clássica em problemas como o TSP e o MTSP —, seu papel é deliberadamente restrito à otimização de trajetos a partir de um conjunto de parâmetros previamente definidos.

Figura 4 – Fluxograma de funcionamento do plug-in



Fonte: elaborado pelo autor (2025)

O diferencial da abordagem proposta não reside na adaptação do algoritmo em si, mas na forma como o problema real da manutenção de áreas verdes é estruturado,

consolidado e iterativamente reformulado. Esse processo permite a avaliação sistemática de diferentes configurações operacionais por meio de sucessivas execuções do algoritmo, com variação controlada do tamanho das equipes e das condições de operação.

A Figura 4 apresenta uma visão geral desse fluxo metodológico, evidenciando a separação entre a modelagem do problema, a consolidação dos dados e a aplicação da meta-heurística. Nesse contexto, a ACO atua como um motor interno de otimização, enquanto os mecanismos de redimensionamento sucessivo das equipes e de interpretação dos resultados são específicos da abordagem desenvolvida nesta tese.

A simulação utiliza grades de pontos sobre o campus (com a escala definida pelo usuário) para representar as atividades de manutenção, tanto para gramados quanto para árvores. Essa estratégia foi utilizada para reduzir a complexidade do problema. No caso dos gramados, ao representar áreas por pontos a abordagem reduz a complexidade que seria tratar matematicamente de áreas contínuas. No caso das árvores, ao agrupar árvores próximas num mesmo ponto de grade a abordagem reduz a complexidade que seria abordar um número muito maior de pontos. Em ambos os casos, ao deixar a cargo do usuário a definição da escala da grade, a abordagem permite que a representação se adeque à escala do problema.

Cada área de gramado tem a integralidade das atividades nela realizadas contabilizada e dividida pelo número de pontos de grade que estão localizados em seu interior. A distância de cada árvore é medida em relação a todos os pontos de grade (de árvores) existentes, e ela é atribuída ao ponto de grade mais próximo. Cada ponto de grade terá um número de árvores atribuído, e a soma dos números de árvores atribuídas a cada um dos pontos é igual ao número total de árvores. Pontos sem nenhuma árvore atribuída são removidos da grade, e deixam de fazer parte da simulação. Cada ponto (seja representante de gramados ou árvores) recebe o que convencionou-se chamar de custo de permanência: o tempo que é acrescido à jornada do trabalhador que passar por aquele ponto.

Todas as árvores localizadas no interior do campus são consideradas; já para gramados, apenas as áreas de gramado desenhadas no campus são consideradas. As atividades de manutenção em gramados (irrigação, poda e remoção de espécies invasoras) podem ser consideradas ou não na simulação, a depender da configuração feita pelo usuário, e essa configuração será aplicada a todas as áreas de gramado; as atividades de manutenção em árvores são obrigatoriamente consideradas. Apesar dos trabalhos de encarregado e ajudante estarem discriminados nas configurações (para reproduzir as composições de custo unitário dos serviços), ambas as atividades são executadas pelo mesmo profissional (de acordo com a configuração feita pelo usuário), de modo que o tempo gasto pelo profissional será igual à soma dos tempos de encarregado e ajudante, como ilustrado pela Equação 19.

$$T_p = T_e + T_a \quad (\text{Eq. 19})$$

onde: T_p = tempo total gasto pelo profissional; T_e = tempo correspondente à função de encarregado; T_a = tempo correspondente à função de ajudante.

O tempo de execução de cada serviço a ser executado nas áreas de gramado é dado pelo produto da área e do consumo de mão de obra por metro quadrado; esse tempo de execução é multiplicado pela frequência (para cada serviço); o total dos produtos, corrigido para um período de um dia, é o custo de permanência considerado na simulação, como ilustrado pela Equação 20.

$$C_g = \frac{\sum_{i=1}^n (A \times c_i \times f_i)}{P} \quad (\text{Eq. 20})$$

onde: C_g = custo de permanência do gramado; A = área do gramado; c_i = consumo de mão de obra para o serviço i ; f_i = frequência de execução do serviço i (vezes/semana, mês ou ano); P = período de referência (em dias) para normalização do custo de permanência; n = número de serviços considerados na área de gramado.

Para as árvores, criou-se o conceito de unidade teórica. Definiu-se que cada unidade real de árvore seria representada por uma unidade teórica, que teria sua altura dividida

de acordo com a distribuição percentual das árvores por faixa de diâmetro médio de tronco (simplificando a representação de milhares de árvores). Em cada unidade teórica de árvore o tempo de execução do serviço de poda seria ponderado de acordo com a distribuição percentual do seu tronco por faixa de diâmetro. O custo de permanência de cada ponto de grade, adicionado ao tempo de trajeto da formiga que passasse pelo respectivo vértice, seria o produto do tempo de execução do serviço de poda numa unidade teórica e do número de árvores atribuídas ao ponto (Equação 21):

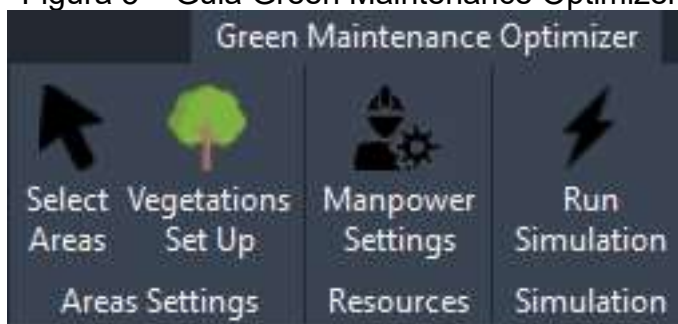
$$C_{aj} = C_a + T_{traj} \quad (\text{Eq. 21})$$

onde: C_{aj} = custo de permanência ajustado no ponto de grade; C_a = custo de permanência de poda para as árvores atribuídas ao ponto; T_{traj} = tempo de trajeto até o ponto. Todos os dados utilizados na construção do modelo proposto para o problema foram retirados da literatura consultada (capítulo 3).

4.4 APRESENTAÇÃO DO SOFTWARE

O *plug-in* adiciona à área de trabalho do AutoCAD a guia *Green Maintenance Optimizer* (Figura 5), que se soma às outras existentes. A aba está dividida em três painéis: *Areas Settings*, *Resources* e *Simulation*. No processo de concepção do *plug-in*, muitos comandos posteriormente agrupados num mesmo botão, ou que se tornaram desnecessários porque a estratégia adotada foi alterada, foram inicialmente pensados como botões específicos. Os painéis foram criados para agrupar os comandos por tipo de funcionalidade: os relacionados às áreas de vegetação, os relacionados a recursos humanos e os relacionados à execução da simulação propriamente dita. Apesar de em muitos casos ter restado apenas um botão nos painéis, eles foram mantidos.

Figura 5 – Guia Green Maintenance Optimizer



Fonte: elaborado pelo autor (2025)

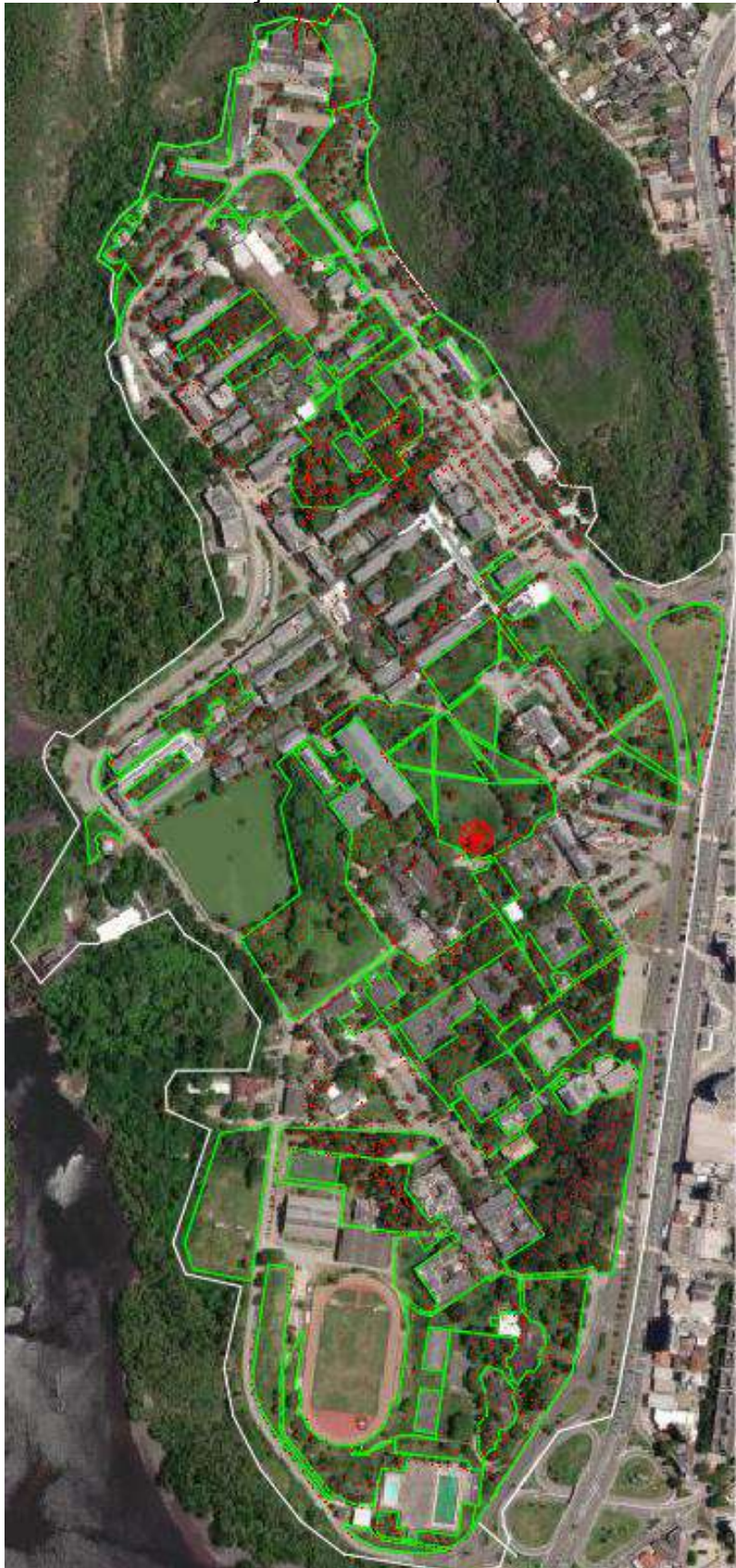
4.4.1 APLICAÇÃO AO CAMPUS DE GOIABEIRAS (CENÁRIO 1)

No AutoCAD, para utilizar o *plug-in*, deve-se criar um novo arquivo, acessar a guia *Insert*, e utilizar o comando *Set Location* (opção *From Map*). Isso abrirá a janela *Geographic Location – Specify Location*. Nela, deve-se selecionar a localização exata do *campus* em que se deseja aplicar o *plug-in*; a unidade de desenho também deve ser definida. Depois de definidas a localização e a unidade de desenho, tudo o que for desenhado na vista geográfica estará na escala definida.

Em relação à parte gráfica, a representação das áreas de gramado, limite do *campus* e origem, e importação da localização das árvores devem ser feitas de acordo com o fluxo descrito em 4.3, em relação a: desenhos das áreas de gramado e importação da localização das árvores.

A Figura 6 mostra o resultado para o *campus* de Goiabeiras, com todas os limites do *campus* e as áreas de gramado desenhadas, e as árvores representadas como pontos. Como pode-se observar, a distribuição geográfica do *campus* é na direção norte-sul; o *plug-in* utiliza as distâncias entre as coordenadas mais externas em ambas as direções para determinar em que direção o *campus* está distribuído, e rotacionar a visualização do *campus* na janela de simulação se for necessário. O objetivo é sempre exibir a maior dimensão da área na horizontal, para otimizar a visualização. No caso do *campus* de Goiabeiras, utilizado como modelo, o *campus* é girado em 90°.

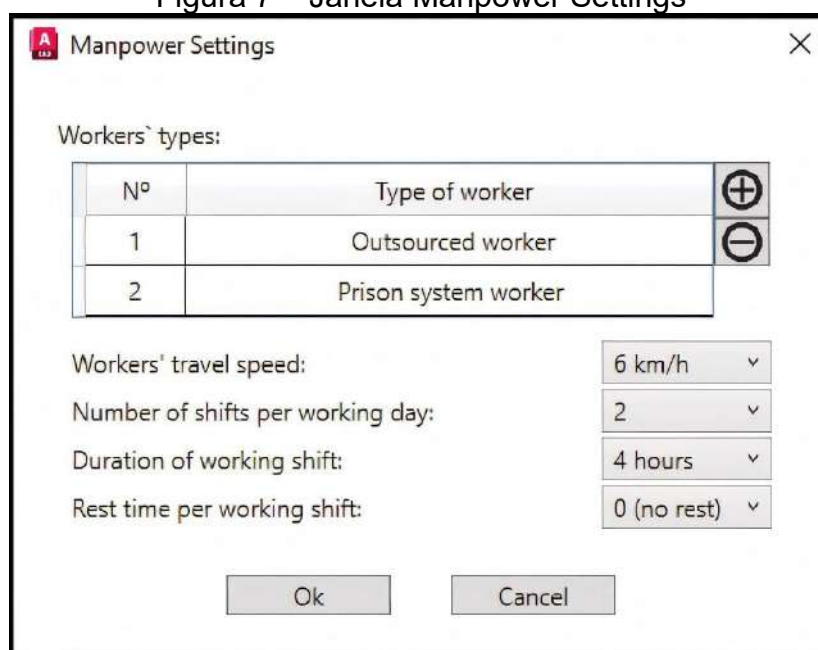
Figura 6 – Visualização do campus de goiabeiras com as áreas desenhadas e a localização das árvores importada



Fonte: elaborado pelo autor (2025)

Após concluída a parte gráfica, deve-se passar às configurações das vegetações e trabalhadores, que são feitas ainda na área de trabalho do AutoCAD. As configurações dos trabalhadores são feitas por meio do comando *Manpower Settings*, localizado no painel *Resources* da aba *Green Maintenance Optimizer*. Quando acionado ele abre uma janela de mesmo nome (Figura 7). Nela, os trabalhadores existentes podem ser renomeados, assim como novos tipos de trabalhadores podem ser criados⁹. Pode-se definir a velocidade de deslocamento dos trabalhadores, o número de turnos de trabalho, a duração dos turnos, e o período de descanso por turno de trabalho.

Figura 7 – Janela Manpower Settings



Fonte: elaborado pelo autor (2025)

Em seguida deve-se passar às configurações das vegetações. Essas configurações são feitas através do comando *Vegetations Set Up*, localizado no painel *Area Settings*. Quando acionado, ele abre a janela *Types of Vegetations*. Ela possui duas abas: *Grass* (Figura 8) e *Trees* (Figura 9). Na aba *Grass*, deve-se configurar as frequências das atividades de poda e irrigação de gramados, para dois conjuntos de estações: (1)

⁹ A criação do trabalhador *Prison system worker* (trabalhador do sistema prisional) buscou refletir a realidade do campus de Goiabeiras da UFES, utilizado como estudo de caso, em que as atividades de manutenção em vegetações são divididas entre trabalhadores terceirizados e do sistema prisional. Isso não, impede, no entanto, que seus nomes sejam modificados ou tipos genéricos de trabalhadores sejam criados para a simulação.

verão e primavera; e (2) inverno e outono; deve-se configurar também o intervalo entre as ações de remoção de espécies invasoras nos gramados. Na aba *Trees*, deve-se configurar a distribuição percentual das árvores existentes no campus por faixa de diâmetro médio de tronco; deve-se configurar também a periodicidade na qual cada árvore será podada.

Figura 8 – Janela Types of Vegetation (aba Grass)

Grass type	Pruning frequency (summer and spring)	Pruning frequency (winter and fall)	Irrigation frequency (summer and spring)	Irrigation frequency (winter and fall)
Zoysia Japônica (esmeralda)	1 time per week	1 time per week	1 time per week	1 time per week

Activities Setup Invasive species removal interval: 12 months

Cancel Ok

Fonte: elaborado pelo autor (2025)

Figura 9 – Janela Types of Vegetation (aba Trees)

Estimate the percentages of trees per trunk size range. These percentages will be applied uniformly to all areas.

Less than 20 cm	Equal to or greater than 20 cm and less than 40 cm	Equal to or greater than 40 cm and less than 60 cm	Greater than or equal to 60 cm
25	25	25	25

Activities Setup Tree pruning frequency: Once a year

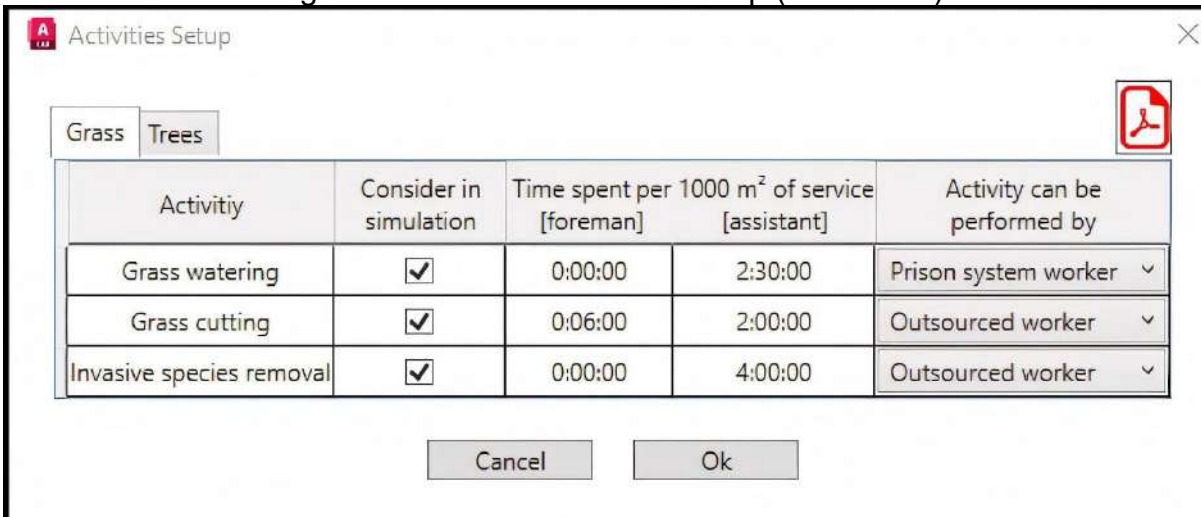
Cancel Ok

Fonte: elaborado pelo autor (2025)

Além disso, a janela *Types of Vegetation* possui o botão *Activities Setup*, que abre uma janela de mesmo nome. A janela *Activities Setup* também está dividida em duas abas: *Grass* (Figura 10) e *Trees* (Figura 11). Na aba *Grass* deve-se configurar, para as atividades de irrigação de gramados, poda de gramados e remoção de espécies invasoras: se serão consideradas ou não na simulação, tempos gastos por encarregado e ajudante para a execução de 1000 m² do serviço, e qual tipo de

trabalhador será responsável pela execução. Na aba *Trees*, deve-se configurar, para cada faixa de tamanho de tronco, os tempos gastos por encarregado e ajudante para a execução da poda em uma árvore. Deve-se configurar também qual trabalhador será responsável pela execução.

Figura 10 – Janela Activities Setup (aba Grass)



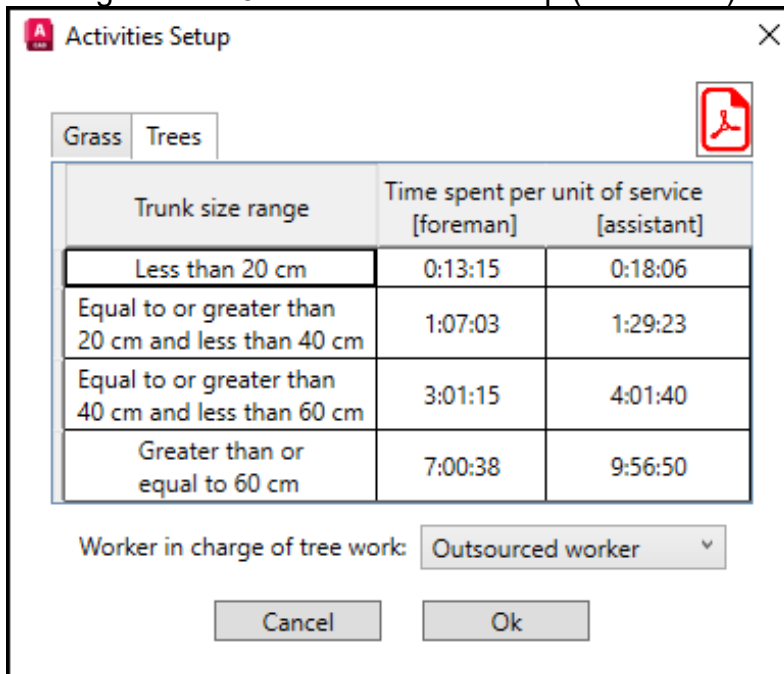
The screenshot shows the 'Activities Setup' window with the 'Grass' tab selected. It contains a table with the following data:

Activity	Consider in simulation	Time spent per 1000 m ² of service		Activity can be performed by
		[foreman]	[assistant]	
Grass watering	<input checked="" type="checkbox"/>	0:00:00	2:30:00	Prison system worker
Grass cutting	<input checked="" type="checkbox"/>	0:06:00	2:00:00	Outsourced worker
Invasive species removal	<input checked="" type="checkbox"/>	0:00:00	4:00:00	Outsourced worker

Buttons for 'Cancel' and 'Ok' are located at the bottom of the window.

Fonte: elaborado pelo autor (2025)

Figura 11 – Janela Activities Setup (aba Trees)



The screenshot shows the 'Activities Setup' window with the 'Trees' tab selected. It contains a table with the following data:

Trunk size range	Time spent per unit of service	
	[foreman]	[assistant]
Less than 20 cm	0:13:15	0:18:06
Equal to or greater than 20 cm and less than 40 cm	1:07:03	1:29:23
Equal to or greater than 40 cm and less than 60 cm	3:01:15	4:01:40
Greater than or equal to 60 cm	7:00:38	9:56:50

Below the table, there is a dropdown menu for 'Worker in charge of tree work:' set to 'Outsourced worker'. Buttons for 'Cancel' and 'Ok' are located at the bottom of the window.

Fonte: elaborado pelo autor (2025)

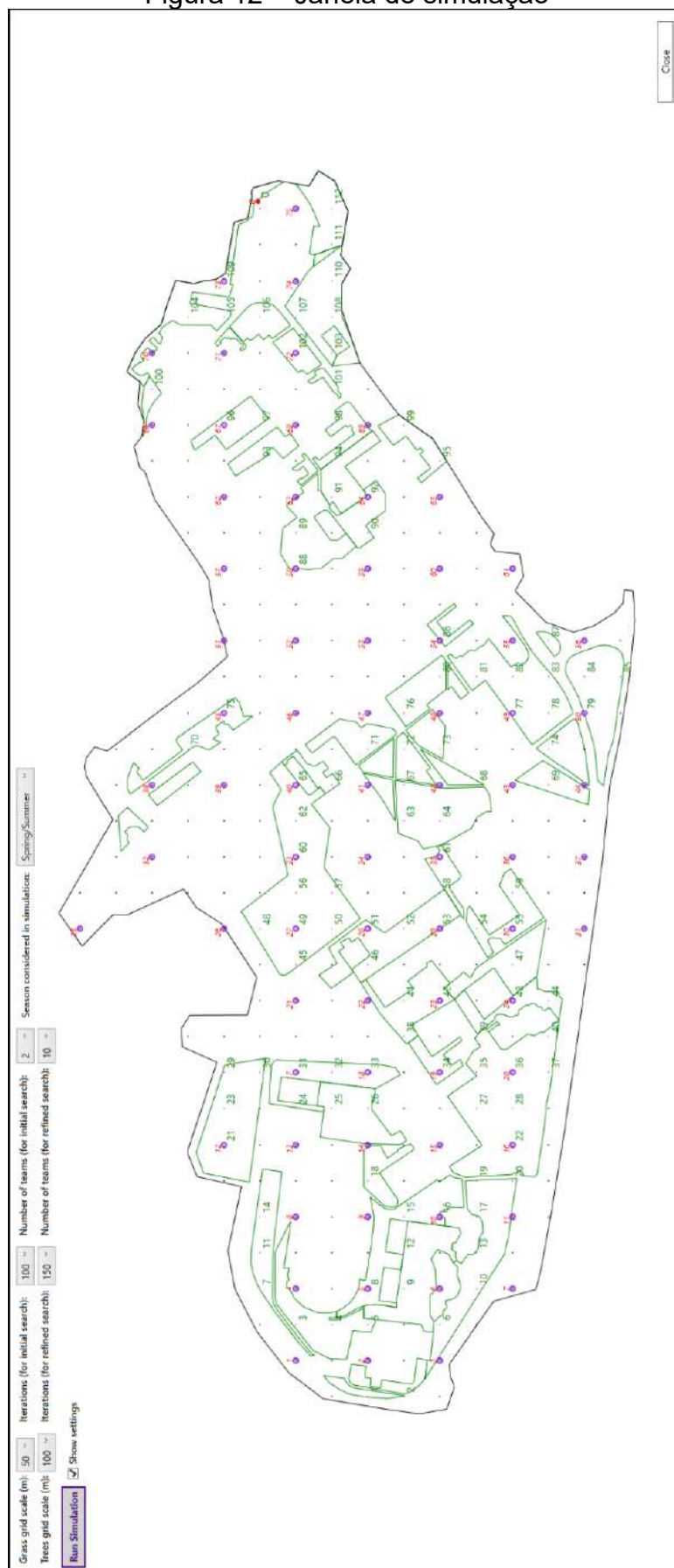
Concluídas essas etapas, pode-se passar às configurações finais da simulação. Para isso, deve-se utilizar o comando *Run Simulation*, localizado na aba *Simulation*. O

comando abre a janela de simulação (Figura 12). Nela podem ser alteradas as configurações de: escala das grades de pontos para gramados e árvores; números de iterações para as buscas inicial e refinada; números de times para as buscas inicial e refinada; e estação do ano a ser considerada na simulação.

A escala da grade pode ser alterada, mas deve ser mantida em um valor tal que não resulte em um número de pontos muito grande (o que aumentaria a complexidade computacional do problema, aumentando o tempo ou mesmo inviabilizando sua solução) e tampouco num número de pontos muito pequeno (que resulte em áreas de gramado sem nenhum ponto em seu interior, fazendo com que tais áreas sejam ignoradas pela simulação). O número de iterações pode ser configurado diferenciando a busca inicial da refinada; números maiores resultarão em um resultado mais preciso, mas acarretando tempos maiores de execução.

O número de times também pode ser configurado diferenciando a busca inicial da refinada. Números de times maiores farão com que mais times de formigas participem dos ciclos da simulação, resultando em um melhor desempenho da simulação em relação ao resultado, mas em um tempo de execução maior. A configuração da estação do ano a ser considerada na simulação irá impactar no alcance do resultado. Considerando que as estações de verão e primavera necessitam de frequências maiores dos serviços de irrigação e poda de gramados, os resultados de simulações que considerem essas estações poderão ser adotados para qualquer período de tempo, já que se tratam das estações com maiores demandas. Os resultados de simulações que considerem as estações de inverno e outono, por outro lado, serão restritos a essas estações, já que as demandas nessas estações são menores.

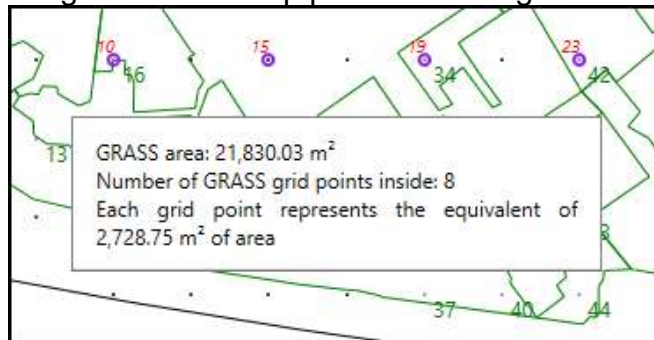
Figura 12 – Janela de simulação



Fonte: elaborado pelo autor (2025)

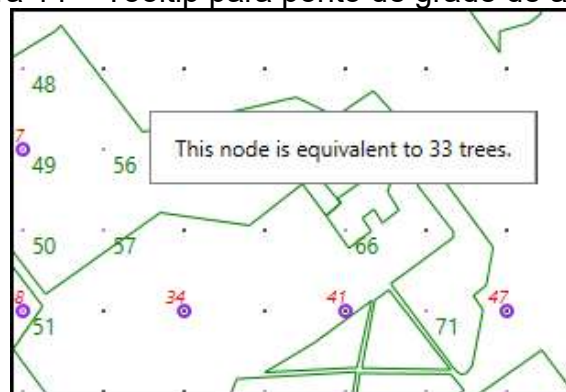
Quando a janela de simulação é exibida, todas as atividades já estão quantificadas pelo *plug-in*. Essas informações são exibidas na forma de *tooltips* quando o usuário para o mouse sobre uma área de gramado (Figura 13) ou um ponto da grade que representa as árvores (Figura 14).

Figura 13 – Tooltip para áreas de gramado



Fonte: elaborado pelo autor (2025)

Figura 14 – Tooltip para ponto de grade de árvores

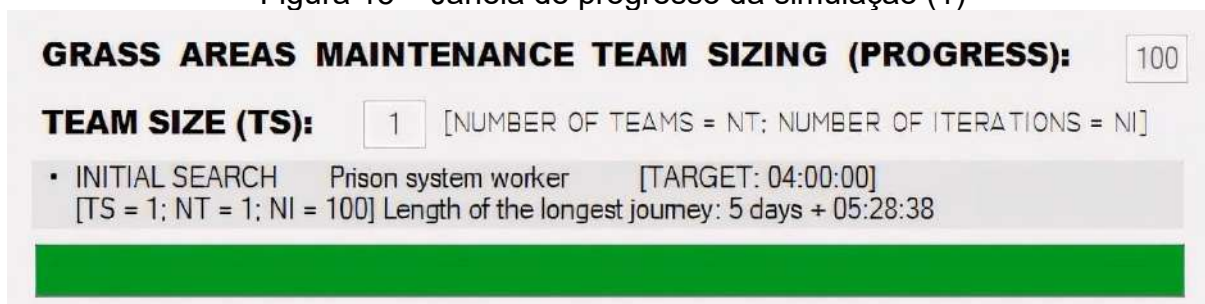


Fonte: elaborado pelo autor (2025)

4.4.2 RESULTADOS GERADOS

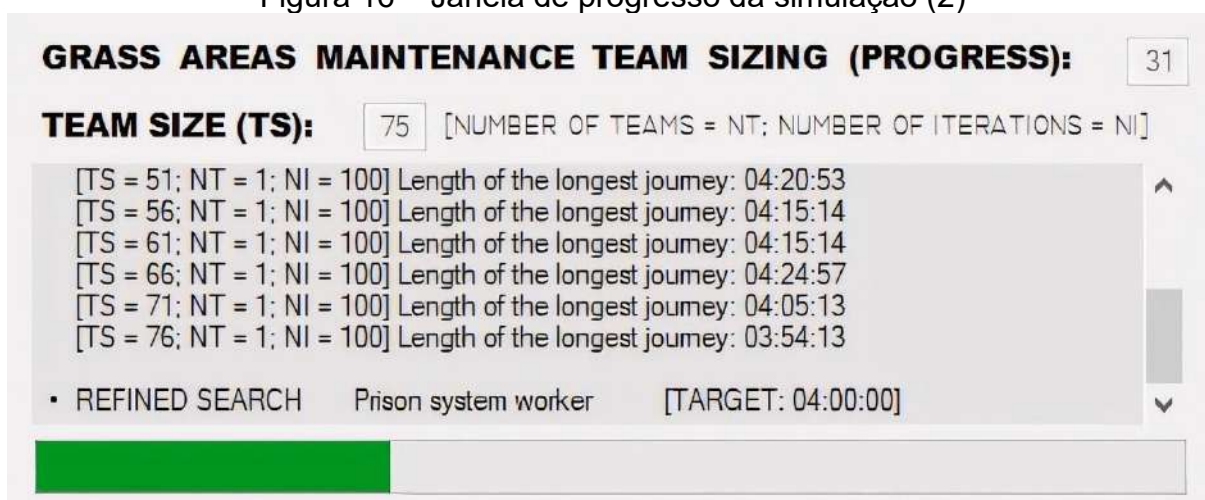
Ao clicar no botão *Run Simulation*, a simulação se inicia, numa janela auxiliar, sobreposta, que indica o progresso, e registra os passos da simulação. A simulação separa as áreas de gramado das árvores; dentro das áreas de gramado, ela diferencia os tipos de trabalhador utilizados para cada atividade, caso haja mais de um. A Figura 15 mostra o início da simulação para os trabalhadores do tipo *Prison system worker* (trabalhador do sistema prisional); a Figura 16 mostra o resultado da busca inicial.

Figura 15 – Janela de progresso da simulação (1)



Fonte: elaborado pelo autor (2025)

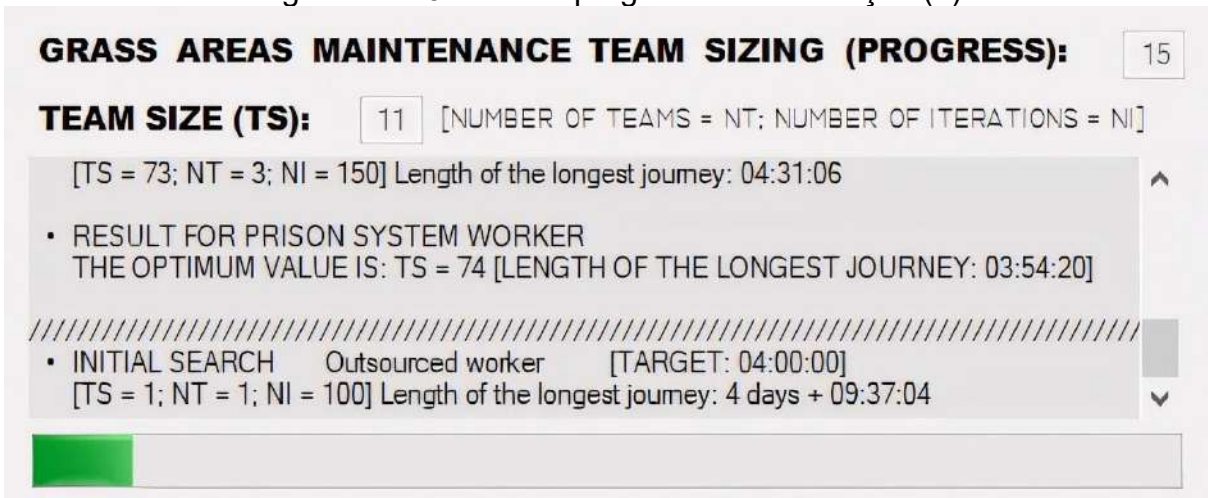
Figura 16 – Janela de progresso da simulação (2)



Fonte: elaborado pelo autor (2025)

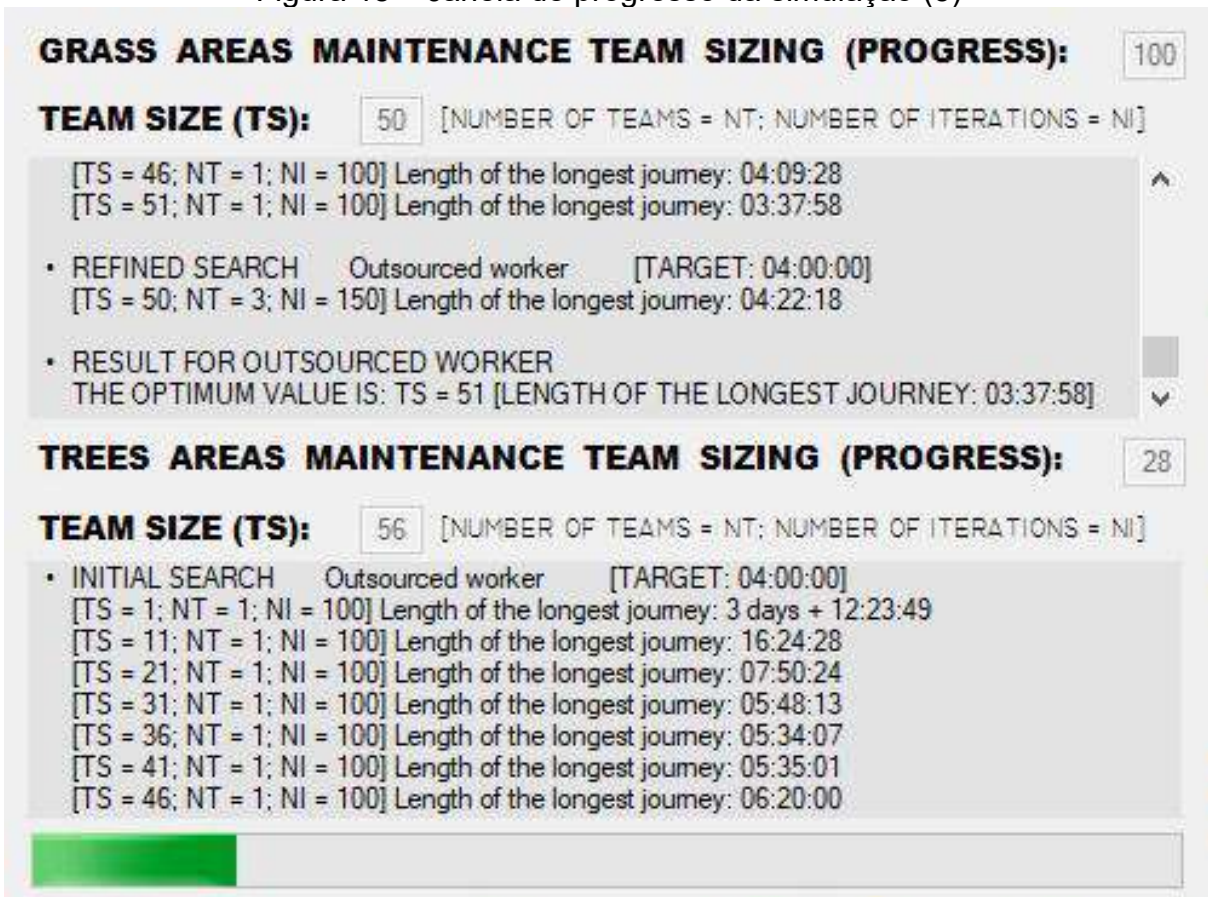
A Figura 17 mostra o resultado da simulação para os trabalhadores do tipo *Prison system worker* e o início da simulação para os trabalhadores do tipo *Outsourced worker*. A Figura 18 mostra o resultado da busca inicial, o resultado da busca refinada e o resultado final; também mostra o início da simulação (busca inicial) para árvores.

Figura 17 – Janela de progresso da simulação (4)



Fonte: elaborado pelo autor (2025)

Figura 18 – Janela de progresso da simulação (5)



Fonte: elaborado pelo autor (2025)

A Figura 19 mostra o resultado da busca inicial, o resultado da busca refinada e o resultado final para árvores.

Figura 19 – Janela de progresso da simulação (5)

GRASS AREAS MAINTENANCE TEAM SIZING (PROGRESS): 100

TEAM SIZE (TS): 50 [NUMBER OF TEAMS = NT; NUMBER OF ITERATIONS = NI]

[TS = 46; NT = 1; NI = 100] Length of the longest journey: 04:35:35
 [TS = 51; NT = 1; NI = 100] Length of the longest journey: 03:50:42

- REFINED SEARCH Outsourced worker [TARGET: 04:00:00]
 [TS = 50; NT = 3; NI = 100] Length of the longest journey: 04:37:46
- RESULT FOR OUTSOURCED WORKER
 THE OPTIMUM VALUE IS: TS = 51 [LENGTH OF THE LONGEST JOURNEY: 03:50:42]

TREES AREAS MAINTENANCE TEAM SIZING (PROGRESS): 100

TEAM SIZE (TS): 64 [NUMBER OF TEAMS = NT; NUMBER OF ITERATIONS = NI]

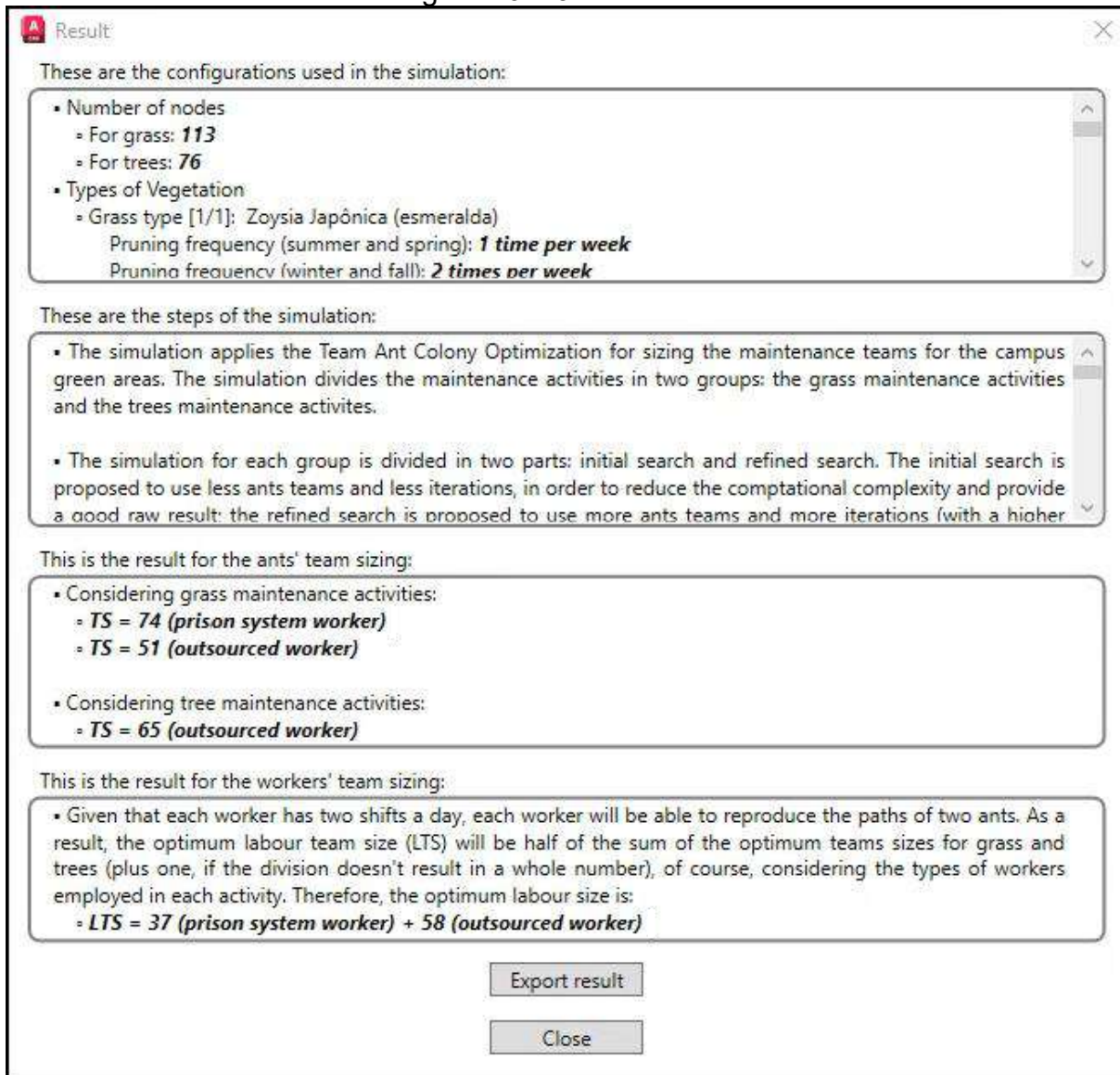
[TS = 66; NT = 1; NI = 100] Length of the longest journey: 03:38:34

- REFINED SEARCH Outsourced worker [TARGET: 04:00:00]
 [TS = 65; NT = 3; NI = 100] Length of the longest journey: 03:36:26
 [TS = 64; NT = 3; NI = 100] Length of the longest journey: 04:02:27
- RESULT OUTSOURCED WORKER
 THE OPTIMUM VALUE IS: TS = 65 [LENGTH OF THE LONGEST JOURNEY: 03:36:26]

Fonte: elaborado pelo autor (2025)

O resultado do dimensionamento do time de formigas para as árvores conclui a simulação. Automaticamente a janela auxiliar de registro de progresso e dos passos da simulação se fecha e a janela *Result* (Figura 20) é aberta. Ela está dividida em quatro regiões: as configurações da simulação; os passos da simulação; os resultados da simulação para o dimensionamento dos times de formigas; e o resultado da simulação para os times de trabalhadores.

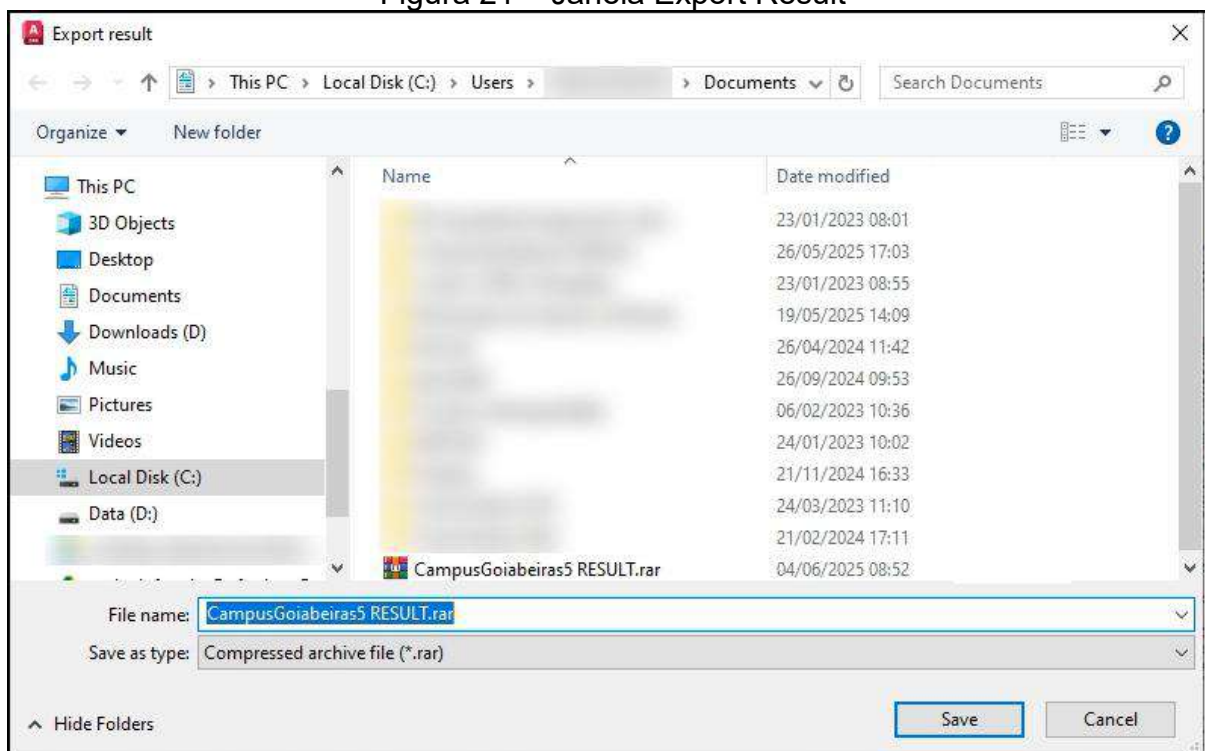
Figura 20 – Janela Result



Fonte: elaborado pelo autor (2025)

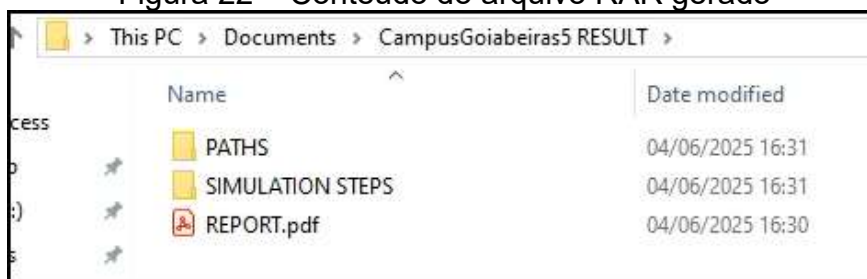
O botão *Export result* abre a janela de mesmo nome, da classe *SaveFileDialog* (Figura 21). Nela, o usuário deve definir o nome e o diretório onde o arquivo de resultado será salvo. O arquivo tem extensão RAR, e contém as pastas PATHS e SIMULATION STEPS e o arquivo REPORT.pdf (Figura 22). As pastas contidas na pasta PATHS (Figuras 23, 24 e 25) contém as imagens dos trajetos individuais de todas as formigas que fazem parte dos times apontados como resultados da simulação (para cada tipo de trabalhador, para gramados e árvores).

Figura 21 – Janela Export Result



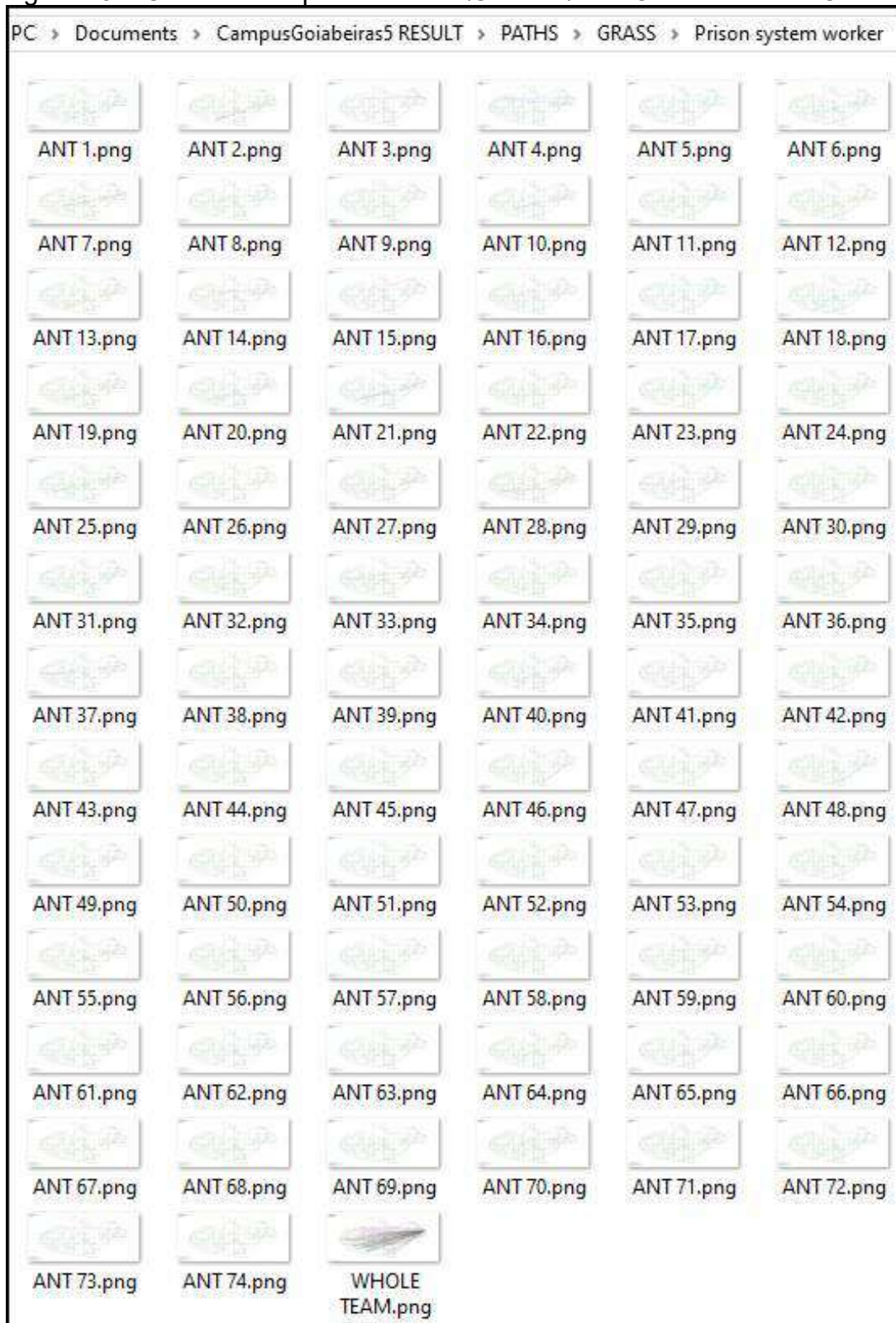
Fonte: elaborado pelo autor (2025)

Figura 22 – Conteúdo do arquivo RAR gerado



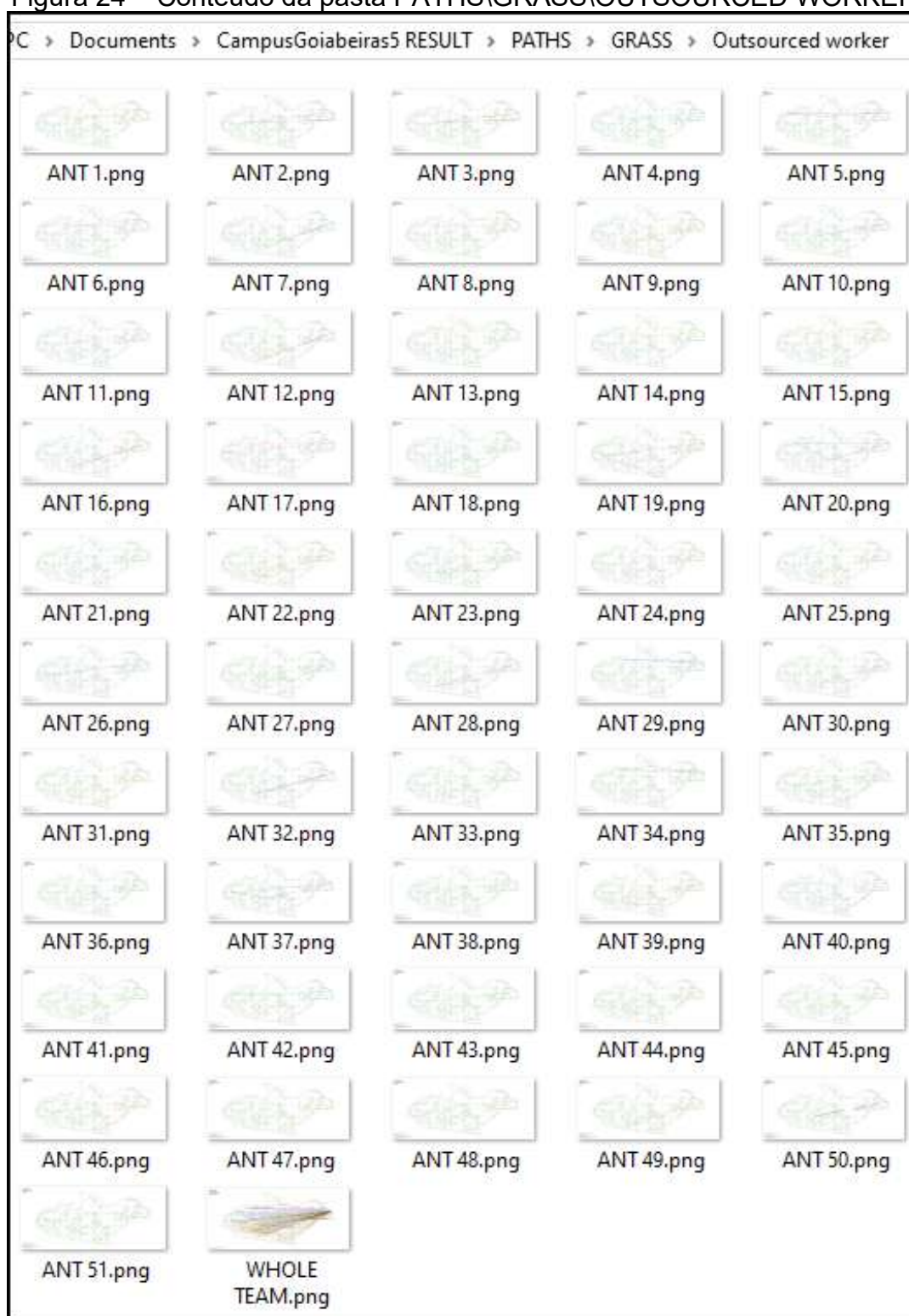
Fonte: elaborado pelo autor (2025)

Figura 23 – Conteúdo da pasta PATHS\GRASS\PRISON SYSTEM WORKER



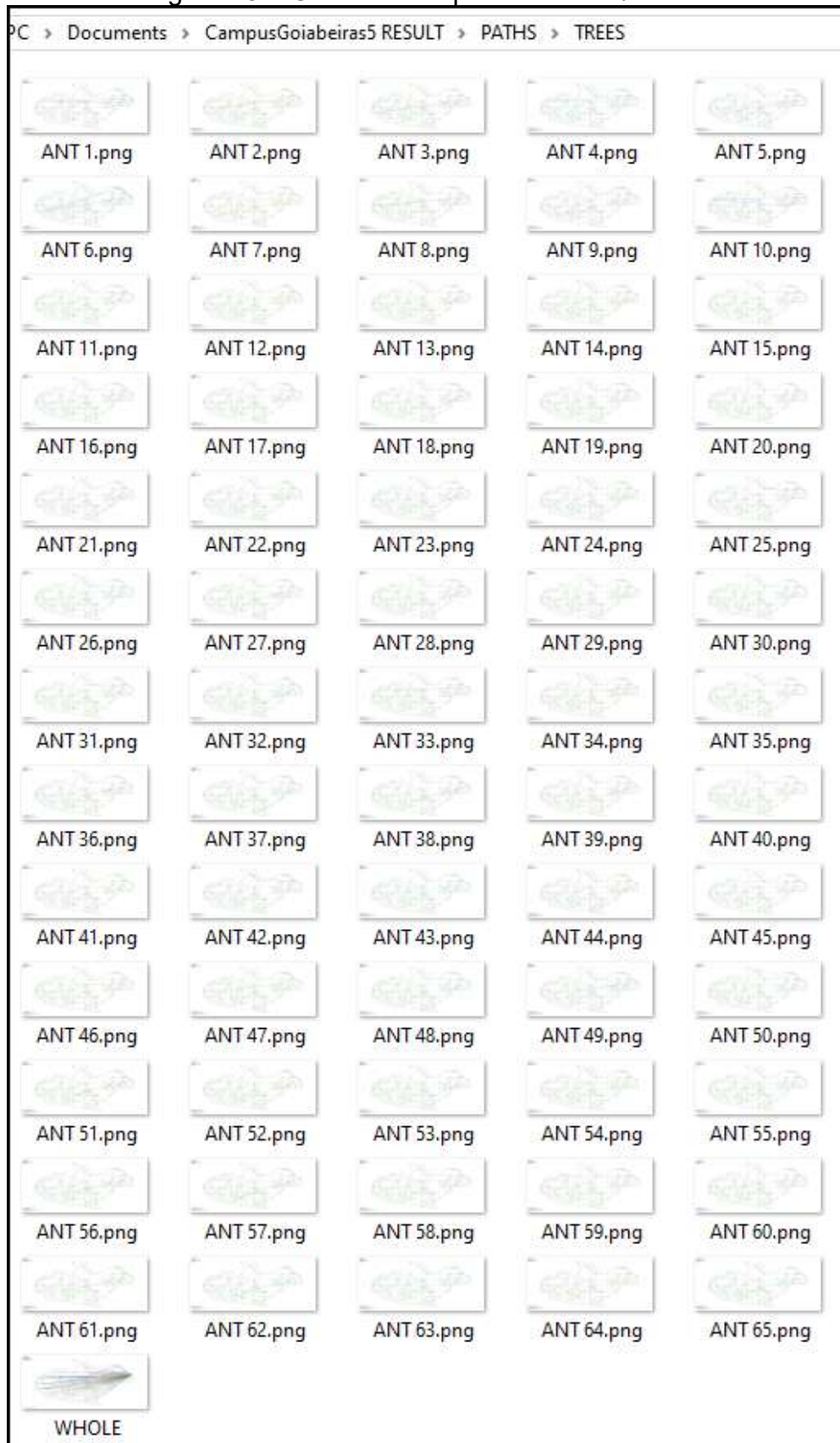
Fonte: elaborado pelo autor (2025)

Figura 24 – Conteúdo da pasta PATHS\GRASS\OUTSOURCED WORKER



Fonte: elaborado pelo autor (2025)

Figura 25 – Conteúdo da pasta PATHS\TREES



Fonte: elaborado pelo autor (2025)

As pastas contidas na pasta SIMULATION STEPS (Figuras 26, 27, 28, 29, 30 e 31) contêm todos os arquivos de texto com os relatórios de todas as etapas da simulação, separados por tipo de trabalhador e por tipo de busca (inicial e refinada).

Figura 26 – Conteúdo da pasta SIMULATION STEPS\GRASS\PRISON SYSTEM WORKER\INITIAL SEARCH

5 RESULT > SIMULATION STEPS > GRASS > Prison system worker > INITIAL SEARCH		
Name	Date modified	Type
GRASS_REPORT_TS = 1.txt	09/06/2025 12:52	Text Document
GRASS_REPORT_TS = 11.txt	09/06/2025 12:53	Text Document
GRASS_REPORT_TS = 21.txt	09/06/2025 12:53	Text Document
GRASS_REPORT_TS = 31.txt	09/06/2025 12:53	Text Document
GRASS_REPORT_TS = 41.txt	09/06/2025 12:54	Text Document
GRASS_REPORT_TS = 46.txt	09/06/2025 12:54	Text Document
GRASS_REPORT_TS = 51.txt	09/06/2025 12:55	Text Document
GRASS_REPORT_TS = 56.txt	09/06/2025 12:55	Text Document
GRASS_REPORT_TS = 61.txt	09/06/2025 12:56	Text Document
GRASS_REPORT_TS = 66.txt	09/06/2025 12:56	Text Document
GRASS_REPORT_TS = 71.txt	09/06/2025 12:57	Text Document
GRASS_REPORT_TS = 76.txt	09/06/2025 12:57	Text Document

Fonte: elaborado pelo autor (2025)

Figura 27 – Conteúdo da pasta SIMULATION STEPS\GRASS\PRISON SYSTEM WORKER\REFINED SEARCH

5 RESULT > SIMULATION STEPS > GRASS > Prison system worker > REFINED SEARCH		
Name	Date modified	Type
GRASS_REPORT_TS = 73.txt	09/06/2025 13:00	Text Document
GRASS_REPORT_TS = 74.txt	09/06/2025 12:59	Text Document
GRASS_REPORT_TS = 75.txt	09/06/2025 12:58	Text Document

Fonte: elaborado pelo autor (2025)

Figura 28 – Conteúdo da pasta SIMULATION STEPS\GRASS\OUTSOURCED WORKER\INITIAL SEARCH

beiras5 RESULT > SIMULATION STEPS > GRASS > Outsourced worker > INITIAL SEARCH		
Name	Date modified	Type
GRASS_REPORT_TS = 1.txt	09/06/2025 13:00	Text Document
GRASS_REPORT_TS = 11.txt	09/06/2025 13:00	Text Document
GRASS_REPORT_TS = 21.txt	09/06/2025 13:01	Text Document
GRASS_REPORT_TS = 31.txt	09/06/2025 13:01	Text Document
GRASS_REPORT_TS = 36.txt	09/06/2025 13:02	Text Document
GRASS_REPORT_TS = 41.txt	09/06/2025 13:02	Text Document
GRASS_REPORT_TS = 46.txt	09/06/2025 13:02	Text Document
GRASS_REPORT_TS = 51.txt	09/06/2025 13:03	Text Document

Fonte: elaborado pelo autor (2025)

Figura 29 – Conteúdo da pasta SIMULATION STEPS\GRASS\OUTSOURCED WORKER\REFINED SEARCH

beiras5 RESULT > SIMULATION STEPS > GRASS > Outsourced worker > REFINED SEARCH		
Name	Date modified	Type
GRASS_REPORT_TS = 50.txt	09/06/2025 13:04	Text Document

Fonte: elaborado pelo autor (2025)

Figura 30 – Conteúdo da PASTA SIMULATION STEPS\TREES\INITIAL SEARCH

ments > CampusGoiabeiras5 RESULT > SIMULATION STEPS > TREES > INITIAL SEARCH		
Name	Date modified	Type
TREES_REPORT_TS = 1.txt	09/06/2025 13:04	Text Document
TREES_REPORT_TS = 11.txt	09/06/2025 13:04	Text Document
TREES_REPORT_TS = 21.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 31.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 36.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 41.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 46.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 56.txt	09/06/2025 13:05	Text Document
TREES_REPORT_TS = 61.txt	09/06/2025 13:06	Text Document
TREES_REPORT_TS = 66.txt	09/06/2025 13:06	Text Document

Fonte: elaborado pelo autor (2025)

Figura 31 – Conteúdo da pasta SIMULATION STEPS\TREES\REFINED SEARCH

Name	Date modified	Type
TREES_REPORT_TS = 64.txt	09/06/2025 13:07	Text Document
TREES_REPORT_TS = 65.txt	09/06/2025 13:06	Text Document

Fonte: elaborado pelo autor (2025)

O arquivo REPORT.txt tem basicamente o mesmo conteúdo da janela *Result*, mas formatado como um relatório que pode ser facilmente anexado a um processo licitatório, por exemplo, necessitando para isso apenas ser traduzido para o português, já que todo o software foi desenvolvido em inglês. A íntegra do relatório gerado pode ser encontrada no Apêndice A da tese. A Figura 32 mostra a primeira página do relatório, e a Figura 33 mostra como o resultado é apresentado.

O relatório foi organizado em três partes complementares: configurações iniciais, detalhamento do processo e apresentação dos resultados. As configurações iniciais reúnem informações como parâmetros de vegetação, tempos de execução das atividades, dados da mão de obra e ajustes técnicos, permitindo que o gestor compreenda exatamente o cenário simulado e avalie sua coerência ou necessidade de ajustes. O detalhamento do processo descreve como a aplicação chegou aos resultados, apresentando, separadamente para gramados e árvores e para cada tipo de trabalhador, a evolução do dimensionamento ao longo das buscas inicial e refinada. Essa etapa dá transparência ao método, facilitando a verificação e validação dos números obtidos. Por fim, a apresentação dos resultados consolida os tamanhos ótimos de equipes — primeiro para os “times de formigas”, depois para os times de trabalhadores —, fornecendo dados prontos para apoiar decisões de planejamento e a elaboração de orçamentos em processos licitatórios, alinhando a clareza técnica à aplicação prática na gestão de recursos.

Figura 32 – Primeira página do relatório gerado pelo software

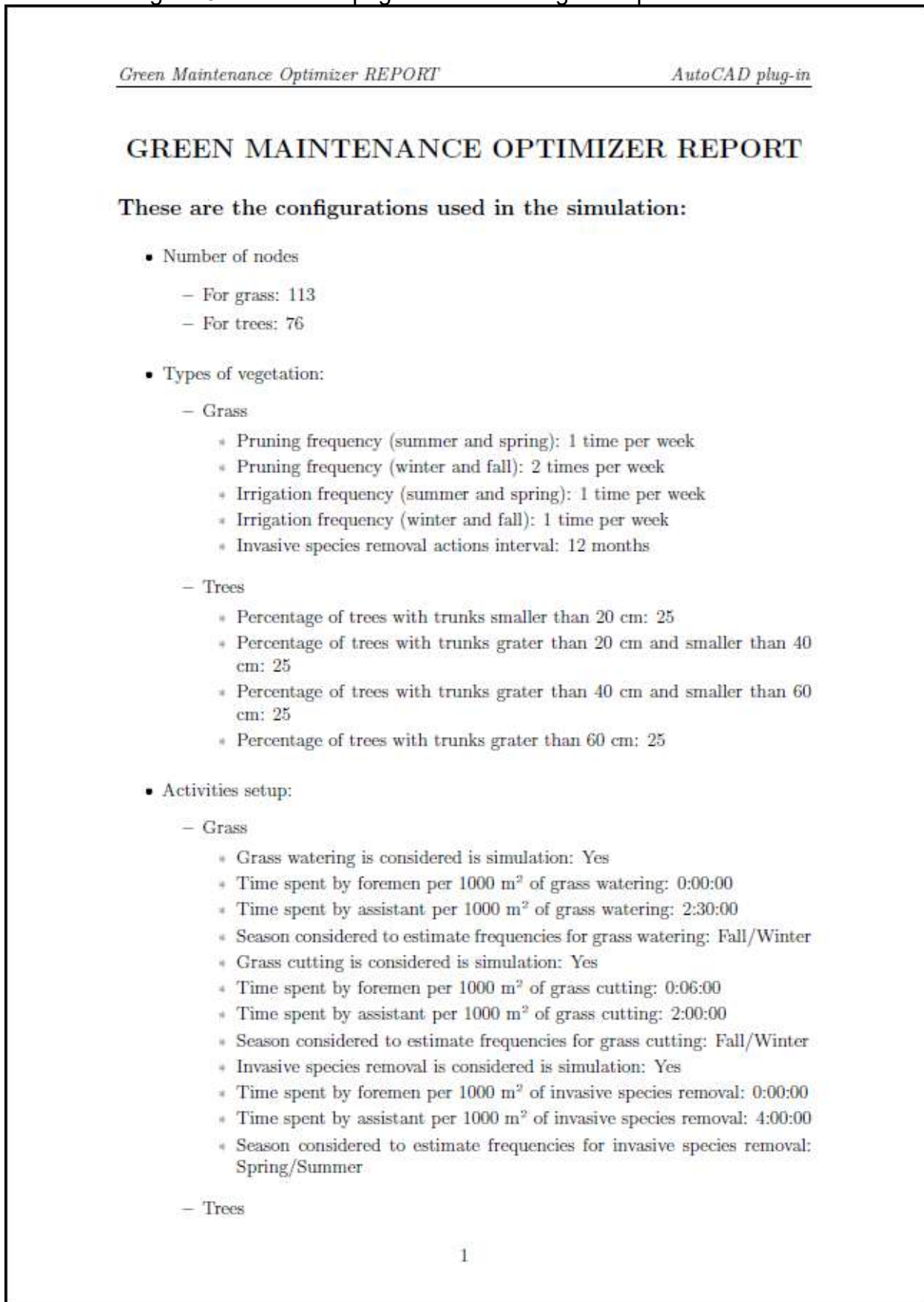
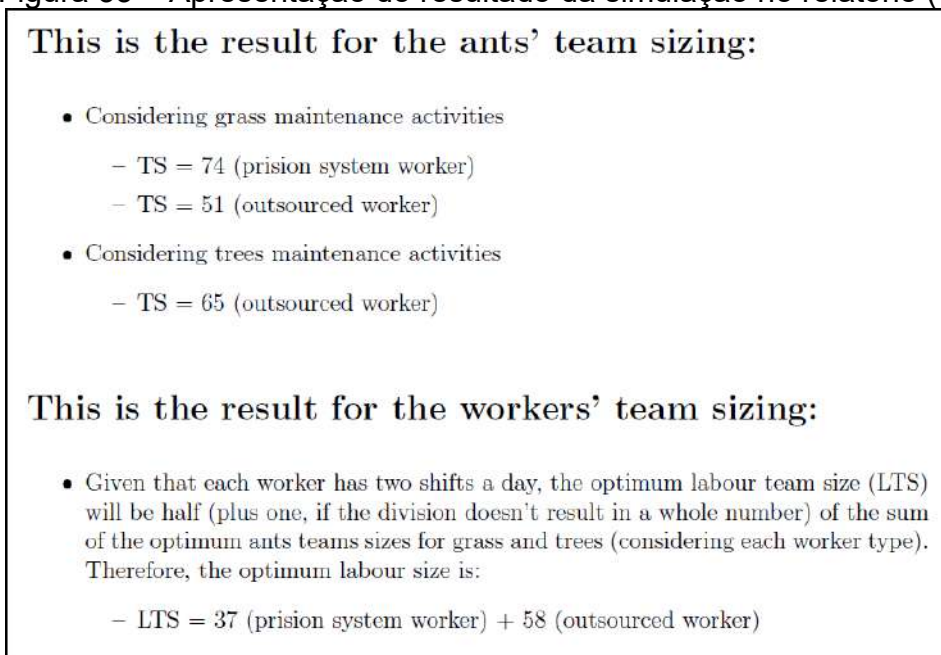


Figura 33 – Apresentação do resultado da simulação no relatório (1)

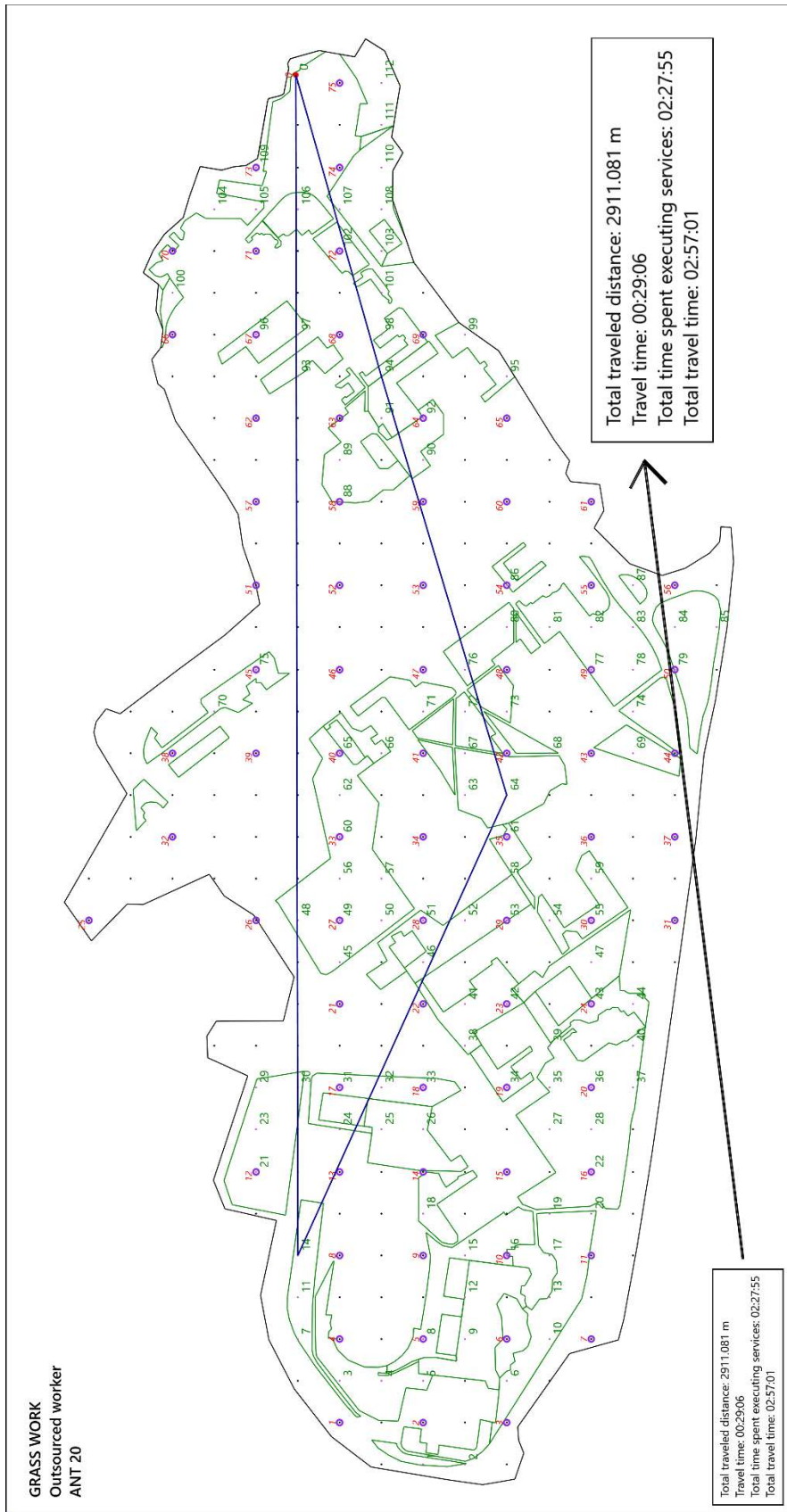


Fonte: elaborado pelo autor (2025)

Todas as imagens salvas nas pastas contidas na pasta PATHS exibem as informações contidas no arquivo TXT referente ao tamanho final do time encontrado pelo *plug-in* para a formiga a que dizem respeito (como exemplificado pelas Figuras 34 e 35)¹⁰. Ou seja, o arquivo RAR gerado pelo *plug-in* oferece um relatório completo e detalhado de todo o processo de dimensionamento realizado, assim como um registro detalhado do resultado. Como exemplo, a Figura 34 mostra o trajeto da formiga 20, e a Figura 35 mostra a parte que se refere à formiga 20 do arquivo GRASS_REPORT_TS = 51.txt, que contém o detalhamento da simulação para o tamanho de time que foi apontado como a resposta para o dimensionamento. Pode-se verificar que os dados da imagem e do arquivo são os mesmos.

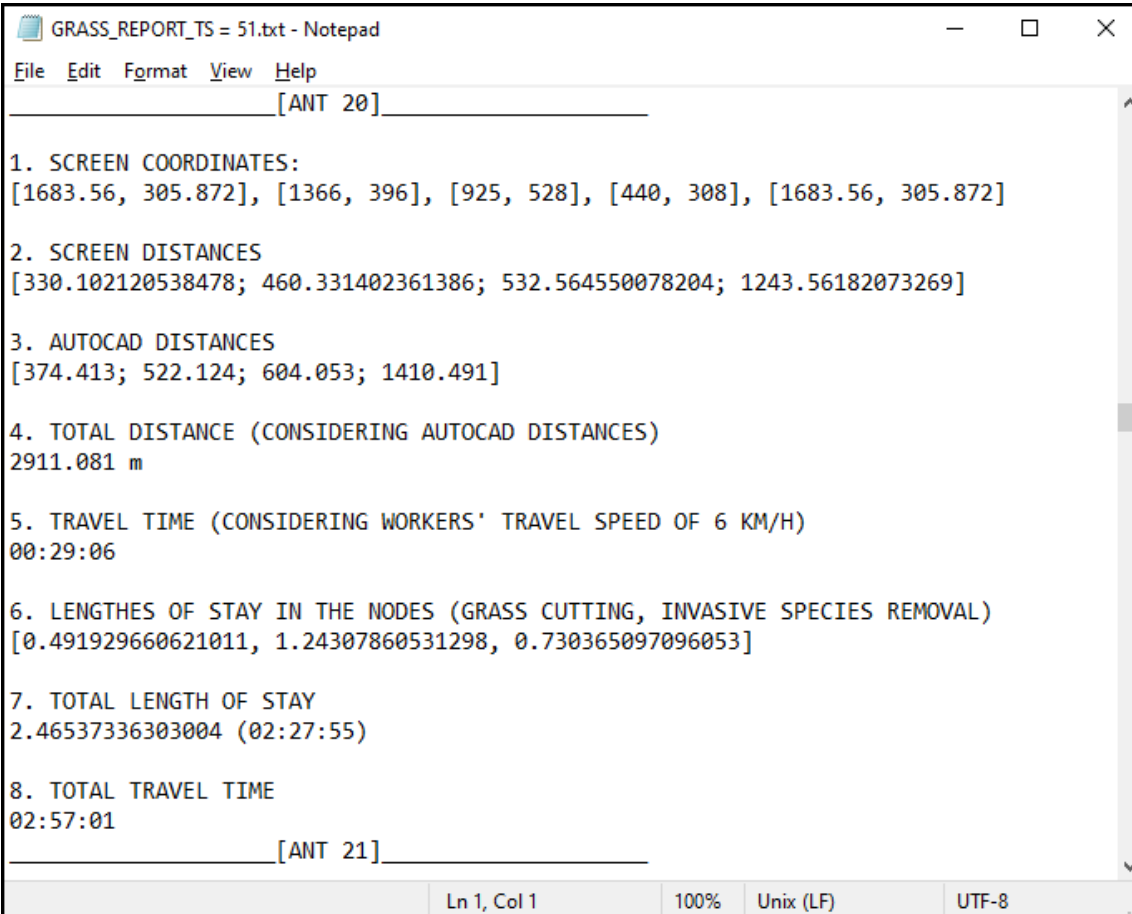
¹⁰ O arquivo TXT referente ao tamanho do time encontrado pelo *plug-in* contém as informações detalhadas sobre os trajetos de cada formiga (incluindo distância total percorrida, tempos e permanência nos nós, tempo total de trajeto). A partir das imagens individuais salvas em cada uma das pastas, é possível fazer uma relação dos dados de cada formiga com seu trajeto no campus, permitindo visualizar de forma clara como cada rota contribui para o resultado final da simulação.

Figura 34 – Imagem do arquivo ANT 20.png, localizado na pasta PATHS\GRASS\OUTSOURCED WORKER



Fonte: elaborado pelo autor (2025)

Figura 35 – Imagem do arquivo ANT 20.png, localizado na pasta PATHS\GRASS\OUTSOURCED WORKER



```

GRASS_REPORT_TS = 51.txt - Notepad
File Edit Format View Help
[ANT 20]
1. SCREEN COORDINATES:
[1683.56, 305.872], [1366, 396], [925, 528], [440, 308], [1683.56, 305.872]
2. SCREEN DISTANCES
[330.102120538478; 460.331402361386; 532.564550078204; 1243.56182073269]
3. AUTOCAD DISTANCES
[374.413; 522.124; 604.053; 1410.491]
4. TOTAL DISTANCE (CONSIDERING AUTOCAD DISTANCES)
2911.081 m
5. TRAVEL TIME (CONSIDERING WORKERS' TRAVEL SPEED OF 6 KM/H)
00:29:06
6. LENGTHES OF STAY IN THE NODES (GRASS CUTTING, INVASIVE SPECIES REMOVAL)
[0.491929660621011, 1.24307860531298, 0.730365097096053]
7. TOTAL LENGTH OF STAY
2.46537336303004 (02:27:55)
8. TOTAL TRAVEL TIME
02:57:01
[ANT 21]
Ln 1, Col 1 100% Unix (LF) UTF-8

```

Fonte: elaborado pelo autor (2025)

4.4.3 COMPARAÇÃO ENTRE CENÁRIOS

Os dados geográficos – de localização dos gramados, árvores, origem dos trabalhadores – são características do campus, e não faz sentido que sejam alterados de uma simulação para outra. Os dados de consumo de mão de obra por unidade de serviço, seja em gramados ou árvores, não são alterados porque constituem dados históricos consolidados, retirados de manuais da área, que orientam a quantificação dos serviços. No entanto, dados de frequências das atividades, jornadas de trabalho dos trabalhadores, utilização dos mesmos trabalhadores ou trabalhadores distintos para diferentes atividades, consideração ou não de serviços, permitem a simulação de diferentes cenários.

As imagens das seções anteriores mostram o cenário de uma simulação com as seguintes características:

- serviços de rega de gramados, poda de gramados e remoção de espécies invasoras consideradas;
- trabalhadores do sistema prisional encarregados do serviço de rega de gramados; trabalhadores terceirizados responsáveis pelos serviços de poda de gramados, remoção de espécies invasoras e poda em árvores;
- trabalhadores com dois turnos de trabalho de 4 horas por dia;
- sem período de descanso por turno;
- velocidade de deslocamento dos trabalhadores de 6 km/h;
- serviços de poda e rega em gramados realizados uma vez por semana (simulação considerada para estações de verão ou primavera);
- intervalo entre a realização do serviço de remoção de espécies invasoras de 12 meses (resultando numa frequência de 1 vez/ano);
- frequência de podas em árvores: 1 vez ao ano;
- distribuição das árvores por faixa de tamanho de tronco de: 25% (até 20 cm), 25% (entre 20cm e 40 cm), 25% (entre 40 cm e 60 cm) e 25% (acima de 60 cm);

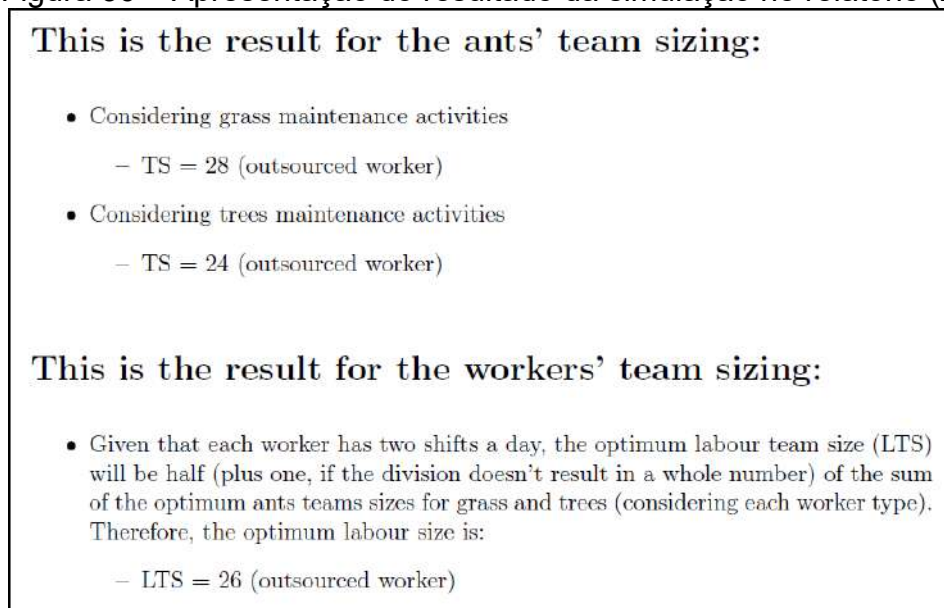
A Figura 34 mostra o resultado de uma simulação em que foram consideradas três alterações principais em relação ao cenário anterior:

1. Apenas o serviço de poda de gramados foi considerado; incluiu-se um período de descanso de 30 minutos por turno de trabalho.
2. A frequência da poda de gramados foi reduzida para uma vez por mês; a poda de árvores passou a ocorrer a cada dois anos.
3. A distribuição das árvores por faixa de diâmetro médio de tronco foi ajustada para 40% (até 20 cm), 25% (entre 20 e 40 cm), 25% (entre 40 e 60 cm) e 10% (acima de 60 cm).

Essas alterações irão impactar diretamente na simulação, de diferentes formas. A primeira alteração reduz a duração do turno que pode ser preenchida pela simulação com deslocamentos ou realização de atividades. A redução das frequências das atividades de poda de gramados e árvores irá reduzir as cargas de trabalho totais consideradas, e conseqüentemente os custos de permanência considerados na simulação, para ambos os modelos. A alteração da distribuição das árvores por faixa

de diâmetro médio de tronco (com uma quantidade maior de árvores de pequenos diâmetros e uma quantidade menor de árvores de maiores diâmetros) irá reduzir as cargas de trabalho consideradas para as árvores na simulação, já que os consumos de mão-de-obra por unidade são menores para pequenos diâmetros e maiores para grandes diâmetros, no caso das árvores.

Figura 36 – Apresentação do resultado da simulação no relatório (2)



Fonte: elaborado pelo autor (2025)

A comparação entre as Figuras 33 e 36 mostra como as mudanças nas configurações permitem simular cenários distintos, de tal modo que o resultado da simulação será sempre função dessas configurações. Por isso o relatório gerado é tão importante, uma vez que, além do resultado da simulação, traz todos os dados de entrada. As íntegras de ambos os relatórios podem ser encontradas nos Apêndices A e B.

5 CONSIDERAÇÕES FINAIS

O percurso da pesquisa iniciou-se pela delimitação conceitual do campo da computação inspirada na natureza, identificando seus princípios e a forma como diferentes fenômenos biológicos têm servido de base para o desenvolvimento de algoritmos de otimização. A partir desse panorama, aprofundou-se o estudo da inteligência de enxames, destacando os mecanismos de auto-organização e cooperação observados em colônias de insetos sociais, e posteriormente da ACO, que simula o comportamento social das formigas, e se mostra extremamente eficiente na solução do TSP.

Partindo desse ponto, o desenvolvimento desta tese permitiu demonstrar a viabilidade da aplicação da meta-heurística da ACO ao problema do dimensionamento de equipes de manutenção de áreas verdes em campi universitários, por meio da formulação matemática de uma instância do MTSP e da implementação de um *plug-in* funcional integrado ao ambiente do AutoCAD. A pesquisa resultou na construção de uma ferramenta científica capaz de processar dados reais de um campus universitário e gerar soluções adaptadas a diferentes cenários de manutenção, confirmando que a inteligência de enxames pode oferecer suporte relevante à gestão de recursos humanos e operacionais na área da arquitetura da paisagem.

O software produzido, de código aberto, constitui-se em uma contribuição concreta desta pesquisa, ao reunir em um mesmo ambiente de modelagem computacional a representação gráfica do problema, a aplicação de algoritmos de otimização e a geração de relatórios de simulação. Essa integração demonstra o potencial da abordagem. Já a documentação das funcionalidades e a disponibilização integral do software, tanto em código-fonte quanto em versão executável, trazem consigo a expectativa de contribuir para a continuidade de investigações acadêmicas e aplicações práticas em contextos semelhantes.

Embora concebido para a gestão de equipes, o *plug-in* apresenta características que o qualificam também como instrumento de apoio ao planejamento estratégico, uma vez que permite estimar previamente os recursos humanos necessários à manutenção de áreas projetadas. Assim, pode ser empregado na concepção de novos

campi universitários ou na readequação paisagística de espaços existentes, bem como em áreas públicas extensas, como parques e praças, que compartilham desafios similares de manutenção.

Como justificado na metodologia, a realização de testes de usabilidade não foi incluída, pois estava fora do escopo da pesquisa. Entre os condicionantes que orientaram a modelagem adotada, destacam-se: a ausência de integração em tempo real com sistemas institucionais de gestão de recursos; a necessidade de simplificações na representação das atividades de manutenção (num cenário de ausência de dados decorrentes de medições reais, em razão dessa ausência de integração com a gestão de recursos); e a restrição da aplicação da modelagem a um único estudo de caso. Mesmo diante dessas condicionantes, os resultados alcançados evidenciam que o modelo proposto é robusto, flexível e capaz de fornecer soluções consistentes para diferentes configurações de cenários de manutenção.

Como perspectivas futuras, recomendam-se:

- *ampliação da base de dados*: incluir múltiplos *campi* universitários com características físicas e operacionais distintas, permitindo maior diversidade de cenários e robustez do modelo;
- *testes formais de usabilidade*: avaliar a interface do *plug-in* com diferentes perfis de usuários, buscando maior aderência e facilidade de operação;
- *integração com plataformas BIM e GIS*: expandir a aplicação do *plug-in* em sistemas de gestão de *facilities*, modelagem de informações da construção e georreferenciamento, potencializando o uso em larga escala e a interoperabilidade com outras ferramentas de planejamento;
- *aplicação empírica e comparação com dados reais*: realizar testes em situações reais de manutenção, coletando dados operacionais para validar e aprimorar o modelo, ajustando parâmetros do algoritmo com base em resultados observados;
- *aprofundamento da análise de desempenho*: avaliar o comportamento do algoritmo em instâncias maiores e mais complexas, explorando plenamente a capacidade adaptativa da ACO e verificando limites de escalabilidade do modelo;

- *generalização do modelo*: adaptar o modelo a fim de que possa ser aplicado a outros tipos de paisagens urbanas.

Em síntese, esta tese cumpriu o objetivo de explorar o potencial da inteligência de enxames aplicada ao campo do paisagismo na arquitetura (representado pela manutenção de áreas verdes em universidades), contribuindo tanto no plano teórico — ao propor e validar um modelo baseado em ACO para o MTSP — quanto no plano prático, ao disponibilizar um software pronto para uso. Os resultados alcançados indicam que a abordagem tem relevância acadêmica e aplicabilidade prática, abrindo caminho para pesquisas futuras que possam expandir, consolidar e transferir esse conhecimento para a gestão efetiva das áreas verdes e dos recursos humanos em instituições de ensino superior e em outros contextos urbanos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

1. ABBASPOUR, K. C.; SCHULIN, R.; GENUCHTEN, M. T. van. Estimating unsaturated soil hydraulic parameters using ant colony optimization. *Advances in Water Resources*, [s.l.], v. 24, n. 8, p. 827-841, 2001. DOI: [https://doi.org/10.1016/S0309-1708\(01\)00018-5](https://doi.org/10.1016/S0309-1708(01)00018-5).
2. ARAÚJO, Luan Henrique Barbosa de; NÓBREGA, Camila Costa da; SILVA, Ane Cristine Fortes da; VIEIRA, Fábio de Almeida. Análise quali-quantitativa da arborização da Praça Pedro Velho, Natal, RN. *Agropecuária Científica no Semiárido*, Patos, v. 11, n. 1, p. 65–71, jan./mar. 2015.
3. ARAÚJO, Paulo Sérgio Fagundes; ONOFRE, Carlos Eduardo Lins; MACEDO, Bruno Rafael Moraes de. Plano de arborização do campus universitário central da Universidade Federal do Rio Grande do Norte. Natal: Universidade Federal do Rio Grande do Norte, Superintendência de Infraestrutura, 2016. 34 p.
4. ASHENA, R.; MOGHADASI, J. Bottom hole pressure estimation using evolved neural networks by real coded ant colony optimization and genetic algorithm. *Journal of Petroleum Science and Engineering*, v. 77, n. 3–4, p. 375–385, 2011. DOI: <https://doi.org/10.1016/j.petrol.2011.04.015>.
5. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5462: Confiabilidade e manutenibilidade. Rio de Janeiro, 1994. 37 p.
6. ASSOCIAÇÃO NACIONAL GRAMA LEGAL. Grama cultivada em áreas de segurança de aeroportos. Piracicaba: [s.n.], 2017a. 8 p.
7. ASSOCIAÇÃO NACIONAL GRAMA LEGAL. Guia: implantação e manejo de gramados na jardinagem. Piracicaba: [s.n.], 2023. 17 p.
8. ASSOCIAÇÃO NACIONAL GRAMA LEGAL. Manual de manutenção de áreas gramadas em locais públicos. Piracicaba: [s.n.], 2017b. 12 p.
9. AUTODESK. API Developer's Guide: AutoCAD Civil 3D 2013. [s.l.], 2012. 410 p.
10. AUTODESK. AutoCAD: create solutions and enhance core software functionality. [s.l.], 2023. Disponível em: <https://aps.autodesk.com/developer/overview/autocad>. Acesso em: 02 out. 2023.
11. BARBOSA, R. M.; SILVA JR., L. J.; KASHIWABARA, A. Y. Application of ant colony optimization for workforce allocation in geographically distributed services. *Journal of Heuristics*, v. 21, n. 3, p. 325–350, 2015.
12. BAR-YAM, Yaneer. Reviews (book & software): complex systems with Herbert Simon. *Complexity*, [s.l.]: John Wiley & Sons, Inc., v. 3, n. 5, p. 47-48, 1998.
13. BECKER, Leonardo Ranft. RELATÓRIO DE ESTÁGIO CURRICULAR OBRIGATÓRIO SUPERVISIONADO: green grass, o tapete de grama. Porto Alegre: [s.n.], 2012. 45 p.

14. BENI, Gerardo; WANG, Jing. Swarm Intelligence in Cellular Robotic Systems. In: NATO ADVANCED WORKSHOP ON ROBOTS AND BIOLOGICAL SYSTEMS, 1., 1989, Toscana, Itália. Anais [...]. Heidelberg: Springer, 1993. p. 703-712.
15. BONABEAU, Eric; DORIGO, Marco; THERAULAZ, Guy. Swarm Intelligence: from natural to artificial systems. Nova York: Oxford University Press, 1999. 307 p.
16. BRASIL. Caixa Econômica Federal. SINAPI – Sistema Nacional de Pesquisa de Custos e Índices da Construção Civil: Cadernos técnicos de composições para paisagismo. Brasília, 2023. 73 p.
17. CGP - CONSULTORIA GERENCIAMENTO E PLANEJAMENTO LTDA. Projeto Paisagístico. Belo Horizonte: Prefeitura Municipal de Belo Horizonte, 2022. 21 p.
18. CHANGDAR, Chiranjit; MONDAL, Moumita; GIRI, Pravash Kumar; NANDI, Utpal; PAL, Rajat Kumar. A two-phase ant colony optimization based approach for single depot multiple travelling salesman problem in Type-2 fuzzy environment. *Artificial Intelligence Review*, v. 56, p. 965-993, 2023. DOI: <https://doi.org/10.1007/s10462-022-10190-9>
19. CHRISTODOULOU, S. E. Scheduling construction projects using ant colony optimization. *Journal of Computing in Civil Engineering*, v. 23, n. 4, p. 373–381, 2009.
20. CHRISTODOULOU, Symeon E.; ELLINAS, Georgios. Pipe routing through ant colony optimization. *Journal of Infrastructure Systems*, v. 16, n. 2, p. 149-159, June 2010. DOI: 10.1061/(ASCE)1076-0342(2010)16:2(149).
21. CHRISTODOULOU, Symeon. Scheduling resource-constrained projects with ant colony optimization artificial agents. *Journal Of Computing In Civil Engineering*, [s.l.], v. 1, n. 24, p. 45-55, fev. 2010.
22. COSTA, Yasel J.; LEDÓN, René Abreu; MACHADO, Norge Isaias Coello; NOWE, Ann. Multi-type ant colony system for solving the multiple traveling salesman problem. *Revista Técnica de la Facultad de Ingeniería, Universidad del Zulia*, v. 35, n. 3, p. 248-256, Dec. 2012.
23. COSTA, Giovani Cecatto Lopes Ribeiro da; FIGUEIREDO, Sílvia Hauelsen; RIBEIRO, Sidnea Eliane Campos. Estudo comparativo da tecnologia CAD com a tecnologia BIM. *Revista de Ensino de Engenharia*, [s.l.], v. 34, n. 2, p. 11-18, 2015.
24. COX, Carlos Henrique Harper. Orçamento estimativo nas licitações e contratações diretas realizadas pela administração pública. *Revista Eletrônica Jurídico-Institucional*, [s.l.], v. 1, n. 12, p. 1-49, 2017.
25. DENEUBOURG, J. L.; PASTEELS, Jacques. M.; VERHAEGHE, J. C. Probabilistic behaviour in ants: a strategy of errors? *Journal of Theoretical Biology*, v. 105, n. 2, p. 259–271, 1983.

26. DENEUBOURG, Jean-Louis; ARON, Serge; GOSS, S; PASTEELS, Jacques. The Self-Organizing Exploratory Pattern of the Argentine Ant. *Journal of Insect Behavior*, [s.l.], v. 3, n. 2, p. 159-168, 1990.
27. DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES. Manual de custos rodoviários: composições de custos unitários de referência - serviços de conservação rodoviária. 3. ed. Rio de Janeiro: DNIT, 2003. 224 p.
28. DORIGO, Marco; DI CARO, Gianni. Ant Colony Optimization: a new meta-heuristic. In: *Conference Evolutionary Computation*, 99, 1999, Nova York. *Anais...*, Nova York: IEEE, julho de 1999, p. 1470-1477.
29. DORIGO, Marco; GAMBARDELLA, Luca Maria. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transaction On Evolutionary Computation*, [s.l.], v. 1, n. 1, p. 53-66, 1997. DOI: <https://doi.org/10.1109/4235.585892>.
30. DORIGO, Marco; MANIEZZO, Vittorio; COLORNI, Alberto. The Ant System: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, v. 26, n. 1, p. 29–41, 1996.
31. DORIGO, Marco; STÜTZLE, Thomas. *Ant Colony Optimization*. [s.l.]: MIT Press, 2004. 305 p.
32. DORIGO, Marco; STÜTZLE, Thomas. *Ant Colony Optimization: overview and recent advances*. Bruxelas: IRIDIA, 2009. 32 p.
33. EMBRAPA. Estratégias para adequação ambiental de propriedades rurais: implantação e manutenção. Araputanga: Ministério da Agricultura, Pecuária e Abastecimento, 2012. 94 slides, color.
34. ENEL. *GUIA DE PODAS DE ÁRVORES URBANAS: a energia em harmonia com a natureza*. São Paulo, 2020.
35. FARISI, Olief Ilmandira Ratu; SETIYONO, Budi; DANANDJOJO, R. Imbang. A hybrid approach to multi-depot multiple traveling salesman problem based on firefly algorithm and ant colony optimization. *IAES International Journal of Artificial Intelligence (IJ-AI)*, v. 10, n. 4, p. 910-918, Dec. 2021. DOI: <https://doi.org/10.11591/ijai.v10.i4.pp910-918>
36. FERNANDES, Jorge. *Projeto paisagístico: campus universitário de Goiabeiras*. Vitória: [s.n.], 1997. 157 p.
37. GENDREAU, Michel; POTVIN, Jean-Yves. *Handbook of Metaheuristics*. 2. ed. Nova York: Springer Science, 2010. 648 p.
38. GHAFURIAN, Soheil; JAVADIAN, Nikbakhsh. An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. *Applied Soft Computing*, v. 11, p. 1256-1262, 2011. DOI: <https://doi.org/10.1016/j.asoc.2010.03.002>
39. GIACOIA NETO, José. Formas práticas de manejo da irrigação. In: *SIMPÓSIO SOBRE GRAMADOS*, 2., 2004, Botucatu. *Anais...* Botucatu: [S.N.], 2004. p. 1-17.

40. GIULIETTI, Ana Maria et al. Guia das Espécies Invasoras e outras que requerem manejo e controle no S11D, Floresta Nacional de Carajás, Pará. [s.l.]: Instituto Tecnológico Vale, 2018. 160 p.
41. GONCALVES, Rafael. Gramados Perfeitos: guia completo. 2020. Disponível em: <https://projetojardinando.com.br/gramas-perfeitas-guia-completo/>. Acesso em: 25 jun. 2024.
42. GOSS, S.; BECKERS, R.; DENEUBOURG, Jean-Louis; ARON, Serge; PASTEELS, Jacques. How trail laying and trail following can solve foraging problems for ant colonies. In: Behavioural Mechanisms of Food Selection, 1990, [s.l.]. Anais... [s.l.]: Springer, 1990, p. 661-678.
43. GRASSÉ, P.-P. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp.: la théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs. In: Insectes Sociaux, Paris, v. 6, n. 1, p. 41-80, mar. 1959. DOI: <https://doi.org/10.1007/BF02223791>
44. HAKEN, Hermann. Synergetics: an introduction. 3. ed. Berlim: Springer, 1983.
45. HINGRAJIYA, Krishna H.; GUPTA, Ravindra Kumar; CHANDEL, Gajendra Singh. An approach for solving multiple travelling salesman problem using ant colony optimization. Computer Engineering and Intelligent Systems, v. 6, n. 2, p. 13-19, 2015.
46. HOFSTADTER, Douglas. GÖDEL, ESCHER, BACH: AN ETERNAL GOLDEN BRAIN: a metaphorical fugue on minds and machines in the spirit of Lewis Carroll. Nova York: Basic Books Inc., abril de 1979. 788 p.
47. ISIK, Senem; ATKIN, Michael. Tackling the traveling salesman problem with graph neural networks. Medium (Stanford CS224W), 2023. Disponível em: <https://medium.com/stanford-cs224w/tackling-the-traveling-salesman-problem-with-graph-neural-networks-b86ef4300c6e>. Acesso em: 6 ago. 2025.
48. JOHNSON, Steven. Emergence: the connected lives of ants, brains, cities, and software. E-book. New York: Scribner, 2012.
49. JUNJIE, Pan; DINGWEI, Wang. An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. In: First International Conference on Innovative Computing, Information and Control (ICICIC), 2006, Beijing, China. Proceedings... ICICIC 2006. Los Alamitos, CA: IEEE, 2006. p. 210-213. DOI: <https://doi.org/10.1109/ICICIC.2006.40>.
50. KAVEH, A.; HASSANI, B.; SHOJAEI, S.; TAVAKKOLI, S. M. Structural topology optimization using ant colony methodology. Engineering Structures, [s.l.], v. 30, n. 9, p. 2559-2565, 2008. DOI: <https://doi.org/10.1016/j.engstruct.2008.02.012>.
51. KAVEH, A.; JAHANSHAH, M. Plastic limit analysis of frames using ant colony systems. Computers and Structures, [s.l.], v. 86, n. 11-12, p. 1152-1163, 2008. DOI: <https://doi.org/10.1016/j.compstruc.2008.01.001>.

52. KAVEH, A.; SHAHROUZI, M. Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization. *International Journal for Numerical Methods in Engineering*, [s.l.], v. 73, n. 4, p. 544-563, 2008. DOI: <https://doi.org/10.1002/nme.2088>.
53. KAVEH, A.; SHOJAEE, S. Optimal design of skeletal structures using ant colony optimization. *International Journal for Numerical Methods in Engineering*, [s.l.], v. 70, n. 5, p. 563-581, 2007. DOI: <https://doi.org/10.1002/nme.1898>.
54. KAVEH, A.; TALATAHARI, S. A hybrid particle swarm and ant colony optimization for design of truss structures. *Asian Journal of Civil Engineering (Building and Housing)*, [s.l.], v. 9, n. 4, p. 329-348, 2008. Disponível em: <https://www.researchgate.net/publication/228637254>. Acesso em: 18 ago. 2025.
55. KAVEH, A.; TALATAHARI, S. An improved ant colony optimization for constrained engineering design problems. *Engineering Computations: International Journal For Computer-Aided Engineering And Software*, [s.l.], v. 27, n. 1, p. 155-182, 2010.
56. KELLER, Evelyn F.; SEGEL, Lee A. Initiation of slime mold aggregation viewed as an instability. *Journal Of Theoretical Biology*, [S.l.], v. 26, n. 3, p. 399-415, mar. 1970.
57. KENNEDY, James; EBERHART, Russel; SHI, Yuhui. *Swarm Intelligence*. São Francisco: Morgan Kaufmann Publishers, 2001. 512 p.
58. KELLY, Diane. Scientific software engineering. *IEEE Software*, v. 24, n. 4, p. 68-78, 2007.
59. LAMPORT, Leslie. *LaTeX: a document preparation system*. 2. ed. Boston: Addison-Wesley, 1994.
60. LANGTON, Christopher. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, [s.l.]: Elsevier, vol. 22, n. 1-3, p. 120-149, 1986.
61. LEE, Dongoun; KIM, Seungho; KIM, Sangyong. Development of Hybrid Model for Estimating Construction Waste for Multifamily Residential Buildings Using Artificial Neural Networks and Ant Colony Optimization. *Sustainability*, [S.l.], v. 9, n. 8, p. 1-14, set. 2016.
62. LEE, Hsin-Yun. An integrated model for planning development projects using ACO and construction simulation. *Civil Engineering And Environmental Systems*, [s.l.], v. 28, n. 4, p. 285-300, 2011.
63. LEE, Hsin-Yun. Optimizing schedule for improving the traffic impact of work zone on roads. *Automation in Construction*, v. 18, n. 8, p. 1034-1044, 2009. <http://dx.doi.org/DOI: 10.1016/j.autcon.2009.05.004>.
64. LEE, Hsin-Yun; TSENG, Hao-Hsi; ZHENG, Meng-Cong; LI, Pei-Ying. Decision support for the maintenance management of green areas. *Expert Systems With Applications*, [s.l.], v. 37, n. 6, p. 4479-4487, jun. 2010. Elsevier BV. DOI: <http://dx.doi.org/10.1016/j.eswa.2009.12.063>.

65. LEMOS, Patrícia Faga Iglecias; FERNANDEZ, Fernanda da Rocha Brando; GOMES, Tamara Maria; MÜLFART, Roberta Consentino Kronka (org.). *Sustentabilidade na USP*. São Paulo: SGA/USP, 2018. 124 p.
66. LI, Y.; HILTON, A. B. C. Reducing spatial sampling in long-term groundwater monitoring networks using ant colony optimization. *International Journal of Computational Intelligence Research*, [s.l.], v. 1, n. 1, p. 19-28, 2005. DOI: <https://doi.org/10.5019/ijcir.2005.1.2>.
67. LU, Li-Chih; YUE, Tai-Wen. Mission-oriented ant-team ACO for min-max MTSP. *Applied Soft Computing Journal*, v. 76, p. 436-444, 2019. DOI: <https://doi.org/10.1016/j.asoc.2018.11.048>.
68. LUPOAIE, Vlad-Ioan; CHILI, Ivona-Alexandra; BREABAN, Mihaela Elena; RASCHIP, Madalina. SOM-Guided Evolutionary Search for Solving MinMax Multiple-TSP. In: *IEEE Congress on Evolutionary Computation (CEC 2019)*, Wellington, New Zealand, 2019. Proceedings... p. 73-80.
69. MACÊDO, Bruno Rafael Morais de; LISBOA, Carolina Maria Cardoso Aires; CARVALHO, Fabíola Gomes de. Diagnóstico e diretrizes para a arborização do campus central da Universidade Federal do Rio Grande do Norte. *Revista da Sociedade Brasileira de Arborização Urbana*, Piracicaba, v. 7, n. 1, p. 35-51, 2012.
70. LAKATOS, Eva Maria; MARCONI, Marina de Andrade. *Fundamentos de metodologia científica*. 5. ed. São Paulo: Atlas, 2003.
71. MAIER, H. R. et al. Ant colony optimization for design of water distribution systems. *Journal of Water Resources Planning and Management*, [s.l.], v. 129, n. 3, p. 200-209, 2003. DOI: [https://doi.org/10.1061/\(ASCE\)0733-9496\(2003\)129:3\(200\)](https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(200)).
72. MATOS, Thiers; RENAULT, Flávia. *Projeto executivo de paisagismo: manual básico de implantação e manutenção de jardim*. Belo Horizonte: [s.n.], 200-?. 23 p.
73. MATTOS, Aldo Dórea. *Como preparar orçamentos de obras: dicas para orçamentistas, estudos de caso, exemplos*. São Paulo, SP: Pini, 2006. 281 p. ISBN 9788572661768 (broch.).
74. MENEZES, Emerson Goulart. *Poda em Oliveiras: princípios básicos*. 2021. Disponível em: <https://www.azeiteseolivais.com.br/post/poda-em-oliveiras-princ%C3%ADpios-b%C3%A1sicos>. Acesso em: 26 abril 2021.
75. MICROSOFT. *Introdução ao .NET Framework*. 2023a. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/framework/get-started/>. Acesso em: 02 out. 2023.
76. MICROSOFT. *What is .NET Framework?* 2023b. Disponível em: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework>. Acesso em: 02 out. 2023.
77. NAGEL, David; BYRD JR., John D. *Establish & Manage Your Home Lawn*. Starkville: Mississippi State University, 2015. 36 p.

78. NG, S. Thomas; ZHANG, Yanshuai. Optimizing construction time and cost using ant colony optimization approach. *Journal of Construction Engineering and Management*, v. 134, n. 9, p. 721-728, Sept. 2008. DOI: <https://doi.org/10.1061/ASCE0733-93642008134:9721>.
79. NING, X.; LAM, K.-C.; LAM, M. C.-K. Dynamic construction site layout planning using max-min ant system. *Automation in Construction*, v. 19, n. 1, p. 55-65, 2010. <https://doi.org/DOI: 10.1016/j.autcon.2009.09.002>.
80. PAIVA, Bárbara Souza; LUZ, Luziane Mesquita da; SILVA, Christian Nunes da. Sistemas de áreas verdes da Cidade Universitária Professor José da Silveira Netto, da UFPA, em Belém (PA). *Novos Cadernos NAEA*, [s.l.], v. 25, n. 1, p. 297-323, jan./abr. 2022.
81. PEREIRA, Sílvia de Castro; PIRES, Eduardo J. Solteiro; OLIVEIRA, Paulo B. de Moura. Ant-Balanced Multiple Traveling Salesmen: ACO-BmTSP. *Algorithms*, v. 16, n. 1, p. 37, 2023. DOI: <https://doi.org/10.3390/a16010037>.
82. PEZZUTTI, Raul Vicente; FLEIG, Frederico Dimas; FINGER, Cesar Augusto Guimaraes; SCHNEIDER, Paulo Renato. Estudo e modelagem do crescimento em diâmetro de povoamentos de *Pinus taeda* L. submetidos a podas e desbastes na Argentina. *Ciência Florestal*, [s.l.], v. 26, n. 4, p. 1225-1237, 28 dez. 2016. Universidad Federal de Santa Maria. DOI: <http://dx.doi.org/10.5902/1980509825113>.
83. PINI. TCPO – Tabela de Composições de Preços para Orçamentos. São Paulo: PINI, 2010. 630 p.
84. PRIGOGINE, I.; NICOLIS, G. Self-organisation in nonequilibrium systems: towards a dynamics of complexity. In: HAZEWINKEL, M.; JURKOVICH, R.; PAELINCK, J. H. P. (eds.). *Bifurcation Analysis: Principles, Applications and Synthesis*. Dordrecht: D. Reidel Publishing Company, 1985. p. 3-12.
85. RAATS, Luis Eduardo de Souza. Estudo de diferentes sistemas de plantio (placas e plugs) para grama esmeralda. 2005. 33 f. Dissertação (Mestrado) - Curso de Agronomia, Universidade Federal de Lavras, Lavras, 2005.
86. RAJENDRAN, C.; ZIEGLER, H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, v. 155, n. 2, p. 426-438, 2004. DOI: [https://doi.org/10.1016/S0377-2217\(02\)00908-6](https://doi.org/10.1016/S0377-2217(02)00908-6).
87. RAMADHANI, T.; HERTONO, G. F.; HANDARI, B. D. An ant colony optimization algorithm for solving the fixed destination multi-depot multiple traveling salesman problem with non-random parameters. In: *INTERNATIONAL SYMPOSIUM ON CURRENT PROGRESS IN MATHEMATICS AND SCIENCES (ISCPMS)*, 2016. Proceedings... Melville, NY: AIP Publishing, 2017. v. 1862, p. 030123-1-030123-7. DOI: <https://doi.org/10.1063/1.4991227>.
88. REICHER, Zac; BIGELOW, Cale; PATTON, Aaron; VOIGT, Tom. 7 simple steps to a better home lawn. West Lafayette: Purdue University Extension; University of Illinois Extension, 2006. (AY-32-W; IL-IN TW-72). Disponível em:

- <https://www.extension.purdue.edu/extmedia/ay/ay-32-w.pdf>. Acesso em: 24 mar. 2026.
89. RODRIGUES, Cleyde Cristina; ÁVILA, Lucimar Antônio Cabral de; TAVARES, Marcelo; VIEIRA, Lais Barbosa. Reflexos dos cortes orçamentários sobre os gastos das universidades públicas federais. In: CONGRESSO INTERNACIONAL DE CUSTOS, 18., 2023, Natal. Anais... Natal: Abc, 2023. p. 1-21.
90. RODRIGUES, Rodrigo Carneiro; CARVALHO, Tamires Koga de; MORI, Tamiris Benassi. Áreas verdes e arborização na área II do Campus de São Carlos. São Carlos: R2T Ambiental, 2014. 53 p.
91. ROSEN, K. H. Discrete mathematics and its applications. 8. ed. New York: McGraw-Hill Education, 2019.
92. SALVADOR, Elisabete Domingues; MINAMI, Keigo. Avaliação de diferentes substratos no cultivo de grama-esmeralda (*Zoysia japonica* Steud.) em bandejas. Ciência e Agrotecnologia, Lavras, v. 26, n. 2, p. 237–243, 2002. ISSN 1413-7054.
93. SANTA CATARINA. PODER JUDICIÁRIO DE SANTA CATARINA. Normativa de jardinagem para o Poder Judiciário de Santa Catarina. [s.l]: [s.n.], 2020. 26 p.
94. SANTOS, Aguinaldo dos. Seleção do método de pesquisa: guia para pós-graduando em design e áreas afins. Curitiba: Insight, 2018.
95. SEVERINO, Antônio Joaquim. Metodologia do trabalho científico. 23. ed. revista e atualizada. São Paulo: Cortez, 2007.
96. SIMON, Herbert Alexander. The sciences of the artificial. 3rd ed. Cambridge, Mass.: MIT, 1996. 231 p. ISBN 0262691914 (broch.).
97. SOLIMANPUR, M.; VRAT, P.; SHANKAR, R. An ant algorithm for the single row layout problem in flexible manufacturing systems. Computers & Operations Research, v. 32, n. 3, p. 583–598, 2005. DOI: <https://doi.org/10.1016/j.cor.2003.08.005>.
98. SOUSA, Rosilvam Ramos de; ARAÚJO, Luan Henrique Barbosa de; BORGES, César Henrique Alves; SOUTO, Patrícia Carneiro; SOUTO, Jacob Silva. Diagnóstico da arborização do campus de Patos - PB. Biofix Scientific Journal, [s. l.], v. 4, n. 1, p. 43-51, 2019.
99. SOUZA, Caroline Almeida; VELASCO, Giuliana del Nero (org.). Resíduo de poda de árvores urbanas: como reaproveitar? São Paulo: Instituto de Pesquisas Tecnológicas do Estado de São Paulo, 2022.
100. SRIKAKULAPU, R.; VINATHA, U. Combined approach based on ACO with MTSP for optimal internal electrical system design of large offshore wind farm. In: INTERNATIONAL CONFERENCE ON POWER, INSTRUMENTATION, CONTROL AND COMPUTING (PICC), 2018, Thrissur, Índia. Proceedings... PICC 2018. Los Alamitos, CA: IEEE, 2018. p. 1–6.

101. SUN, Bing; WANG, Chuan; YANG, Qiang; LIU, Weili; YU, Weijie. Ant Colony Optimization for Balanced Multiple Traveling Salesmen Problem. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND COMPUTATIONAL INTELLIGENCE (CSCI), 2021. Proceedings.... [s. l.]: IEEE, 2021. p. 476-481. DOI: <https://doi.org/10.1109/CSCI54926.2021.00148>.
102. SUN, Bing; WANG, Chuan; ZHENG, Zhong-Long; LIN, Ying; YU, Wei-Jie; LI, Mu-Xuan. Matrix-based ant colony optimization for large-scale balanced multiple traveling salesmen problem. In: 15th INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND SECURITY (CIS), 2023. Proceedings.... [s. l.: s. n.], 2023.
103. TAPPARO, Sergio André. Irrigação de precisão em diferentes variedades de gramíneas sob condição simulada de pastejo rotacionado em pivô central. 2008. 112 f. Dissertação (Mestrado) - Curso de Agronomia, Universidade de São Paulo, Piracicaba, 2008.
104. TURING, Alan. The chemical basis of morphogenesis. Philosophical Transactions of The Royal Society B., [s. l.: s. n.], vol. 237, n. 641, p. 37-72, agosto de 1952.
105. TURING, Alan. The Prof's Book: Turing's treatise on the Enigma. 1942. Disponível em: <https://archive.org/details/hw-25-3>.
106. UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO. Laboratório de Planejamento e Projetos. Plano diretor físico da UFES: campus Almor de Queiroz Araújo. Vitória: Universidade Federal do Espírito Santo, 2008. 113 p.
107. UNIVERSIDADE FEDERAL FLUMINENSE. Pregão eletrônico nº 110/2018/AD: anexo I (termo de referência). Niterói, 2018. 26 p.
108. UNIVERSIDADE DE BRASÍLIA. Cronograma dos Serviços de Manutenção das Áreas Verdes Externas dos campi da Universidade de Brasília (UnB): maio/julho. Brasília, 2017. 17 p.
109. UNIVERSIDADE DE SÃO PAULO. Sustentabilidade na USP. São Paulo: SGA/USP, 2018. 124 p.
110. VENNEN, Marie. Control of invasive species. Washington, D.C.: Transportation Research Board, 2006.
111. WALMSLEY, Kean. AutoCAD 2014 for developers. 2013. Disponível em: https://through-the-interface.typepad.com/through_the_interface/2013/03/autocad-2014-for-developers.html. Acesso em: 16 jul. 2025.
112. WANG, Min; MA, Tongmao; LI, Guiling; ZHAI, Xue; QIAO, Sibao. Ant Colony Optimization With an Improved Pheromone Model for Solving MTSP With Capacity and Time Window Constraint. IEEE Access, v. 8, p. 106872–106879, 2020. DOI: <https://doi.org/10.1109/ACCESS.2020.3000501>.
113. WILSON, Greg et al. Best practices for scientific computing. PLoS Biology, v. 12, n. 1, e1001745, 2014.

114. XIONG, Ying; KUANG, Yaping. Applying an Ant Colony Optimization Algorithm-Based Multiobjective Approach for Time–Cost Trade-Off. *Journal of Construction Engineering and Management*, v. 134, n. 2, p. 153–156, fev. 2008. DOI: [https://doi.org/10.1061/\(ASCE\)0733-9364\(2008\)134:2\(153\)](https://doi.org/10.1061/(ASCE)0733-9364(2008)134:2(153)).
115. XU, Nuo; WU, Deming; YANG, Qiang; WANG, Hua; ZHOU, Xiangmin; ZHENG, Zhonglong; GE, Yisu; GAO, Xudong. Ant Colony Optimization for Multiple Travelling Salesmen Problem with Pivot Cities. In: *15th International Conference on Advanced Computational Intelligence (ICACI), 2023. Proceedings...*, p. 1–8. DOI: <https://doi.org/10.1109/ICACI58115.2023.10146169>.
116. YOUSEFIKHOSHBAKHT, Majid; DIDEHVAR, Farzad; RAHMATI, Farhad. Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem. *Romanian Journal of Information Science and Technology*, [s.l.], v. 16, n.1, p.65–80, 2013. Disponível em: <https://www.romjist.ro/content/pdf/05-myousefikhoshbakht.pdf>. Acesso em: 20 ago. 2025.

7 APÊNDICES

APÊNDICE A – Íntegra do relatório gerado pelo plug-in para o primeiro cenário

Green Maintenance Optimizer REPORT

AutoCAD plug-in

GREEN MAINTENANCE OPTIMIZER REPORT

These are the configurations used in the simulation:

- Number of nodes
 - For grass: 113
 - For trees: 76
- Types of vegetation:
 - Grass
 - * Pruning frequency (summer and spring): 1 time per week
 - * Pruning frequency (winter and fall): 2 times per week
 - * Irrigation frequency (summer and spring): 1 time per week
 - * Irrigation frequency (winter and fall): 1 time per week
 - * Invasive species removal actions interval: 12 months
 - Trees
 - * Percentage of trees with trunks smaller than 20 cm: 25
 - * Percentage of trees with trunks greater than 20 cm and smaller than 40 cm: 25
 - * Percentage of trees with trunks greater than 40 cm and smaller than 60 cm: 25
 - * Percentage of trees with trunks greater than 60 cm: 25
- Activities setup:
 - Grass
 - * Grass watering is considered in simulation: Yes
 - * Time spent by foremen per 1000 m² of grass watering: 0:00:00
 - * Time spent by assistant per 1000 m² of grass watering: 2:30:00
 - * Season considered to estimate frequencies for grass watering: Fall/Winter
 - * Grass cutting is considered in simulation: Yes
 - * Time spent by foremen per 1000 m² of grass cutting: 0:06:00
 - * Time spent by assistant per 1000 m² of grass cutting: 2:00:00
 - * Season considered to estimate frequencies for grass cutting: Fall/Winter
 - * Invasive species removal is considered in simulation: Yes
 - * Time spent by foremen per 1000 m² of invasive species removal: 0:00:00
 - * Time spent by assistant per 1000 m² of invasive species removal: 4:00:00
 - * Season considered to estimate frequencies for invasive species removal: Spring/Summer
 - Trees

- * Time spent by foreman for pruning a tree with a trunk with less than 20 cm: 0:13:15
 - * Time spent by assistant for pruning a tree with a trunk with less than 20 cm: 0:18:06
 - * Time spent by foreman for pruning a tree with a trunk with more than 20 cm and less than 40 cm: 1:07:03
 - * Time spent by assistant for pruning a tree with a trunk with more than 20 cm and less than 40 cm: 1:29:23
 - * Time spent by foreman for pruning a tree with a trunk with more than 40 cm and less than 60 cm: 3:01:15
 - * Time spent by assistant for pruning a tree with a trunk with more than 40 cm and less than 60 cm: 4:01:40
 - * Time spent by foreman for pruning a tree with a trunk with more than 60 cm: 7:00:38
 - * Time spent by assistant for pruning a tree with a trunk with more than 60 cm: 9:56:50
- Manpower settings:
 - Workers' travel speed: 6 km/h
 - Number of shifts per working day: 2
 - Duration of working shift: 4 hours
 - Rest time per work shift: 0 (no rest)
 - Grid scale settings
 - Grid scale for grass: 50
 - Grid scale for trees: 100
 - Number of iteration settings
 - Number of iterations for initial search: 100
 - Number of iterations for refined search: 150
 - Number of teams settings
 - Number of teams for initial search: 2
 - Number of teams for refined search: 4

These are the steps of the simulation:

- The simulation applies the Team Ant Colony Optimization for sizing the maintenance teams for the campus green areas. The simulation divides the maintenance activities in two groups: the grass maintenance activities and the trees maintenance activities.

- The simulation for each group is divided in two parts: initial search and refined search. The initial search is proposed to use less ants teams and less iterations, in order to reduce the computational complexity and provide a good raw result; the refined search is proposed to use more ants teams and more iterations (with a higher computational complexity) to refine the raw result found in the initial search.
- The initial search starts the simulation with 1 ant in the ants teams; the journey times of all the ants are calculated, and if the longest journey between the ants is greater than the work shift, a new ant is added and the simulation runs again. This process is repeated until the longest journey among the ants is shorter than the work shift. When the longest journey is shorter than the work shift, the current team size represents the result of the initial search.
- The refined search then starts with a team size equals the initial search result minus one, and continues removing ants from the team, one at a time, to check if with more ants teams and more iterations the simulation is capable of finding better solutions. Below is the summary of the simulation.
- Grass maintenance activities (performed by `grassWorker0`)
 - Initial search:
 - * [TS = 1] LENGTH OF THE LONGEST JOURNEY: 5 days + 05:28:38
TOTAL TRAVEL TIME: 01:32:55
TOTAL LENGTH OF STAY IN NODES: 5.03:55:43
 - * [TS = 11] LENGTH OF THE LONGEST JOURNEY: 18:02:42
TOTAL TRAVEL TIME: 00:32:04
TOTAL LENGTH OF STAY IN NODES: 04:49:22
 - * [TS = 21] LENGTH OF THE LONGEST JOURNEY: 10:29:32
TOTAL TRAVEL TIME: 00:31:41
TOTAL LENGTH OF STAY IN NODES: 06:58:11
 - * [TS = 31] LENGTH OF THE LONGEST JOURNEY: 06:21:45
TOTAL TRAVEL TIME: 00:32:38
TOTAL LENGTH OF STAY IN NODES: 03:30:09
 - * [TS = 41] LENGTH OF THE LONGEST JOURNEY: 05:40:15
TOTAL TRAVEL TIME: 00:34:12
TOTAL LENGTH OF STAY IN NODES: 01:24:49
 - * [TS = 46] LENGTH OF THE LONGEST JOURNEY: 05:06:38
TOTAL TRAVEL TIME: 00:26:13
TOTAL LENGTH OF STAY IN NODES: 02:49:40
 - * [TS = 51] LENGTH OF THE LONGEST JOURNEY: 04:20:53
TOTAL TRAVEL TIME: 00:31:10
TOTAL LENGTH OF STAY IN NODES: 02:01:38
 - * [TS = 56] LENGTH OF THE LONGEST JOURNEY: 04:15:14
TOTAL TRAVEL TIME: 00:31:13
TOTAL LENGTH OF STAY IN NODES: 02:34:44

- * [TS = 61] LENGTH OF THE LONGEST JOURNEY: 04:15:14
TOTAL TRAVEL TIME: 00:32:09
TOTAL LENGTH OF STAY IN NODES: 02:19:49
- * [TS = 66] LENGTH OF THE LONGEST JOURNEY: 04:24:57
TOTAL TRAVEL TIME: 00:23:33
TOTAL LENGTH OF STAY IN NODES: 01:56:56
- * [TS = 71] LENGTH OF THE LONGEST JOURNEY: 04:05:13
TOTAL TRAVEL TIME: 00:23:21
TOTAL LENGTH OF STAY IN NODES: 02:05:01
- * [TS = 76] LENGTH OF THE LONGEST JOURNEY: 03:54:13
TOTAL TRAVEL TIME: 00:24:41
TOTAL LENGTH OF STAY IN NODES: 02:31:19
- Refined search:
 - * [TS = 73] LENGTH OF THE LONGEST JOURNEY: 04:31:06
TOTAL TRAVEL TIME: 00:23:57
TOTAL LENGTH OF STAY IN NODES: 02:03:47
 - * [TS = 74] LENGTH OF THE LONGEST JOURNEY: 03:54:20
TOTAL TRAVEL TIME: 00:24:14
TOTAL LENGTH OF STAY IN NODES: 02:03:47
 - * [TS = 75] LENGTH OF THE LONGEST JOURNEY: 03:47:36
TOTAL TRAVEL TIME: 00:24:38
TOTAL LENGTH OF STAY IN NODES: 01:47:43
- Grass maintenance activities (performed by grassWorker1)
 - Initial search:
 - * [TS = 1] LENGTH OF THE LONGEST JOURNEY: 4 days + 09:37:04
TOTAL TRAVEL TIME: 01:31:04
TOTAL LENGTH OF STAY IN NODES: 4.08:06:00
 - * [TS = 11] LENGTH OF THE LONGEST JOURNEY: 14:34:53
TOTAL TRAVEL TIME: 00:31:59
TOTAL LENGTH OF STAY IN NODES: 11:04:27
 - * [TS = 21] LENGTH OF THE LONGEST JOURNEY: 07:38:56
TOTAL TRAVEL TIME: 00:31:38
TOTAL LENGTH OF STAY IN NODES: 02:54:07
 - * [TS = 31] LENGTH OF THE LONGEST JOURNEY: 05:51:17
TOTAL TRAVEL TIME: 00:27:36
TOTAL LENGTH OF STAY IN NODES: 04:08:46
 - * [TS = 36] LENGTH OF THE LONGEST JOURNEY: 04:40:06
TOTAL TRAVEL TIME: 00:30:33
TOTAL LENGTH OF STAY IN NODES: 03:58:42
 - * [TS = 41] LENGTH OF THE LONGEST JOURNEY: 04:38:45
TOTAL TRAVEL TIME: 00:32:10
TOTAL LENGTH OF STAY IN NODES: 01:57:26

- * [TS = 46] LENGTH OF THE LONGEST JOURNEY: 04:09:28
TOTAL TRAVEL TIME: 00:26:09
TOTAL LENGTH OF STAY IN NODES: 02:22:31
- * [TS = 51] LENGTH OF THE LONGEST JOURNEY: 03:37:58
TOTAL TRAVEL TIME: 00:24:26
TOTAL LENGTH OF STAY IN NODES: 01:32:35
- Refined search:
 - * [TS = 50] LENGTH OF THE LONGEST JOURNEY: 04:22:18
TOTAL TRAVEL TIME: 00:23:21
TOTAL LENGTH OF STAY IN NODES: 01:45:01
- Trees maintenance activities
 - Initial search:
 - * [TS = 1] LENGTH OF THE LONGEST JOURNEY: 3 days + 12:23:49
TOTAL TRAVEL TIME: 01:38:17
TOTAL LENGTH OF STAY IN NODES: 3.10:45:32
 - * [TS = 11] LENGTH OF THE LONGEST JOURNEY: 16:24:28
TOTAL TRAVEL TIME: 00:32:23
TOTAL LENGTH OF STAY IN NODES: 05:55:27
 - * [TS = 21] LENGTH OF THE LONGEST JOURNEY: 07:50:24
TOTAL TRAVEL TIME: 00:32:58
TOTAL LENGTH OF STAY IN NODES: 04:38:35
 - * [TS = 31] LENGTH OF THE LONGEST JOURNEY: 05:48:13
TOTAL TRAVEL TIME: 00:25:30
TOTAL LENGTH OF STAY IN NODES: 03:39:45
 - * [TS = 36] LENGTH OF THE LONGEST JOURNEY: 05:34:07
TOTAL TRAVEL TIME: 00:29:23
TOTAL LENGTH OF STAY IN NODES: 02:31:18
 - * [TS = 41] LENGTH OF THE LONGEST JOURNEY: 05:35:01
TOTAL TRAVEL TIME: 00:32:44
TOTAL LENGTH OF STAY IN NODES: 02:39:42
 - * [TS = 46] LENGTH OF THE LONGEST JOURNEY: 06:20:00
TOTAL TRAVEL TIME: 00:20:33
TOTAL LENGTH OF STAY IN NODES: 01:54:04
 - * [TS = 56] LENGTH OF THE LONGEST JOURNEY: 05:03:05
TOTAL TRAVEL TIME: 00:26:39
TOTAL LENGTH OF STAY IN NODES: 01:46:52
 - * [TS = 61] LENGTH OF THE LONGEST JOURNEY: 05:05:07
TOTAL TRAVEL TIME: 00:26:39
TOTAL LENGTH OF STAY IN NODES: 01:46:52
 - * [TS = 66] LENGTH OF THE LONGEST JOURNEY: 03:30:28
TOTAL TRAVEL TIME: 00:28:45
TOTAL LENGTH OF STAY IN NODES: 01:38:28
 - Refined search:

- * [TS = 64] LENGTH OF THE LONGEST JOURNEY: 04:27:20
TOTAL TRAVEL TIME: 00:26:46
TOTAL LENGTH OF STAY IN NODES: 02:50:31
- * [TS = 65] LENGTH OF THE LONGEST JOURNEY: 03:32:22
TOTAL TRAVEL TIME: 00:28:39
TOTAL LENGTH OF STAY IN NODES: 01:49:16

This is the result for the ants' team sizing:

- Considering grass maintenance activities
 - TS = 74 (prision system worker)
 - TS = 51 (outsourced worker)
- Considering trees maintenance activities
 - TS = 65 (outsourced worker)

This is the result for the workers' team sizing:

- Given that each worker has two shifts a day, the optimum labour team size (LTS) will be half (plus one, if the division doesn't result in a whole number) of the sum of the optimum ants teams sizes for grass and trees (considering each worker type). Therefore, the optimum labour size is:
 - LTS = 37 (prision system worker) + 58 (outsourced worker)

APÊNDICE B – Íntegra do relatório gerado pelo plug-in para o segundo cenário

Green Maintenance Optimizer REPORT

AutoCAD plug-in

GREEN MAINTENANCE OPTIMIZER REPORT

These are the configurations used in the simulation:

- Number of nodes
 - For grass: 113
 - For trees: 76
- Types of vegetation:
 - Grass
 - * Pruning frequency (summer and spring): Twice a month
 - * Pruning frequency (winter and fall): Twice a month
 - * Irrigation frequency (summer and spring): Twice a month
 - * Irrigation frequency (winter and fall): Twice a month
 - * Invasive species removal actions interval: 12 months
 - Trees
 - * Percentage of trees with trunks smaller than 20 cm: 25
 - * Percentage of trees with trunks greater than 20 cm and smaller than 40 cm: 25
 - * Percentage of trees with trunks greater than 40 cm and smaller than 60 cm: 25
 - * Percentage of trees with trunks greater than 60 cm: 25
- Activities setup:
 - Grass
 - * Grass watering is considered in simulation: No
 - * Time spent by foremen per 1000 m² of grass watering: 0:00:00
 - * Time spent by assistant per 1000 m² of grass watering: 2:30:00
 - * Season considered to estimate frequencies for grass watering: Spring/Summer
 - * Grass cutting is considered in simulation: Yes
 - * Time spent by foremen per 1000 m² of grass cutting: 0:06:00
 - * Time spent by assistant per 1000 m² of grass cutting: 2:00:00
 - * Season considered to estimate frequencies for grass cutting: Spring/Summer
 - * Invasive species removal is considered in simulation: No
 - * Time spent by foremen per 1000 m² of invasive species removal: 0:00:00
 - * Time spent by assistant per 1000 m² of invasive species removal: 4:00:00
 - * Season considered to estimate frequencies for invasive species removal: Spring/Summer
 - Trees

- * Time spent by foreman for pruning a tree with a trunk with less than 20 cm: 0:13:15
 - * Time spent by assistant for pruning a tree with a trunk with less than 20 cm: 0:18:06
 - * Time spent by foreman for pruning a tree with a trunk with more than 20 cm and less than 40 cm: 1:07:03
 - * Time spent by assistant for pruning a tree with a trunk with more than 20 cm and less than 40 cm: 1:29:23
 - * Time spent by foreman for pruning a tree with a trunk with more than 40 cm and less than 60 cm: 3:01:15
 - * Time spent by assistant for pruning a tree with a trunk with more than 40 cm and less than 60 cm: 4:01:40
 - * Time spent by foreman for pruning a tree with a trunk with more than 60 cm: 7:00:38
 - * Time spent by assistant for pruning a tree with a trunk with more than 60 cm: 9:56:50
- Manpower settings:
 - Workers' travel speed: 6 km/h
 - Number of shifts per working day: 2
 - Duration of working shift: 4 hours
 - Rest time per work shift: 30 minutes
 - Grid scale settings
 - Grid scale for grass: 50
 - Grid scale for trees: 100
 - Number of iteration settings
 - Number of iterations for initial search: 100
 - Number of iterations for refined search: 100
 - Number of teams settings
 - Number of teams for initial search: 2
 - Number of teams for refined search: 4

These are the steps of the simulation:

- The simulation applies the Team Ant Colony Optimization for sizing the maintenance teams for the campus green areas. The simulation divides the maintenance activities in two groups: the grass maintenance activities and the trees maintenance activities.

- The simulation for each group is divided in two parts: initial search and refined search. The initial search is proposed to use less ants teams and less iterations, in order to reduce the computational complexity and provide a good raw result; the refined search is proposed to use more ants teams and more iterations (with a higher computational complexity) to refine the raw result found in the initial search.
- The initial search starts the simulation with 1 ant in the ants teams; the journey times of all the ants are calculated, and if the longest journey between the ants is greater than the work shift, a new ant is added and the simulation runs again. This process is repeated until the longest journey among the ants is shorter than the work shift. When the longest journey is shorter than the work shift, the current team size represents the result of the initial search.
- The refined search then starts with a team size equals the initial search result minus one, and continues removing ants from the team, one at a time, to check if with more ants teams and more iterations the simulation is capable of finding better solutions. Below is the summary of the simulation.
- Grass maintenance activities (performed by grassWorker0)
 - Initial search:
 - * [TS = 1] LENGTH OF THE LONGEST JOURNEY: 2 days + 05:30:03
TOTAL TRAVEL TIME: 01:27:03
TOTAL LENGTH OF STAY IN NODES: 2.04:03:00
 - * [TS = 11] LENGTH OF THE LONGEST JOURNEY: 07:29:03
TOTAL TRAVEL TIME: 00:31:49
TOTAL LENGTH OF STAY IN NODES: 01:26:23
 - * [TS = 21] LENGTH OF THE LONGEST JOURNEY: 04:19:10
TOTAL TRAVEL TIME: 00:31:38
TOTAL LENGTH OF STAY IN NODES: 01:27:03
 - * [TS = 26] LENGTH OF THE LONGEST JOURNEY: 04:05:42
TOTAL TRAVEL TIME: 00:32:12
TOTAL LENGTH OF STAY IN NODES: 00:58:43
 - * [TS = 31] LENGTH OF THE LONGEST JOURNEY: 03:00:54
TOTAL TRAVEL TIME: 00:32:10
TOTAL LENGTH OF STAY IN NODES: 00:58:43
 - Refined search:
 - * [TS = 27] LENGTH OF THE LONGEST JOURNEY: 03:46:29
TOTAL TRAVEL TIME: 00:26:08
TOTAL LENGTH OF STAY IN NODES: 01:35:22
 - * [TS = 28] LENGTH OF THE LONGEST JOURNEY: 03:22:11
TOTAL TRAVEL TIME: 00:30:35
TOTAL LENGTH OF STAY IN NODES: 00:51:15
 - * [TS = 29] LENGTH OF THE LONGEST JOURNEY: 03:06:58
TOTAL TRAVEL TIME: 00:30:51
TOTAL LENGTH OF STAY IN NODES: 02:35:09

* [TS = 30] LENGTH OF THE LONGEST JOURNEY: 03:06:58
 TOTAL TRAVEL TIME: 00:31:34
 TOTAL LENGTH OF STAY IN NODES: 01:13:19

- Trees maintenance activities

- Initial search:

- * [TS = 1] LENGTH OF THE LONGEST JOURNEY: 1 days + 18:55:32
 TOTAL TRAVEL TIME: 01:32:46
 TOTAL LENGTH OF STAY IN NODES: 1.17:22:46
- * [TS = 11] LENGTH OF THE LONGEST JOURNEY: 08:03:05
 TOTAL TRAVEL TIME: 00:23:26
 TOTAL LENGTH OF STAY IN NODES: 03:30:08
- * [TS = 21] LENGTH OF THE LONGEST JOURNEY: 04:05:51
 TOTAL TRAVEL TIME: 00:31:37
 TOTAL LENGTH OF STAY IN NODES: 02:01:17
- * [TS = 26] LENGTH OF THE LONGEST JOURNEY: 03:21:17
 TOTAL TRAVEL TIME: 00:29:58
 TOTAL LENGTH OF STAY IN NODES: 01:16:51

- Refined search:

- * [TS = 23] LENGTH OF THE LONGEST JOURNEY: 03:48:35
 TOTAL TRAVEL TIME: 00:22:47
 TOTAL LENGTH OF STAY IN NODES: 02:07:17
- * [TS = 24] LENGTH OF THE LONGEST JOURNEY: 03:16:43
 TOTAL TRAVEL TIME: 00:24:46
 TOTAL LENGTH OF STAY IN NODES: 02:42:42
- * [TS = 25] LENGTH OF THE LONGEST JOURNEY: 03:16:43
 TOTAL TRAVEL TIME: 00:26:46
 TOTAL LENGTH OF STAY IN NODES: 02:42:06

This is the result for the ants' team sizing:

- Considering grass maintenance activities
 - TS = 28 (outsourced worker)
- Considering trees maintenance activities
 - TS = 24 (outsourced worker)

This is the result for the workers' team sizing:

- Given that each worker has two shifts a day, the optimum labour team size (LTS) will be half (plus one, if the division doesn't result in a whole number) of the sum of the optimum ants teams sizes for grass and trees (considering each worker type). Therefore, the optimum labour size is:
 - $LTS = 26$ (outsourced worker)