

# Drones na Cidade Grande: Reduzindo Colisões em Entregas Aéreas

Fabiola M. C. de Oliveira<sup>1</sup>, Luiz F. Bittencourt<sup>2</sup>, Reinaldo A. C. Bianchi<sup>3</sup>,  
Carlos A. Kamienski<sup>1</sup>

<sup>1</sup>Centro de Matemática, Computação e Cognição  
Universidade Federal do ABC (UFABC), Santo André, SP – Brazil

<sup>2</sup>Instituto de Computação – Universidade Estadual de Campinas  
Campinas, SP

<sup>3</sup>Departamento de Engenharia Elétrica – Centro Universitário da FEI  
São Bernardo do Campo, SP

{fabiola.oliveira, cak}@ufabc.edu.br, bit@ic.unicamp.br

rbianchi@fei.edu.br

**Abstract.** *Delivery companies have already started to perform small-scale tests with drones. In general, to comply with big cities, it is necessary to consider a large number of drones. In this scenario, it is fundamental to avoid collisions with other drones or typical obstacles of urban spaces. This paper proposes a geometric collision avoidance strategy called GeoDrone, which has its effectiveness compared with two approaches: keeping the original route and a literature geometric procedure. GeoDrone reduced the collisions to up to 14.5 times related to the literature procedure. We conclude that geometric strategies substantially reduce the collisions in this scenario but do not eliminate them, indicating the necessity of specifying more sophisticated approaches.*

**Resumo.** *Empresas de entregas já estão realizando testes em pequena escala com drones. Em geral, para atender a grandes cidades, é necessário considerar um grande número de drones. Neste cenário, é fundamental evitar colisões com outros drones ou obstáculos típicos de ambientes urbanos. Este artigo propõe uma estratégia de prevenção de colisões chamada GeoDrone, que tem sua eficácia comparada com duas abordagens: manter a rota original e um algoritmo geométrico da literatura. GeoDrone reduziu as colisões em até 14,5 vezes em relação ao algoritmo da literatura, portanto, concluímos que estratégias geométricas reduzem substancialmente as colisões neste cenário, mas não as eliminam, indicando a necessidade de especificar abordagens mais sofisticadas.*

## 1. Introdução

Veículos aéreos não tripulados (VANT), ou drones, serão pervasivos em áreas urbanas, oferecendo aos cidadãos serviços como entrega de alimentos e encomendas, detecção de incêndio, combate a incêndio e transporte de equipamentos médicos [Sánchez et al., 2020]. Drones voam em baixas altitudes, o que aumenta a probabilidade de colisão com prédios, árvores, pessoas, pássaros ou mesmo a colisão entre drones devida a sua alta densidade em áreas urbanas [Al-Mousa et al., 2019]. Assim, o primeiro passo para implementar um sistema multidrones é evitar colisões [Sánchez et al., 2020]. Em um cenário com

alta densidade de obstáculos, os drones podem necessitar de um conjunto de estratégias de prevenção de colisões para aplicar a técnica mais adequada em cada situação [Yasin et al., 2020].

Em um trabalho anterior [de Oliveira et al., 2021], propusemos um cenário de logística multidrones, no qual várias empresas de compras pela Internet, logísticas e de entrega de alimentos entregam pacotes usando drones. Avaliamos técnicas simples de prevenção de colisões nesse cenário, como tomar um desvio aleatório ou tomar um desvio à direita, que é uma estratégia inspirada pelas regras vigentes da aviação [Guan et al., 2020]. Mostramos que essas estratégias aumentam as colisões em relação a uma estratégia em que o drone não desvia.

Além do nosso trabalho anterior, segundo nosso conhecimento, os trabalhos que propõem estratégias de prevenção de colisões não avaliam as técnicas propostas em cenários de entregas com drones em que há pousos e decolagens e alta densidade de drones. A maioria dos trabalhos considera apenas o voo em cruzeiro, propondo ou testando estratégias em apenas duas dimensões, e não considera o pouso e a decolagem dos drones [Choi and Kim, 2020; Yasin et al., 2020], que podem gerar muitas colisões. Os trabalhos também indicam que nenhuma técnica de prevenção de colisões pode evitar todas as colisões [Yasin et al., 2020].

Este artigo propõe uma estratégia geométrica de prevenção de colisões baseada nas estratégias propostas por Seo et al. [2017]. Avaliamos o número de colisões no cenário de serviço de entregas aéreas proposto anteriormente [de Oliveira et al., 2021] para três estratégias de prevenção de colisões, que tomam atitudes diferentes quando calculam que um drone colidirá com um obstáculo: 1) *fazer nada*, em que os drones mantêm sua rota inicial; 2) algoritmo proposto por Seo et al. [2017]; e 3) nossa estratégia, chamada GeoDrone. As duas últimas estratégias são geométricas, que utilizam informações como localização e velocidade dos drones e obstáculos para o cálculo e a evasão de colisões.

Incluímos essas estratégias no simulador UTSim [Al-Mousa et al., 2019] e o instrumentamos para gerar resultados sobre as colisões para diferentes tamanhos de frotas de drones. Em seguida, avaliamos os resultados gerais e específicos sobre a viagem dos drones para explicar as situações de colisão. Os resultados mostram que nosso algoritmo GeoDrone reduziu o número de colisões em até 14,5 vezes em relação à estratégia proposta por Seo et al. [2017] e 25 vezes em relação a *fazer nada* para frotas com mais de 20 drones no cenário de serviço de entregas aéreas. As principais contribuições deste trabalho são mostrar que: 1) estratégias geométricas como a de Seo et al. [2017] podem aumentar as colisões em relação a *fazer nada*; 2) GeoDrone é uma estratégia eficaz para prevenção de colisões; e 3) estratégias mais sofisticadas podem ser necessárias para eliminar as colisões em cruzeiro.

Este artigo está organizado da seguinte maneira: a seção 2 apresenta os trabalhos relacionados nas áreas de prevenção de colisão aérea e simuladores de drones; a seção 3 descreve o cenário que simulamos; a seção 4 apresenta o algoritmo da literatura utilizado nas simulações; a seção 5 descreve nossa estratégia geométrica de prevenção de colisões; a seção 6 apresenta os parâmetros das simulações realizadas e as estratégias de prevenção de colisão utilizadas; a seção 7 descreve os resultados; a seção 8 discute os resultados; e a seção 9 descreve a conclusão e os trabalhos futuros.

## 2. Trabalhos Relacionados

Nesta seção, discutimos técnicas de prevenção de colisões e simuladores de drones.

### 2.1. Técnicas de Prevenção de Colisões

Yasin et al. [2020] revisaram várias estratégias de prevenção de colisão para veículos autônomos com ênfase em drones. As estratégias foram categorizadas em: geométricas, que utilizam localização e velocidade dos objetos; campos de força, que impõem uma força de atração no destino e uma força de repulsão nos obstáculos para se ter uma rota otimizada; otimizadas, que utilizam algoritmos de otimização como algoritmos genéticos e métodos gulosos; e detecção e prevenção, que tomam decisões em tempo de execução quando um obstáculo é detectado.

Seo et al. [2017] propuseram algumas estratégias geométricas de prevenção de colisões para VANTs sozinhos ou em formação. A estratégia define um envelope de prevenção de colisão, que leva em conta os limites das taxas angulares e do alcance do sensor de obstáculos e indicam se é possível desviar. Os autores também propuseram uma estratégia para evitar a colisão considerando múltiplos obstáculos através da definição de um ângulo de desvio em relação a todos os obstáculos com os quais há possibilidade de colisão. Entretanto, os autores consideraram que a magnitude e a direção da velocidade de todos os VANTs e obstáculos são constantes, executaram simulações em apenas duas dimensões e com no máximo dois obstáculos, o que são limitações do trabalho e não representam adequadamente nosso cenário de serviço de entregas.

Choi and Kim [2020] propuseram uma estratégia do tipo campos de força para ambientes estáticos ou dinâmicos, modificando a força de repulsão axial de obstáculos para forças de repulsão circulares. Esta mudança também elimina a possibilidade de as aeronaves ficarem presas em mínimos locais e gera caminhos mais eficientes do que outras estratégias de campos de força. Entretanto, a estratégia foi aplicada em um cenário com apenas três drones e duas dimensões.

Gageik et al. [2015] adotaram uma estratégia de detecção e prevenção com três classificações de regiões dentro do alcance dos sensores dos drones. Na região mais próxima ao drone, um desvio deve ser tomado imediatamente quando um obstáculo se encontra nela. Um obstáculo presente na região intermediária deve ser monitorado e o drone deve controlar sua distância em relação a ele. Na região mais afastada, o drone não toma decisões mesmo na presença de obstáculos. Contudo, essa estratégia não considera ambientes tridimensionais.

Chen et al. [2022] propuseram DroneTalk, um sistema de entregas por drones baseado na Internet das Coisas para entrega de correspondência em locais internos. Para evitar colisões, DroneTalk utiliza uma estratégia geométrica baseada no cálculo simultâneo das latências de comunicação, processamento e controle. Ele define uma velocidade máxima segura de acordo com essas latências e a distância entre o drone e qualquer obstáculo no caminho. Entretanto, as simulações envolveram apenas um único drone.

As estratégias de detecção e prevenção e geométrica possuem melhor desempenho em tempo real e são as únicas que consideram a dinâmica do drone e do obstáculo [Yasin et al., 2020]. Também são as únicas estratégias que funcionam em ambientes estáticos e dinâmicos, embora as estratégias geométricas sejam mais adequadas para áreas externas. Essas duas classes de estratégias e a estratégia do tipo otimização podem simular

ambientes tridimensionais mais facilmente. As estratégias de otimização e de campos de força são mais adequadas para o planejamento de voos em ambientes estáticos. Por fim, as estratégias geométricas e de campos de força possuem alto custo computacional, as estratégias otimizadas possuem custo médio e as estratégias de detecção e prevenção, baixo custo. Devido às vantagens da estratégia geométrica, neste trabalho, propomos uma estratégia deste tipo e a comparamos com a estratégia geométrica proposta por Seo et al. [2017]. Escolhemos essa estratégia devido aos seus resultados promissores e às características de: 1) verificar se um desvio será realizado em tempo hábil para evitar uma colisão; 2) realizar desvios considerando múltiplos obstáculos dinâmicos; e 3) considerar um ambiente tridimensional.

Os trabalhos avaliados nesta seção mostram que o campo de prevenção de colisões é bastante explorado. A maioria dos trabalhos avaliou cenários com poucos drones e em duas dimensões, enquanto o cenário de serviço de entregas ocorre em três dimensões e pode conter um grande número de drones em operação. Neste artigo, consideramos um cenário em três dimensões e simulamos frotas de até 200 drones.

## **2.2. Simuladores de Drones**

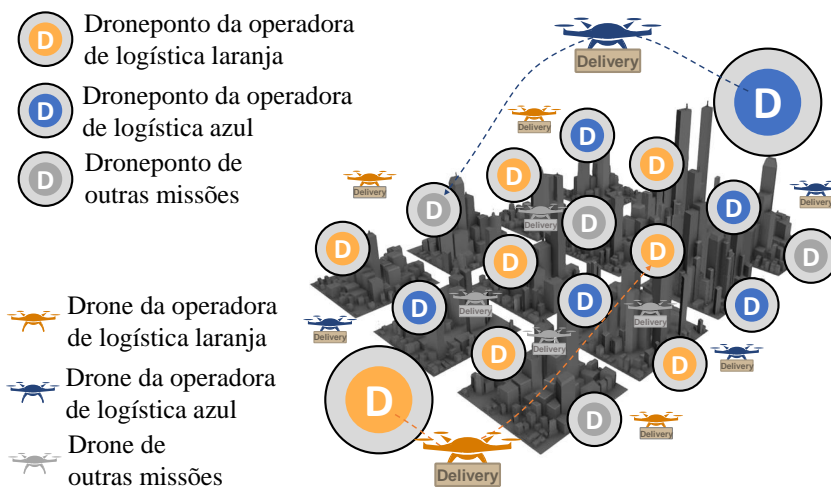
Realizamos uma extensa revisão de simuladores de drones em nosso trabalho anterior [de Oliveira et al., 2021]. Diversos trabalhos revisaram simuladores comerciais e de código aberto de VANTs, simuladores para análise de desempenho em sistemas de drones e simuladores de drones para aplicações específicas [Craighead et al., 2007; Hentati et al., 2018; Mairaj et al., 2019]. A maioria dos trabalhos foca no controle de navegação de um drone único ou poucos drones, no treinamento de pilotos e no entretenimento.

UTSim é um simulador de drones de código aberto desenvolvido em C# sobre a plataforma Unity, que simula a dinâmica dos objetos [Al-Mousa et al., 2019]. Ele pode simular mais de 1500 drones, suas especificações físicas e características de integração do tráfego aéreo, como navegação, controle, protocolos de comunicação e detecção e prevenção de colisões com objetos estáticos e dinâmicos. O simulador permite definir drones com diferentes características e utilizar diferentes algoritmos para detecção e prevenção de colisões. Escolhemos esse simulador para desenvolver nosso trabalho devido às suas características apropriadas para simulação de serviço de entregas por drones.

## **3. Cenário de Serviço de Entregas Aéreas**

A Figura 1 representa o cenário de serviço de entregas aéreas de encomendas utilizando drones proposto em nosso artigo anterior [de Oliveira et al., 2021]. Nessa figura, existem duas operadoras de logística de entregas (laranja e azul), podendo estar em operação drones de outras missões (cinza), como missões esporádicas ou transporte de pessoas [Ayamga et al., 2021; Kasliwal et al., 2019]. Cada operadora de logística pode ter um ou mais centros de distribuição (CD) de mercadorias, onde existem dronepontos (identificados pela letra D) para pousos e decolagens dos drones, e as entregas são realizadas em dronepontos menores espalhados pela cidade. Além das duas operadoras de logística da Figura 1, diversas outras podem estar presentes [Frachtenberg, 2019].

O desafio fundamental em um serviço de entregas baseado em drones é evitar colisões [Yang and Wei, 2021], que podem ocorrer com outros drones, objetos voadores, pássaros ou obstáculos estáticos como prédios e árvores. Como em aeronaves, a



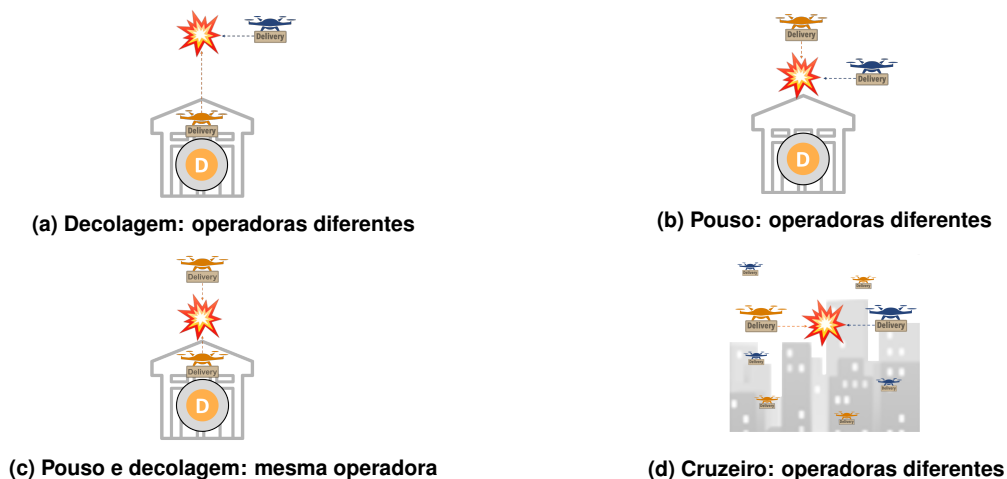
**Figura 1. Cenário de serviço de entregas por drones [de Oliveira et al., 2021].**

probabilidade de colisão é maior próximo aos dronepontos, durante pousos e decolagens [Kleinbekman et al., 2018]. Entretanto, um serviço de entregas envolvendo várias operadoras logísticas não possui coordenação centralizada global, como o sistema de controle de tráfego aéreo. Cada operadora pode implementar o próprio sistema otimizado para sequenciamento de pousos e decolagens nos dronepontos de seus CDs. Contudo, os drones devem evitar colisões em cruzeiro, ou seja, fora da região dos dronepontos.

A Figura 2 apresenta quatro situações típicas de colisões entre drones que não tomam atitudes para evitar colisões. Na Figura 2a, um drone decolando de um droneponto está em rota de colisão com um drone que está sobrevoando o droneponto e, na Figura 2b, um drone pousando em um droneponto está em rota de colisão com outro drone que sobrevoa o droneponto. Uma solução para colisões desses tipos com drones de operadoras diferentes é a instituição de regiões proibidas, como são os aeroportos na aviação. A Figura 2c mostra dois drones de uma mesma operadora em rota de colisão, em que um drone está pousando e o outro está decolando do mesmo droneponto. Esta situação ocorre quando não há um controle de sequenciamento de pousos e decolagens no droneponto de uma operadora. Por fim, a Figura 2d apresenta drones de operadoras diferentes em rota de colisão enquanto estão em cruzeiro, situação que requer abordagens mais sofisticadas de prevenção de colisões. Embora a Figura 2d apresente drones de operadoras diferentes, sem um controle centralizado ou comunicação, essa situação também pode acontecer com drones de uma mesma operadora. Neste trabalho, não há um controle centralizado de prevenção de colisões, sequenciamento de pousos e decolagens nos dronepontos ou comunicação entre os drones. Os drones tomam decisões de desvio individuais baseados nos dados fornecidos pelos seus sensores.

#### **4. Estratégia Proposta por Seo et al. [2017]**

Utilizamos a estratégia que considera múltiplos obstáculos proposta por Seo et al. [2017] para comparação e como base para nossa proposta, portanto, explicamos seu funcionamento a seguir. O raio de detecção de um drone indica o tamanho de uma esfera ao redor do drone dentro da qual qualquer obstáculo é identificado pelo sensor de obstáculos do drone. Similarmente, o raio de segurança de um drone indica o tamanho de uma esfera



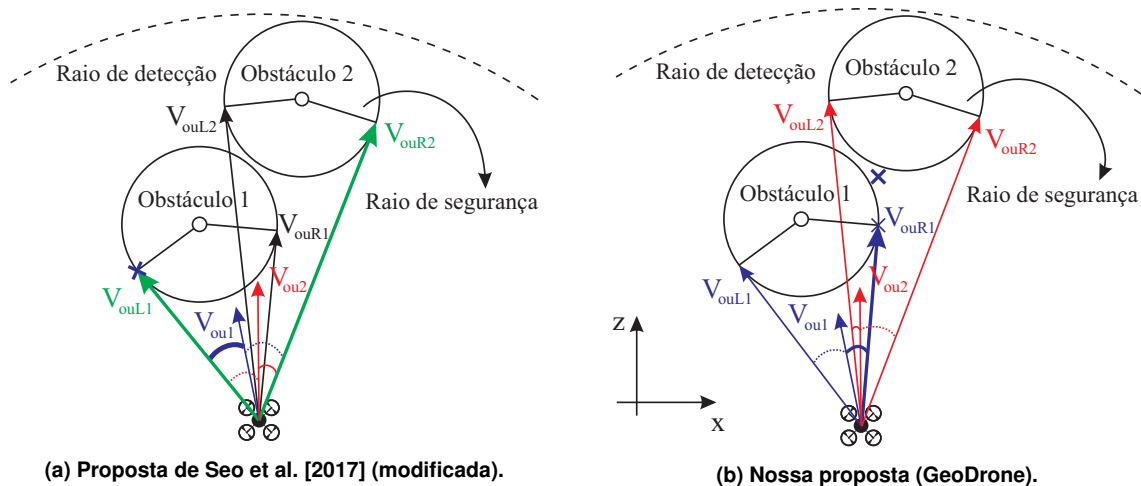
**Figura 2. Diferentes situações de colisão entre drones [de Oliveira et al., 2021].**

ao redor do drone dentro da qual os autores não querem nenhum obstáculo. A estratégia de Seo et al. [2017] funciona para obstáculos estáticos e dinâmicos e é dividida em duas etapas, detecção de colisão e desvio, de forma que um desvio deverá ser realizado sempre que uma colisão for detectada. Quando um obstáculo está a uma distância menor ou igual ao raio de detecção de um drone e dentro da semi-esfera definida pelo vetor de velocidade de um drone e o raio de detecção, ocorre a detecção. Os autores detectam que haverá uma colisão se o ponto mais próximo de abordagem em que o drone estará do obstáculo estiver dentro da esfera formada pelo raio de segurança. Esta situação nem sempre causa uma colisão, contudo, os autores definiram que a esfera formada pelo raio de segurança não deve conter nenhum obstáculo. Assim, o drone deve realizar um desvio.

Na etapa de desvio, um cone de raio igual ao raio de segurança é traçado ao redor de cada obstáculo com o qual o algoritmo identificou uma colisão; este é o cone de colisão representado na Figura 3a. O algoritmo escolhe a direção que está à esquerda e a direção que está à direita de todos esses obstáculos (vetores  $V_{ouL1}$  e  $V_{ouR2}$  em verde) e calcula o ângulo de desvio entre ambas as direções e o vetor de velocidade relativa entre o drone e cada obstáculo ( $V_{ou1}$  e  $V_{ou2}$ ). Em seguida, o algoritmo escolhe o menor ângulo de desvio (à direita ou à esquerda) para cada obstáculo (ângulos com linha cheia), para o VANT realizar um desvio menor e gastar menos energia. Por fim, o algoritmo escolhe o maior ângulo de desvio entre todos os obstáculos (ângulo em azul com linha cheia), porque, segundo os autores, a prevenção de colisão é garantida dessa forma [Seo et al., 2017]. O algoritmo calcula se é possível realizar um desvio horizontal para a taxa angular (manobrabilidade) vertical do VANT e, se não for possível, o algoritmo realiza um desvio vertical. O desvio termina quando o vetor de velocidade relativa fica perpendicular ao vetor de linha de visão em relação ao obstáculo que será tangenciado, ou seja, quando a projeção da velocidade relativa sobre o vetor de linha de visão for igual a zero, o que resulta no ponto indicado por uma cruz na Figura 3a. Indicaremos este algoritmo pelo nome *Seo et al* daqui em diante.

## 5. GeoDrone

Nossa estratégia para prevenção de colisões é do tipo geométrica, foi inspirada na estratégia que considera múltiplos obstáculos proposta por Seo et al. [2017] (algoritmo *Seo*



**Figura 3. Diferentes desvios geométricos considerando múltiplos drones.**

*et al*) e funciona para quaisquer valores de raio de detecção e velocidade máxima dos drones. Realizamos sete características distintivas em relação a esse algoritmo. A primeira característica foi na forma como o desvio é indicado no simulador. Enquanto o *Seo et al* define um ângulo de desvio para o drone alterar, o simulador UTSim trabalha com a definição da posição de destinos. Assim, calculamos, no momento de detecção de colisão, a posição para onde um drone deve ir para que o desvio termine segundo a condição explicada na Seção 4. A segunda característica foi a inclusão de uma tolerância para o cálculo do término de desvio, necessária devido ao período de amostragem da simulação. Enquanto em um passo de tempo a magnitude da projeção é um pouco maior que zero, no passo seguinte, essa magnitude pode já ter sido igual a zero e ser positiva novamente e, assim, o desvio nunca terminaria. Utilizamos o valor padrão da Unity de 0,02 para o período de amostragem da simulação e 0,12 para a tolerância no cálculo do término de desvio, baseado nas simulações. Devido ao uso do UTSim, foi necessário aplicar as duas primeiras características na implementação do *Seo et al* no simulador.

A terceira característica foi a escolha de um ângulo de desvio para o caso de um obstáculo estar dentro da esfera definida pelo raio de segurança do drone, ou seja, um desvio emergencial. Como Seo et al. [2017] não consideraram esse caso, aplicamos essa característica em sua implementação no simulador. Nossa estratégia foi adotar um valor de  $180^\circ$  para que o drone diminua sua velocidade a zero e desvie no sentido contrário ao que estava voando no momento da detecção de colisão. Esse valor foi escolhido baseado nas simulações, nas quais a dinâmica dos drones é simulada pela Unity. Verificamos que o drone do UTSim possui diferentes taxas angulares dependendo do destino escolhido, portanto, fixamos o valor de  $150^\circ/s$ , que é um valor comum em drones comerciais [DJI, 2022]. Para fixar a taxa angular, a quarta característica foi no eixo longitudinal do destino, sendo a cruz com linha menos espessa o destino no *Seo et al* e a cruz com linha mais espessa o nosso destino na Figura 3b.

O algoritmo *Seo et al* produziu alguns impasses nas simulações do nosso cenário. Seo et al. [2017] utilizaram velocidades constantes e iguais para todos os VANTs em suas simulações. Para resolver os impasses, a quinta característica foi a velocidade inicial dos drones, que passou a admitir um valor aleatório entre a velocidade máxima do drone e

metade dessa velocidade. A sexta característica foi a velocidade do drone quando ele detecta uma colisão, que foi alterada para um valor aleatório entre a velocidade inicial e metade dela. Assim, removemos a sincronia entre os drones e nenhum impasse ocorreu no GeoDrone.

Por fim, o caso da Figura 3a raramente acontece nas simulações do nosso cenário, pois os obstáculos deveriam entrar no raio de detecção de um drone no mesmo passo de simulação para que esse caso ocorra. Assim, a sétima característica foi a execução do algoritmo conforme mostra a Figura 3b, para que o drone desvie do primeiro obstáculo que entra em seu raio de detecção e em rota de colisão. Esse desvio pode acontecer horizontalmente ou verticalmente, de acordo com o mesmo cálculo de *Seo et al.* Contudo, se não for possível executar o desvio em nenhum dos dois planos, o drone colidirá. Em caso de sucesso, após o drone terminar esse primeiro desvio, caso ainda haja obstáculos em rota de colisão com ele, o drone desvia do obstáculo que está mais perto dele. Como o caso da Figura 3a raramente acontece, a característica de priorizar o obstáculo mais próximo também foi implementada no algoritmo *Seo et al* visando padronizar as comparações nas simulações.

O Algoritmo 1 mostra os passos de nossa estratégia em alto nível. Implementamos duas funções, a *GeraDrones()* e a *GeoDrone()*. A função *GeraDrones()* implementa a quinta característica do GeoDrone, criando um drone na simulação a cada 10 s com uma velocidade inicial aleatória entre a velocidade máxima do drone e metade dessa velocidade e manobrabilidade de  $150^\circ/s$  até que *númeroDeDrones* sejam criados (linhas 3–7).

A função *GeoDrone()* é chamada toda vez que um obstáculo está no raio de detecção de um drone (linhas 9–33). Inicialmente, o algoritmo calcula a velocidade relativa entre o drone e o obstáculo, armazena esse resultado em *velRel* e calcula se há a possibilidade de colisão (linhas 10–11). Se uma colisão for detectada e o drone não estiver desviando, o algoritmo altera a velocidade do drone para um valor aleatório entre a velocidade inicial e metade dessa velocidade (linhas 12 e 13). Se a distância entre o drone e o obstáculo for menor que o raio de segurança, o ângulo de desvio passa a ser  $180^\circ$ ; se não, o algoritmo calcula o ângulo de desvio para o drone tangenciar o raio de segurança em volta do obstáculo (linhas 14–18).

Em seguida, o algoritmo escolhe o menor ângulo de desvio para cada obstáculo com o qual foi calculada uma colisão (em passos anteriores da simulação) e, dentre esses valores, o maior ângulo de desvio (linhas 19 e 20). Na linha 21, o algoritmo calcula se é possível realizar um desvio horizontal com a manobrabilidade vertical do drone. Se esse desvio for possível, ele é calculado da mesma maneira que o *Seo et al* para o eixo transversal e, em seguida, o desvio no eixo longitudinal recebe este valor dividido pela manobrabilidade vertical para garantir a manobrabilidade do drone (linhas 22 e 23). Se o desvio horizontal não for possível, o algoritmo calcula um desvio vertical com o eixo vertical equivalente ao desvio de *Seo et al* e o eixo longitudinal recebe este valor dividido pela manobrabilidade transversal para garantir a manobrabilidade do drone (linhas 24–28). Embora utilizamos a manobrabilidade vertical igual à manobrabilidade transversal, o algoritmo pode ser facilmente modificado para considerar manobrabilidades diferentes. O desvio horizontal é preferível ao desvio vertical devido aos custos de se voar em altitudes mais altas, ou seja, ao maior tempo gasto em pousos e decolagens, que pode ser relevante



se o drone realizar muitos desvios verticais. Outra vantagem da preferência ao desvio horizontal é a facilidade de uma futura implementação de um serviço com vias aéreas pré-estabelecidas em diferentes altitudes. Se o drone estiver desviando e a magnitude da projeção do vetor da velocidade relativa (*velRel*) sobre o vetor de linha de visão for menor que 0,12, o drone termina o desvio e volta à viagem original (linhas 30–32).

---

### Algoritmo 1 GeoDrone.

---

```

1: function UTSIM(númeroDeDrones, manobrabilidade)
2:   /* Geração e inicialização dos drones; implementação da quinta característica */
3:   function GERADRONES(númeroDeDrones, manobrabilidade)
4:     Gera um drone i a cada 10 s por númeroDeDrones vezes;
5:     i.velocidade ← número aleatório entre velocidade máxima e metade dela;
6:     i.manobrabilidade ← manobrabilidade;
7:   end function
8:   /* Função chamada toda vez que um obstáculo está no raio de detecção de um drone */
9:   function GEODRONE(Obstáculo)
10:    velRel ← Calcula velocidade relativa entre drone e Obstáculo;
11:    Calcula a possibilidade de colisão;
12:    if colisão for detectada E o drone não estiver desviando then
13:      drone.velocidade ← número aleatório entre velocidade inicial e metade dela;
14:      if Distância(drone, Obstáculo) < raio de segurança then
15:        Ângulo de desvio ← 180°;
16:      else
17:        Calcula ângulo de desvio para tangenciar Obstáculo + raio de segurança;
18:      end if
19:      Escolhe o menor ângulo de desvio para cada obstáculo que colidirá;
20:      Escolhe o maior ângulo entre os ângulos escolhidos na etapa anterior;
21:      if for possível realizar desvio horizontal para manobrabilidade then
22:        Desvio.transversal ← desvio de Seo et al. no eixo transversal;
23:        Desvio.longitudinal ← Desvio.transversal / manobrabilidade;
24:      else
25:        /* Desvia verticalmente */
26:        Desvio.vertical ← desvio de Seo et al. no eixo vertical;
27:        Desvio.longitudinal ← Desvio.vertical / manobrabilidade;
28:      end if
29:    end if
30:    if drone está desviando E Projeção(velRel, linha de visão) < 0,12 then
31:      Termina desvio;
32:    end if
33:  end function
34: end function

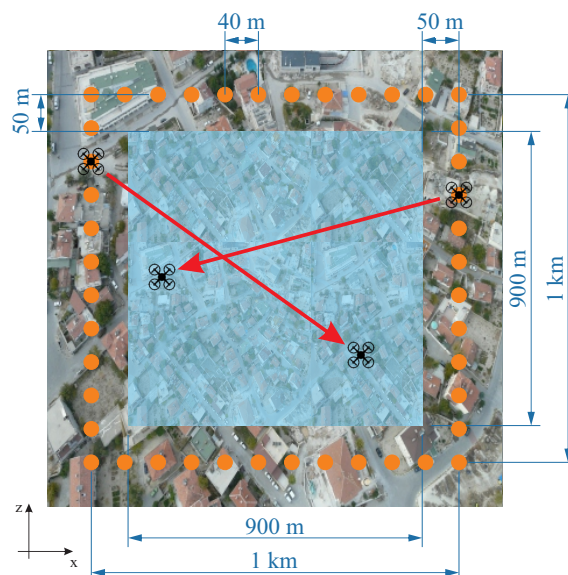
```

---

## 6. Metodologia

Executamos as simulações em nossa versão modificada do simulador UTSim. Os cenários simulados visam analisar o desempenho dos algoritmos sob uma alta densidade de drones, representando um cenário mais crítico do que o cenário da Seção 3. A Figura 4 mostra a vista superior do cenário para as simulações. Os círculos laranjas representam os 100 centros de distribuição (CD), dispostos a cada 40 m ao longo de um quadrado de 1 km de lado e de onde os drones decolam e pousam. Escolhemos um número grande de CDs para diminuir o número de colisões durante os pousos e decolagens, já que este tipo de colisão pode ser evitado com um controle centralizado. Os drones realizam entregas numa área

de  $0,81 \text{ km}^2$  concêntrica ao quadrado onde os CDs se situam para que todos os pontos de entrega tenham uma distância de segurança de pelo menos 50 m de todos os CDs. A região azul na figura representa a área de entrega. Não há vias aéreas pré-estabelecidas; os drones podem viajar livremente em diferentes altitudes, latitudes e longitudes.



**Figura 4. Vista superior do cenário utilizado nas simulações [de Oliveira et al., 2021] (modificada).**

Os drones têm um tamanho de  $2 \text{ m} \times 2 \text{ m} \times 0,5 \text{ m}$ , de forma que eles sejam capazes de transportar cargas maiores e não apenas realizar a entrega da última milha, ou seja, a entrega de uma encomenda ao ponto final do trajeto. Os drones também possuem uma velocidade máxima de  $20 \text{ m/s}$ , um raio de segurança de  $30 \text{ m}$  e um raio de detecção de  $100 \text{ m}$ . Consideramos sensores de detecção de luz (LiDAR, do inglês *Light Detection and Ranging*), que podem ter uma boa precisão e exatidão até  $100 \text{ m}$ . Simulamos sete frotas de tamanhos crescentes, com 2, 5, 10, 20, 50, 100 e 200 drones, que decolam sequencialmente a cada  $10 \text{ s}$  de um CD escolhido aleatoriamente. Cada drone realiza apenas uma entrega e retorna ao CD de onde decolou. A viagem dos drones é configurada inicialmente em 10 passos: 1) inicialização; 2) definição da velocidade inicial; 3) decolagem à altitude de  $10 \text{ m}$ ; 4) viagem até destino escolhido aleatoriamente dentro da área de  $0,81 \text{ km}^2$  indicada na Figura 4; 5) pouso à altitude de  $1 \text{ m}$ ; 6) espera de  $10 \text{ s}$  para a entrega; 7) decolagem à altitude de  $10 \text{ m}$ ; 8) viagem até CD de decolagem; 9) pouso à altura de  $1 \text{ m}$ ; e 10) inativação. Quando um drone decide tomar um desvio, ele inclui destinos temporários nesta configuração inicial. A escolha pela altitude de voo a  $10 \text{ m}$  deve-se ao menor tempo gasto nos pousos e decolagens, o que diminui o tempo total de simulação e aumenta a densidade dos drones em cruzeiro, tornando o cenário das simulações mais desafiador para estratégias de prevenção de colisões.

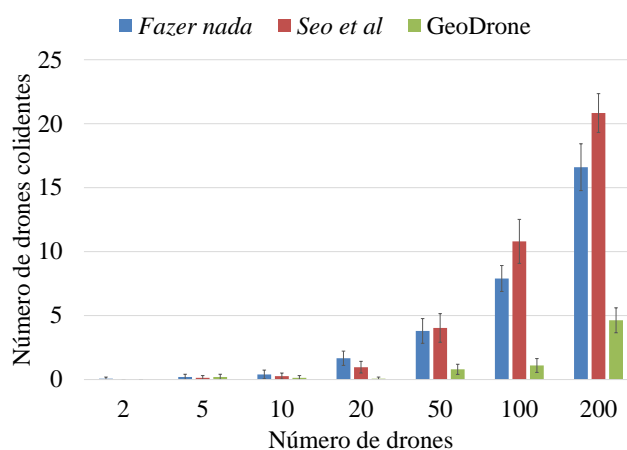
A simulação termina quando uma das seguintes condições é satisfeita: 1) todos os drones terminam suas viagens ou colidem; ou 2) o tempo de relógio de uma hora é atingido. Caso o tempo de relógio seja atingido, a simulação é interrompida e consideramos que houve ao menos um impasse, já que todas as simulações terminam em média em até  $17,5 \text{ min}$ . A escala de tempo da simulação foi acelerada em três vezes em relação ao

tempo de relógio, assim, uma hora de relógio equivale a três horas de simulação.

Simulamos três estratégias de prevenção de colisões para comparação. Quando um obstáculo está a 100 m ou menos de um drone, o drone calcula se haverá uma colisão e pode tomar um desvio em caso positivo. Na primeira estratégia de desvio, chamada *fazer nada*, o drone mantém sua rota inicial. A segunda estratégia foi o algoritmo *Seo et al* e a terceira estratégia foi o nosso algoritmo, GeoDrone. Avaliamos o número de drones colidentes e a etapa da viagem que o drone estava quando colidiu. Para isso, executamos cada simulação 30 vezes e reportamos a média e o intervalo de confiança de 95% de cada medida.

## 7. Resultados

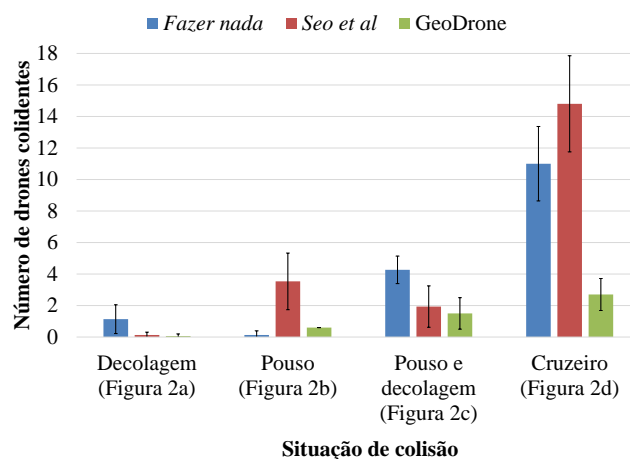
A Figura 5 mostra a média e o intervalo de confiança de 95% do número de drones que colidiram para cada frota e estratégia de prevenção de colisão. Para frotas com dois drones, ambas as estratégias *Seo et al* e GeoDrone eliminam as colisões, mostrando que são estratégias adequadas para evitar colisões entre dois drones. Não há diferença estatística significativa entre todas as estratégias para frotas com 5 ou 10 drones. Também não há diferença estatística significativa entre *Seo et al* e *fazer nada* para frotas com 20 e 50 drones. Contudo, o algoritmo *Seo et al* causa 1,25 e 1,36 mais colisões do que a estratégia *fazer nada* para frotas com 100 e 200 drones, respectivamente. Nossa estratégia GeoDrone reduz significativamente o número de colisões a partir de frotas com 20 drones, entre 3,6 vezes e 25 vezes em relação a *fazer nada* e 4,5 vezes e 14,5 vezes em relação a *Seo et al*. Assim, mostramos a eficácia da nossa estratégia GeoDrone e que ele é o melhor algoritmo para prevenção de colisões entre as estratégias testadas no cenário de serviço de entregas aéreas. Embora no artigo de Seo et al. [2017] não ocorreram colisões, no nosso cenário de entrega de encomendas, que possui uma densidade de drones maior, a estratégia *Seo et al* mostra-se pior que não realizar desvios.



**Figura 5. Número de drones colidentes para tamanhos diferentes de frotas.**

A Figura 6 mostra a média e o intervalo de confiança de 95% do número de drones colidentes nas quatro situações de colisão mostradas na Figura 2 por estratégia de prevenção de colisão para a frota com 200 drones. A maior parte das colisões ocorrem quando ambos os drones estão em cruzeiro nas três estratégias. Contudo, enquanto não há diferença estatística significativa entre *fazer nada* e *Seo et al* – com a estratégia *Seo et*

al causando um número maior de colisões na média –, nossa estratégia GeoDrone reduz o número de colisões em 4,07 vezes em relação a *fazer nada* e 5,48 vezes em relação a *Seo et al*. Assim, esse resultado mostra que a estratégia proposta GeoDrone apresenta uma melhoria significativa no número de colisões em cruzeiro em relação às estratégias *fazer nada* e *Seo et al* para o cenário considerado.



**Figura 6. Número de drones colidentes por situação de colisão para frota com 200 drones.**

Na situação de colisão representada na Figura 2a, quando um drone está decolando e o outro está em cruzeiro, as três estratégias não mostram diferenças significativas na prevenção de colisão. Na situação de colisão em que um drone está pousando e o outro está em cruzeiro (Figura 2b), *fazer nada* e GeoDrone não têm diferenças significativas entre si, enquanto *Seo et al* aumenta o número de colisões em 5,89 vezes em relação a GeoDrone. Por fim, na situação de colisão em que um drone está pousando e o outro está decolando (Figura 2c), *Seo et al* e GeoDrone reduzem ligeiramente o número de colisões, mas não têm diferenças significativas entre si. Embora a estratégia GeoDrone não foi projetada para reduzir colisões nas situações que envolvem pouso e decolagem (casos das Figuras 2a, 2b e 2c), ela indiretamente melhora o número de colisões em alguns desses casos, o que demonstra mais uma vantagem para sua adoção.

Também analisamos a média e o intervalo de confiança de 95% do número de drones detectados no momento da colisão de um drone para as frotas de 200 drones, independentemente de os drones detectados estarem em rota de colisão com o drone colidente. Para a estratégia *fazer nada*, este valor é de  $2,12 \pm 0,11$  drones, para *Seo et al*,  $2,79 \pm 0,12$  drones e, para GeoDrone,  $2,94 \pm 0,44$  drones. Enquanto as estratégias *Seo et al* e GeoDrone não têm diferenças significativas, as colisões com o uso dessas duas estratégias ocorrem quando há mais drones dentro do raio de detecção de um drone colidente do que *fazer nada*, o que pode indicar que as colisões com essas estratégias são mais complexas e requerem estratégias mais sofisticadas de prevenção de colisões.

Por fim, calculamos a média e o intervalo de confiança de 95% do tempo de execução das simulações com as frotas de 200 drones. As simulações com a estratégia *fazer nada* demoram em média  $11,8 \pm 0,031$  min, a estratégia *Seo et al*,  $17,5 \pm 5,13$  min, e GeoDrone,  $15,9 \pm 2,3$  min. As estratégias *fazer nada* e GeoDrone não produzem impasses no cenário simulado, mas a estratégia *Seo et al* produz 10% de execuções com

impasses. Novamente, as estratégias *Seo et al* e GeoDrone não têm diferenças significativas no tempo total de execução, mas seus resultados podem indicar que estratégias mais eficientes podem ser utilizadas para diminuir o tempo total das viagens.

## 8. Discussão

Analizamos a melhoria que as características propostas no GeoDrone efetuaram através dos resultados para a frota com 100 drones. A característica que apresentou o melhor desempenho foi a garantia da manobrabilidade de  $150^\circ/s$ , que diminuiu as colisões em 4,4 vezes. Em seguida, a definição de uma velocidade inicial aleatória entre a velocidade máxima do drone e metade dessa velocidade para cada drone proporcionou uma redução nas colisões de 2,24 vezes. Embora a característica de alterar a velocidade para um valor aleatório entre a velocidade inicial e metade dessa velocidade em caso de colisão detectada não tenha demonstrado resultados estatisticamente diferentes, ela eliminou os impasses que ocorreram no *Seo et al*. Assim, a sincronização do sistema foi interrompida no GeoDrone. Intuitivamente, se todos os drones voarem em sua velocidade máxima, as viagens seriam mais rápidas. Contudo, as simulações da estratégia *Seo et al*, na qual os drones viajaram a suas velocidades máximas durante toda a viagem, resultaram em um tempo total de execução maior. Além da menor eficiência, drones de modelos diferentes têm velocidades máximas diferentes, uma heterogeneidade que seria esperada de um cenário de entregas.

## 9. Conclusão

Neste trabalho, propusemos uma estratégia geométrica de prevenção de colisões para o cenário de serviço de entregas por drones proposto anteriormente [de Oliveira et al., 2021]. Avaliamos as colisões sob frotas de tamanhos crescentes, variando de 2 a 200 drones, e três estratégias quando um drone entra em rota de colisão com outro drone: *fazer nada*, quando os drones mantêm sua rota inicial, a estratégia geométrica proposta por Seo et al. [2017] e GeoDrone, nossa proposta de estratégia geométrica. GeoDrone apresentou o menor número de colisões para frotas com 20 drones ou mais e não houve diferenças significativas em relação às outras estratégias para frotas menores. Também mostramos que a maior redução ocorre quando os drones colidem em cruzeiro, que foi o objetivo deste trabalho. Como trabalhos futuros, controlaremos de forma centralizada os pousos e decolagens em cada drone ponto e simularemos cenários com drones cujas velocidades variam de acordo com o tipo da carga e a distância. Por fim, consideraremos os recursos computacionais presentes em drones para calcular o custo e a viabilidade das estratégias de prevenção de colisões.

## Agradecimentos

Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes, financiado por CNPq (proc. 465446/2014-0), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e FAPESP (procs. 14/50937-1, 15/24485-9 e 2020/14771-2), e da UFABC.

## Referências

Al-Mousa, A., Sababha, B. H., Al-Madi, N., Barghouthi, A., and Younis, R. (2019). UT-Sim: A framework and simulator for UAV air traffic integration, control, and communication. *International Journal of Advanced Robotic Systems*, 16(5):1729881419870937.

- Ayamga, M., Akaba, S., and Nyaaba, A. A. (2021). Multifaceted applicability of drones: A review. *Technological Forecasting and Social Change*, 167:120677.
- Chen, K.-W., Xie, M.-R., Chen, Y.-M., Chu, T.-T., and Lin, Y.-B. (2022). DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14.
- Choi, D. and Kim, D. (2020). Enhanced Potential Field-Based Collision Avoidance for Unmanned Aerial Vehicles in a Dynamic Environment. *Journal of Intelligent & Robotic Systems*, pages 1–7.
- Craighead, J., Murphy, R., Burke, J., and Goldiez, B. (2007). A survey of commercial open source unmanned vehicle simulators. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 852–857.
- de Oliveira, F. M. C., Bittencourt, L. F., and Kamienski, C. A. (16 de agosto de 2021). Prevenção de colisões em serviços de entregas por drones em cidades inteligentes. In *V Workshop de Computação Urbana (CoUrb 2021)*, pages 182–195, Porto Alegre, RS, Brasil. Sociedade Brasileira de Computação (SBC).
- DJI (2022). Comparação de Drones de Consumo. <https://www.dji.com/br/products/compare-consumer-drones>. Acessado em 19 de abril de 2022.
- Frachtenberg, E. (2019). Practical drone delivery. *Computer*, 52(12):53–57.
- Gageik, N., Benz, P., and Montenegro, S. (2015). Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors. *IEEE Access*, 3:599–609.
- Guan, X., Lyu, R., Shi, H., and Chen, J. (2020). A survey of safety separation management and collision avoidance approaches of civil UAS operating in integration national airspace system. *Chinese Journal of Aeronautics*, 33(11):2851–2863. SI: Emerging Technologies of Unmanned Aerial Vehicles.
- Hentati, A. I., Krichen, L., Fourati, M., and Fourati, L. C. (2018). Simulation tools, environments and frameworks for uav systems performance analysis. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1495–1500.
- Kasliwal, A., Furbush, N. J., Gawron, J. H., McBride, J. R., Wallington, T. J., De Kleine, R. D., Kim, H. C., and Keoleian, G. A. (2019). Role of flying cars in sustainable mobility. *Nature communications*, 10(1):1–9.
- Kleinbekman, I. C., Mitici, M. A., and Wei, P. (2018). eVTOL arrival sequencing and scheduling for on-demand urban air mobility. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–7. IEEE.
- Mairaj, A., Baba, A. I., and Javaid, A. Y. (2019). Application specific drone simulators: Recent advances and challenges. *Simulation Modelling Practice and Theory*, 94:100–117.
- Seo, J., Kim, Y., Kim, S., and Tsourdos, A. (2017). Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight. *IEEE Transactions on Aerospace and Electronic Systems*, 53(6):2718–2734.
- Sánchez, P., Casado, R., and Bermúdez, A. (2020). Real-Time Collision-Free Navigation of Multiple UAVs Based on Bounding Boxes. *Electronics*, 9(10).
- Yang, X. and Wei, P. (2021). Autonomous free flight operations in urban air mobility with computational guidance and collision avoidance. *IEEE Transactions on Intelligent Transportation Systems*.
- Yasin, J. N., Mohamed, S. A. S., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. *IEEE Access*, 8:105139–105155.