ATP 2 - Computação Natural

Aluno: Fabricio Magalhães Sena

In [1]:
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import pytesseract
import cv2
from sklearn.cluster import KMeans
import math
import time
from itertools import chain
from typing import Any, Callable, List, Tuple, Union
import pandas as pd
from itertools import product
import numpy as np
import random
```

Helpers

In [2]:
```
def show_map_with_route_points(path, w=12, h=8, image=None):
    if isinstance(path, dict):
        path = list(path.values())
    if isinstance(path[0][0], str):
        path = [item[1] for item in path]
    plt.imshow(image)
    for x0, y0 in path:
        # y* = yellow star for starting point
        plt.plot(x0, y0, 'y*', markersize=15)
    plt.axis("off")
    fig = plt.gcf()
    fig.set_size_inches([w, h])


def show_path(path, starting_city=None, w=12, h=8, image=None, title=None
    plt.clf()

    if isinstance(path, dict):
        path = list(path.values())
    if isinstance(path[0][0], str):
        path = [item[1] for item in path]

    starting_city = starting_city or path[0]
    x, y = list(zip(*path))
    (x0, y0) = starting_city
    if title:
        plt.title(title)
    plt.imshow(image)
    plt.plot(x + x[:1], y + y[:1])
    plt.axis("off")
    fig = plt.gcf()
    fig.set_size_inches([w, h])

    if title:
```

```
        image_path = f'./output/{title}.png'
        plt.savefig(image_path, bbox_inches='tight', pad_inches=0)
        return image_path


def polyfit_plot(x, y, deg, **kwargs):
    coefficients = np.polyfit(x, y, deg, **kwargs)
    poly = np.poly1d(coefficients)
    new_x = np.linspace(x[0], x[-1])
    new_y = poly(new_x)
    plt.plot(x, y, "o", new_x, new_y)
    plt.xlim([x[0]-1, x[-1] + 1])

    terms = []
    for p, c in enumerate(reversed(coefficients)):
        term = str(round(c, 1))
        if p == 1:
            term += 'x'
        if p >= 2:
            term += 'x^'+str(p)
        terms.append(term)
    plt.title(" + ".join(reversed(terms)))


def calculate_distance(xy1, xy2) -> float:
    if isinstance(xy1[0], str):
        xy1 = xy1[1]
        xy2 = xy2[1]
    return math.sqrt((xy1[0]-xy2[0])**2 + (xy1[1]-xy2[1])**2)


def path_distance(path) -> int:
    if isinstance(path, dict):
        path = list(path.values())
    if isinstance(path[0][0], str):
        path = [item[1] for item in path]
    return int(sum(
        [calculate_distance(path[i],  path[i+1]) for i in range(len(path)
        # include cost of return journey
        + [calculate_distance(path[-1], path[0])]
    ))
```

In [3]:

```
class AntColonySolver:
    def __init__(self,
                 cost_fn:                Callable[[Any, Any], Union[floa

                 time=0,                 # run for a fixed amount of tim
                 min_time=0,             # minimum runtime
                 timeout=0,              # maximum time in seconds to ru
                 stop_factor=2,          # how many times to redouble ef
                 min_round_trips=10,     # minimum number of round trips
                 max_round_trips=0,      # maximum number of round trips
                 min_ants=0,             # Total number of ants to use
                 max_ants=0,             # Total number of ants to use

                 # this is the bottom of the near-optimal range for numpy
                 ant_count=64,
                 ant_speed=1,            # how many steps do ants travel

                 distance_power=1,       # power to which distance affec
```

```
                pheromone_power=1.25,    # power to which differences in
                decay_power=0,           # how fast do pheromones decay
                reward_power=0,          # relative pheromone reward bas
                best_path_smell=2,       # queen multiplier for pheromon
                # amount of starting pheromones [0 defaults to `10**self
                start_smell=0,

                verbose=False,

                ):
        assert callable(cost_fn)
        self.cost_fn = cost_fn
        self.time = int(time)
        self.min_time = int(min_time)
        self.timeout = int(timeout)
        self.stop_factor = float(stop_factor)
        self.min_round_trips = int(min_round_trips)
        self.max_round_trips = int(max_round_trips)
        self.min_ants = int(min_ants)
        self.max_ants = int(max_ants)

        self.ant_count = int(ant_count)
        self.ant_speed = int(ant_speed)

        self.distance_power = float(distance_power)
        self.pheromone_power = float(pheromone_power)
        self.decay_power = float(decay_power)
        self.reward_power = float(reward_power)
        self.best_path_smell = float(best_path_smell)
        self.start_smell = float(start_smell or 10**self.distance_power)

        self.verbose = int(verbose)
        self._initalized = False

        if self.min_round_trips and self.max_round_trips:
            self.min_round_trips = min(
                self.min_round_trips, self.max_round_trips)
        if self.min_ants and self.max_ants:
            self.min_ants = min(self.min_ants, self.max_ants)

    def solve_initialize(
            self,
            problem_path: List[Any],
    ) -> None:
        # Cache of distances between nodes
        self.distances = {
            source: {
                dest: self.cost_fn(source, dest)
                for dest in problem_path
            }
            for source in problem_path
        }

        # Cache of distance costs between nodes - division in a tight loo
        self.distance_cost = {
            source: {
                dest: 1 / (1 + self.distances[source]
                           [dest]) ** self.distance_power
                for dest in problem_path
            }
```

```
            for source in problem_path
        }

        # This stores the pheromone trail that slowly builds up
        self.pheromones = {
            source: {
                # Encourage the ants to start exploring in all directions
                dest: self.start_smell
                for dest in problem_path
            }
            for source in problem_path
        }

        # Sanitise input parameters
        if self.ant_count <= 0:
            self.ant_count = len(problem_path)
        if self.ant_speed <= 0:
            self.ant_speed = np.median(
                list(chain(*[d.values() for d in self.distances.values()]
        self.ant_speed = int(max(1, self.ant_speed))

        # Heuristic Exports
        self.ants_used = 0
        self.epochs_used = 0
        self.round_trips = 0
        self._initalized = True

    def solve(self,
              problem_path: List[Any],
              restart=False,
              ) -> List[Tuple[int, int]]:
        if restart or not self._initalized:
            self.solve_initialize(problem_path)

        # Here come the ants!
        ants = {
            "distance":    np.zeros((self.ant_count,)).astype('int32'),
            "path":        [[problem_path[0]] for n in range(self.ant_cou
            "remaining":   [set(problem_path[1:]) for n in range(self.ant
            "path_cost":   np.zeros((self.ant_count,)).astype('int32'),
            "round_trips": np.zeros((self.ant_count,)).astype('int32'),
        }

        best_path = None
        best_path_cost = np.inf
        best_epochs = []
        epoch = 0
        time_start = time.perf_counter()
        while True:
            epoch += 1

            # Vectorized walking of ants
            # Small optimization here, testing against `> self.ant_speed`
            #       avoids computing ants_arriving in the main part of th
            ants_travelling = (ants['distance'] > self.ant_speed)
            ants['distance'][ants_travelling] -= self.ant_speed
            if all(ants_travelling):
                continue  # skip termination checks until the next ant ar

            # Vectorized checking of ants arriving
```

```
            ants_arriving = np.invert(ants_travelling)
            ants_arriving_index = np.where(ants_arriving)[0]
            for i in ants_arriving_index:

                # ant has arrived at next_node
                this_node = ants['path'][i][-1]
                next_node = self.next_node(ants, i)
                ants['distance'][i] = self.distances[this_node][next_node
                ants['remaining'][i] = ants['remaining'][i] - {this_node}
                ants['path_cost'][i] = ants['path_cost'][i] + \
                    ants['distance'][i]
                ants['path'][i].append(next_node)

                # ant has returned home to the colony
                if not ants['remaining'][i] and ants['path'][i][0] == ant
                    self.ants_used += 1
                    self.round_trips = max(
                        self.round_trips, ants["round_trips"][i] + 1)

                    # We have found a new best path - inform the Queen
                    was_best_path = False
                    if ants['path_cost'][i] < best_path_cost:
                        was_best_path = True
                        best_path_cost = ants['path_cost'][i]
                        best_path = ants['path'][i]
                        best_epochs += [epoch]
                        if self.verbose:
                            print({
                                "path_cost":   int(ants['path_cost'][i]),
                                "ants_used":   self.ants_used,
                                "epoch":       epoch,
                                "round_trips": ants['round_trips'][i] + 1
                                "clock":       int(time.perf_counter() -
                            })

                    # leave pheromone trail
                    # doing this only after ants arrive home improves ini
                    #  * self.round_trips has the effect of decaying old
                    # ** self.reward_power = -3 has the effect of encoura
                    #                          in combination with doubl
                    reward = 1
                    if self.reward_power:
                        reward *= ((best_path_cost /
                                    ants['path_cost'][i]) ** self.reward_p
                    if self.decay_power:
                        reward *= (self.round_trips ** self.decay_power)
                    for path_index in range(len(ants['path'][i]) - 1):
                        this_node = ants['path'][i][path_index]
                        next_node = ants['path'][i][path_index+1]
                        self.pheromones[this_node][next_node] += reward
                        self.pheromones[next_node][this_node] += reward
                        if was_best_path:
                            # Queen orders to double the number of ants f
                            self.pheromones[this_node][next_node] *= self
                            self.pheromones[next_node][this_node] *= self

                    # reset ant
                    ants["distance"][i] = 0
                    ants["path"][i] = [problem_path[0]]
                    ants["remaining"][i] = set(problem_path[1:])
```

```
                    ants["path_cost"][i] = 0
                    ants["round_trips"][i] += 1

            # Do we terminate?

            # Always wait for at least 1 solutions (note: 2+ solutions ar
            if not len(best_epochs):
                continue

            # Timer takes priority over other constraints
            if self.time or self.min_time or self.timeout:
                clock = time.perf_counter() - time_start
                if self.time:
                    if clock > self.time:
                        break
                    else:
                        continue
                if self.min_time and clock < self.min_time:
                    continue
                if self.timeout and clock > self.timeout:
                    break

            # First epoch only has start smell - question: how many epoch
            if self.min_round_trips and self.round_trips < self.min_round
                continue
            if self.max_round_trips and self.round_trips >= self.max_roun
                break

            # This factor is most closely tied to computational power
            if self.min_ants and self.ants_used < self.min_ants:
                continue
            if self.max_ants and self.ants_used >= self.max_ants:
                break

            # Lets keep redoubling our efforts until we can't find anythi
            if self.stop_factor and epoch > (best_epochs[-1] * self.stop_
                break

            # Nothing else is stopping us: Queen orders the ants to conti
            if True:
                continue

        # We have (hopefully) found a near-optimal path, report back to t
        self.epochs_used = epoch
        self.round_trips = np.max(ants["round_trips"])
        return best_path

    def next_node(self, ants, index):
        this_node = ants['path'][index][-1]

        weights = []
        weights_sum = 0
        if not ants['remaining'][index]:
            return ants['path'][index][0]  # return home
        for next_node in ants['remaining'][index]:
            if next_node == this_node:
                continue
            reward = (
                self.pheromones[this_node][next_node] ** self.pheromone_p
                # Prefer shorter paths
```

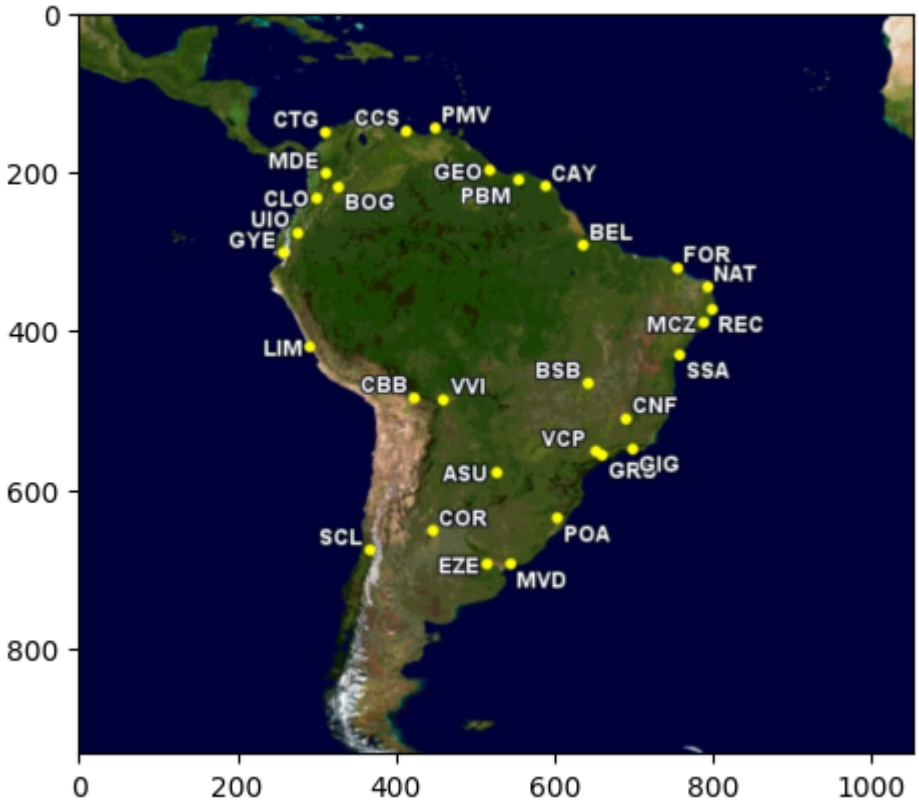
```
                * self.distance_cost[this_node][next_node]
            )
            weights.append((reward, next_node))
            weights_sum += reward

        # Pick a random path in proportion to the weight of the pheromone
        rand = random.random() * weights_sum
        for (weight, next_node) in weights:
            if rand > weight:
                rand -= weight
            else:
                break
        return next_node

    @staticmethod
    def run(cities, map_image=None, verbose=False, plot=False, label={},
        solver = AntColonySolver(cost_fn=calculate_distance, verbose=verb
        start_time = time.perf_counter()
        result = solver.solve(cities)
        stop_time = time.perf_counter()
        if label:
            kwargs = {**label, **kwargs}

        for key in ['verbose', 'plot', 'animate', 'label', 'min_time', 'm
            if key in kwargs:
                del kwargs[key]
        print("N={:<3d} | {:5.0f} -> {:4.0f} | {:4.0f}s | ants: {:5d} | t
            .format(len(cities), path_distance(cities), path_distance(res
            + " ".join([f"{k}={v}" for k, v in kwargs.items()])
            )
        if plot:
            show_path(result, image=map_image)
        return result
```

Questão 1:

Vocês vão substituir a imagem do mapa atual por outro de sua escolha, como um mapa de cidades ou de pontos de interesse. Com base nesse novo mapa, selecionem dez pontos que serão os locais, cidades ou estados que vocês vão otimizar na rota do caixeiro-viajante. Criem um dicionário com os dez locais e suas coordenadas X e Y, seguindo o formato do exemplo fornecido no notebook. Observação: se tiverem dificuldade em substituir a imagem, podem utilizar a mesma imagem que já está no notebook, porém o indicado é trocar a imagem.

In [4]:
```
image_path = './input/latim-america-airports.png'
img = mpimg.imread(image_path)
plt.imshow(img)
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x286f96750>
```



Eu escolhi um problema real de logistica, a distribuição de voos entre cidades do Brasil para otimizar a rota de uma entrega de pacotes baseado apenas na distância, sem considerar o peso de cada cidade ou a quantidade de pacotes a serem entregues.

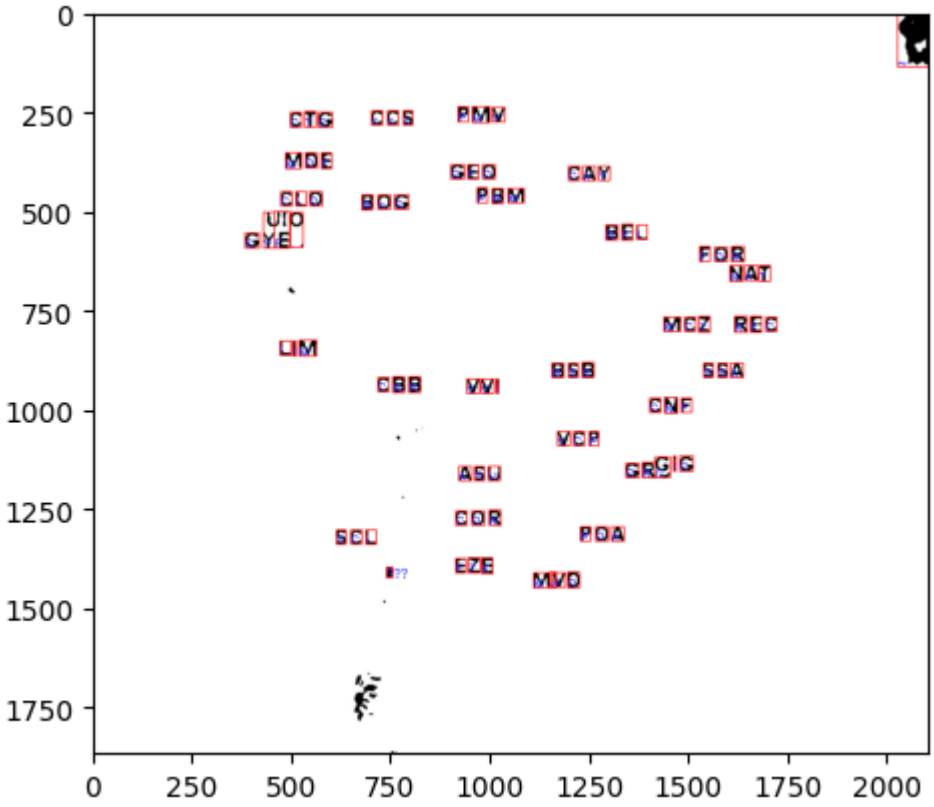
Abaixo irei extrair usando visão computacional, todos os aeroportos do Brasil e suas coordenadas nas imagens para utilizar como cidades no problema do caixeiro viajante ralatado na atividade somativa.

In [5]:
```python
class ImageTextCoordsExtractor:
    def get_colors(self, image, num_colors=10):
        # to numpy img
        image = np.array(image)
        image = image.reshape((image.shape[0] * image.shape[1], 3))
        kmeans = KMeans(n_clusters=num_colors)
        kmeans.fit(image)
        colors = kmeans.cluster_centers_
        colors_as_rgb = [tuple(map(int, color)) for color in colors]
        return colors_as_rgb

    def show_colors(self, colors):
        fig, ax = plt.subplots(1, 1, figsize=(10, 1))
        ax.imshow([colors], aspect='auto')
        ax.set_axis_off()
        plt.show()

    def preprocess_image(self, image):
        white_level = [
            190,
            190,
            190
        ]
```

```
        image = image.convert("RGB")
        image = np.array(image)
        image[np.where((image < white_level).all(axis=2))] = [0, 0, 0]

        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
        image = Image.fromarray(image)

        colors = self.get_colors(image)
        colors = sorted(colors, key=lambda x: math.prod(x))
        self.show_colors(colors)

        image = np.array(image)

        def rgb_product(pixel):
            return math.prod(pixel)

        image_product = np.apply_along_axis(rgb_product, 2, image)
        white_level = rgb_product(colors[-4])

        image[np.where(image_product < white_level)] = [0, 0, 0]

        gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)

        _, thresh_image = cv2.threshold(
            blurred_image, 128, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

        rescaled_image = cv2.resize(
            thresh_image, None, fx=2, fy=2, interpolation=cv2.INTER_CUBIC
        rescaled_image = cv2.cvtColor(rescaled_image, cv2.COLOR_GRAY2BGR)
        rescaled_image = cv2.bitwise_not(rescaled_image)

        return rescaled_image

    def extract(self, image_path):
        image = Image.open(image_path)
        image_processed = self.preprocess_image(image)
        pil_image = Image.fromarray(image_processed)
        boxes = pytesseract.image_to_boxes(pil_image)

        letters_coords = []

        for box in boxes.splitlines():
            box = box.split(' ')
            letter, x, y, w, h = box[0], int(box[1]), int(box[2]), int(bo
            y = image_processed.shape[0] - y
            h = image_processed.shape[0] - h
            # put a description with the label of value
            cv2.putText(image_processed, box[0], (x, y),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (50, 50, 255), 2)
            cv2.rectangle(image_processed,
                        (x, y),
                        (w, h),
                        (255, 0, 0), 2)
            O_letter_variants = letter in ['0', '®', '°']
            if O_letter_variants:
                letter = 'O'
            letters_coords.append((letter, (x, y, w, h)))

        return letters_coords, image_processed
```

```
letters_coords, image = ImageTextCoordsExtractor().extract(image_path)

plt.imshow(image)
```

Out[5]: <matplotlib.image.AxesImage at 0x289d8acf0>



Os aeroportos marcados de vermelho na imagem são os aeroportos que serão utilizados como cidades no problema do caixeiro viajante. Abaixo estarei fazendo um algoritmo para extrair as coordenadas desses aeroportos na imagem.

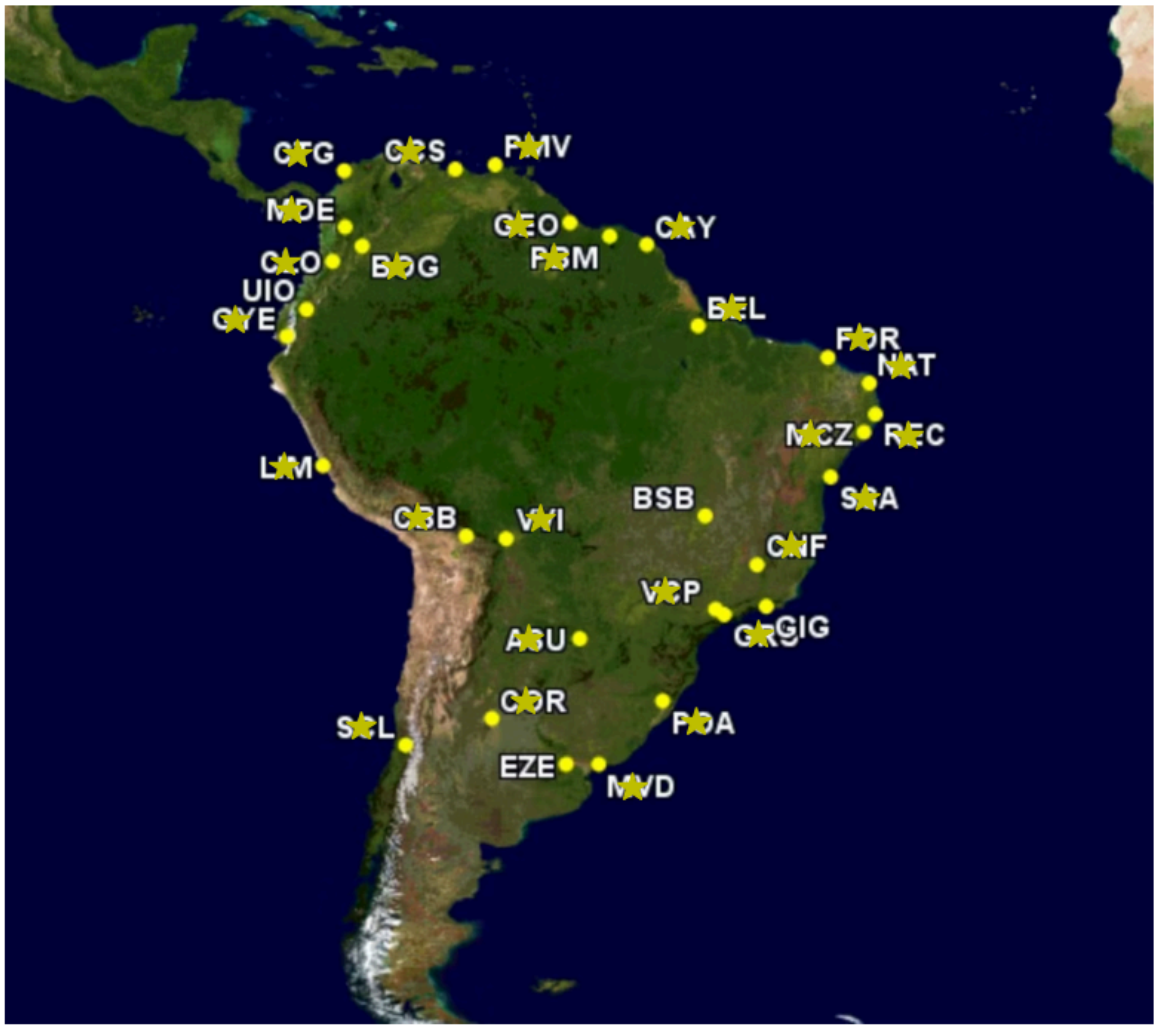
In [6]:
```
class AirportsDiscover:
    def get_only_alphabetic_chars(self, city):
        return "".join([letter for letter in city if letter.isalpha()])

    def plot_cities(self, cities_coords):
        print('Airports: ')
        for city, coords in cities_coords.items():
            print(f"{city}: {coords}")

    def run(self, letters_coords: dict):
        letters_in_the_same_Y = {}

        for letter, (x, y, x1, y1) in letters_coords:
            # we need have a margin error of 10 pixels
            for y0, letters in letters_in_the_same_Y.items():
                if abs(y0 - y) < 20:
                    letters.append((letter, (x, y, x1, y1)))
                    break
            else:
```

```
                letters_in_the_same_Y[y] = [(letter, (x, y, x1, y1))]

        # each group of 3 letters is a city
        letter_coords_grouped = []
        for y, letters in letters_in_the_same_Y.items():
            for i in range(0, len(letters), 3):
                city = "".join([letter for letter, _ in letters[i:i+3]])
                letter_coords_grouped.append(
                    (city,
                     (
                        letters[i][1][0],
                        letters[i][1][1],
                    )))

        airports_coords = dict()

        for city, coords in letter_coords_grouped:
            city = self.get_only_alphabetic_chars(city)
            if len(city) == 3:
                airports_coords[city.upper()] = (coords[0] + 40, coords[1

        self.plot_cities(airports_coords)

        return airports_coords

airports_coords = AirportsDiscover().run(letters_coords)
```

```
Airports:
CTG: (536, 270)
CCS: (740, 264)
PMV: (958, 256)
MDE: (524, 374)
GEO: (940, 400)
CAY: (1236, 404)
CLO: (512, 468)
BOG: (716, 478)
PBM: (1006, 460)
GYE: (422, 574)
BEL: (1330, 554)
FOR: (1566, 608)
NAT: (1642, 658)
MCZ: (1476, 784)
REC: (1654, 786)
LIM: (510, 844)
SSA: (1576, 900)
CBB: (756, 936)
VVI: (980, 940)
CNF: (1440, 988)
VCP: (1208, 1072)
ASU: (960, 1160)
GRG: (1380, 1152)
COR: (952, 1272)
SCL: (652, 1320)
POA: (1266, 1312)
MVV: (1148, 1430)
```

As cordenadas acima serão as cordenadas utilizadas como ponto de despacho dos pacotes pelo o avião.

In [7]:
```
original_image = Image.open(image_path)
original_image = np.array(original_image)
original_image = cv2.resize(original_image, None, fx=2, fy=2, interpolati

show_map_with_route_points(airports_coords, w=10, h=10, image=original_im
```



In [8]:
```
route_point_coords = list(sorted(airports_coords.items()))
```

Questão 2:

Escolham um parâmetro do algoritmo para alterar e observem como isso impacta a otimização da rota. Algumas opções de parâmetros incluem distance_power, pheromone_power, min_ants e ant_count. Consultem o link sugerido para ideias de valores. Depois da alteração, deve-se apresentar a figura resultante (com o caminho/rota traçado) e comentar o resultado obtido.

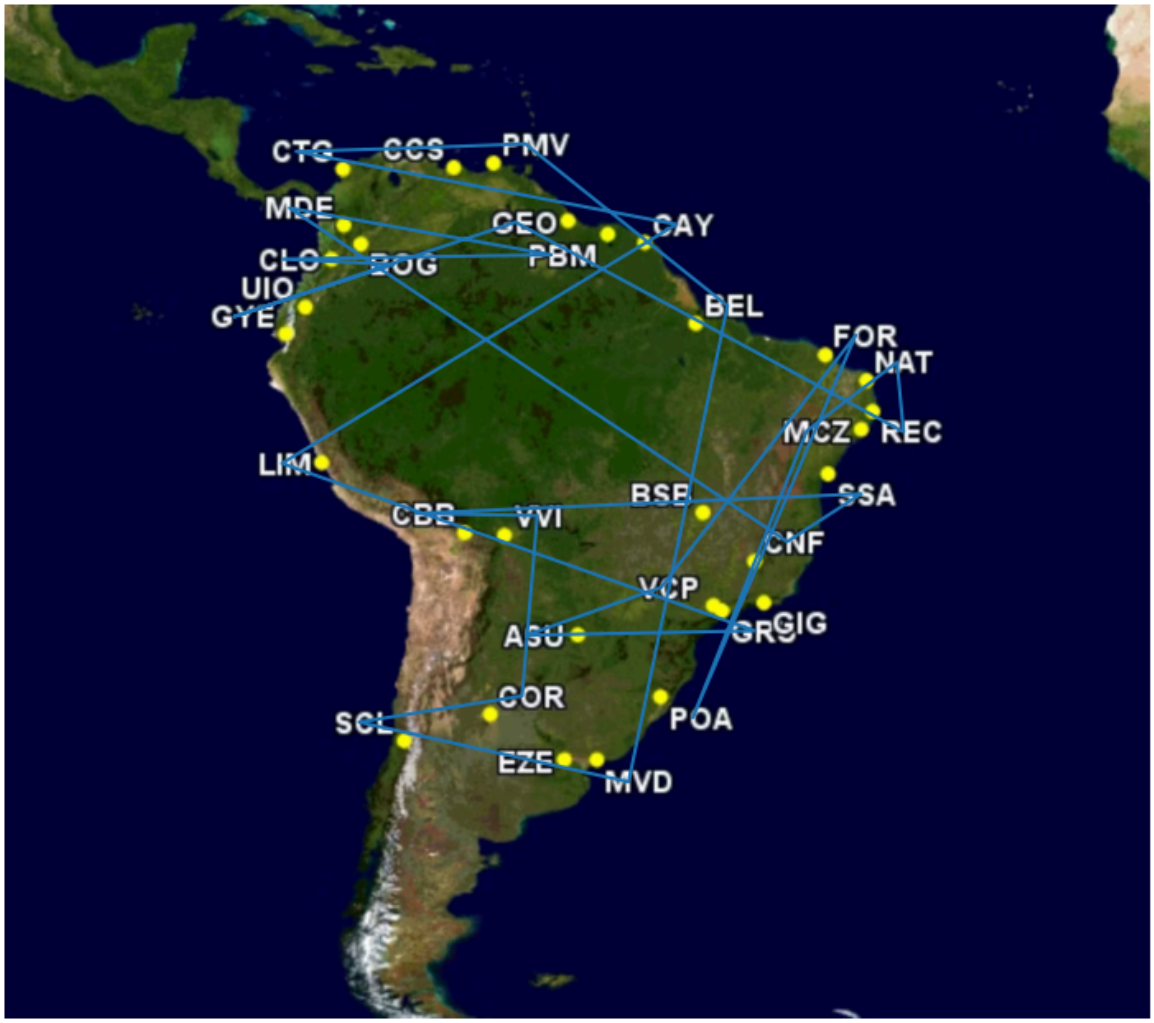
In [9]:
```
results = AntColonySolver.run(
    route_point_coords,
    map_image=original_image,
    distance_power=1,
    verbose=True,
    plot=True
)
```

{'path_cost': 11636, 'ants_used': 1, 'epoch': 10716, 'round_trips': 1, 'clock': 0}
{'path_cost': 9877, 'ants_used': 65, 'epoch': 21759, 'round_trips': 2, 'clock': 0}
{'path_cost': 8175, 'ants_used': 68, 'epoch': 22498, 'round_trips': 2, 'clock': 0}
{'path_cost': 7436, 'ants_used': 133, 'epoch': 32630, 'round_trips': 3, 'clock': 0}
{'path_cost': 7184, 'ants_used': 195, 'epoch': 40787, 'round_trips': 4, 'clock': 0}
{'path_cost': 6571, 'ants_used': 203, 'epoch': 42160, 'round_trips': 4, 'clock': 0}
{'path_cost': 6381, 'ants_used': 321, 'epoch': 56049, 'round_trips': 6, 'clock': 0}
{'path_cost': 6281, 'ants_used': 324, 'epoch': 56714, 'round_trips': 6, 'clock': 0}
{'path_cost': 6189, 'ants_used': 379, 'epoch': 61509, 'round_trips': 6, 'clock': 0}
{'path_cost': 5885, 'ants_used': 385, 'epoch': 63277, 'round_trips': 7, 'clock': 0}
{'path_cost': 5616, 'ants_used': 405, 'epoch': 65365, 'round_trips': 7, 'clock': 0}
{'path_cost': 5514, 'ants_used': 453, 'epoch': 70107, 'round_trips': 8, 'clock': 0}
{'path_cost': 5337, 'ants_used': 462, 'epoch': 70571, 'round_trips': 8, 'clock': 0}
N=27 | 20097 -> 5349 | 1s | ants: 1236 | trips: 20 | distance_power=1

Sem utilizar distance power:

In [16]:
```
results = AntColonySolver.run(
    route_point_coords,
    map_image=original_image,
    distance_power=0,
    verbose=True,
    plot=True
)
```

```
{'path_cost': 15744, 'ants_used': 1, 'epoch': 14900, 'round_trips': 1, 'cl
ock': 0}
{'path_cost': 15575, 'ants_used': 2, 'epoch': 15157, 'round_trips': 1, 'cl
ock': 0}
{'path_cost': 13722, 'ants_used': 65, 'epoch': 29745, 'round_trips': 2, 'c
lock': 0}
{'path_cost': 13516, 'ants_used': 204, 'epoch': 65897, 'round_trips': 4,
'clock': 0}
N=27  | 20097 -> 13527 |    1s | ants:   565 | trips:   10 | distance_powe
r=0
```



In [11]:
```
params_to_test = {
    'max_round_trips': [
        100, 1000, 10000
    ],
    'max_ants': [
        1024, 4096
    ],
```

```
    'ant_count': [
        128, 256, 512
    ],
    'distance_power': [
        0, 1
    ],
    'reward_power': [
         0, 1
    ],
    'pheromone_power': [
        1, 2
    ],
}

params = list(product(*params_to_test.values()))
params_possibilities = {}

for index, param in enumerate(params):
    kwargs = dict(zip(params_to_test.keys(), param))
    params_possibilities[index] = kwargs


print(f"Total of {len(params_possibilities)} possibilities")

# print posibilities
for index, kwargs in params_possibilities.items():
    print(f"Index: {index} | Params: {kwargs}")
```

```
Total of 144 possibilities
Index: 0 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 1 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 2 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 3 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 4 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 5 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 6 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 7 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 8 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 9 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_count':
256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 10 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 11 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 12 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 13 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 14 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 15 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 16 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 17 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 18 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 19 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 20 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 21 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 22 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 23 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 24 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 25 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 26 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 27 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 28 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 29 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
```

```
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 30 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 31 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 32 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 33 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 34 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 35 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 36 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 37 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 38 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 39 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 40 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 41 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 42 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 43 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 44 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 45 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 46 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 47 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 48 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 49 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 50 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 51 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 52 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 53 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 54 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 55 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 56 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 57 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 58 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 59 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
```

```
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 60 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 61 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 62 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 63 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 64 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 65 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 66 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 67 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 68 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 69 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 70 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 71 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 72 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 73 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 74 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 75 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 76 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 77 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 78 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 79 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 80 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 81 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 82 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 83 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 84 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 85 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 86 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 87 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 88 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 89 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
```

```
t': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 90 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 91 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 92 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 93 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 94 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 95 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_coun
t': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 96 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 97 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 98 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 99 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 100 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 101 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 102 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 103 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 104 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 105 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 106 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 107 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 108 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 109 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 110 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 111 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 112 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 113 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 114 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 115 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 116 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 117 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 118 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 119 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_cou
```

```
nt': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 120 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 121 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 122 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 123 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 124 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 125 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 126 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 127 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 128 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 129 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 130 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 131 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 132 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 133 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 134 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 135 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
Index: 136 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
Index: 137 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
Index: 138 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
Index: 139 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
Index: 140 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
Index: 141 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
Index: 142 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
Index: 143 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'ant_cou
nt': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
```

Questão 3:

Agora, modifiquem um segundo parâmetro diferente daquele que vocês ajustaram anteriormente e analisem as mudanças na eficiência da otimização da rota. Algumas opções de parâmetros incluem distance_power, pheromone_power, min_ants e ant_count. Consultem o link sugerido para ideias de valores. Depois da alteração, deve-se apresentar a figura resultante (com o caminho/rota traçado) e comentar o resultado obtido.

In [12]:
```
tests_results = {}
points = list(sorted(airports_coords.items()))

for index, param_possibility in params_possibilities.items():
    results = AntColonySolver.run(
        points,
        map_image=original_image,
        **param_possibility
    )
    distance = path_distance(results)
    print(f"Possibility: {index} | Params: {param_possibility} | Distance
    image_result = show_path(results, image=original_image, title=f"Posib

    tests_results[index] = {
        "params": param_possibility,
        "distance": distance,
        "image_result": image_result
    }
```

```
N=27  | 20097 -> 13218 |    1s | ants:  1146 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 0 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 13218
N=27  | 20097 -> 9124 |    1s | ants:  1145 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 1 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 9124
N=27  | 20097 -> 12165 |    1s | ants:  1144 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 pheromo
ne_power=1
Possibility: 2 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 12165
N=27  | 20097 -> 10016 |    1s | ants:  1131 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 3 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 10016
N=27  | 20097 -> 5538 |    0s | ants:  1094 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 4 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5538
N=27  | 20097 -> 5409 |    0s | ants:  1138 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 5 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 5409
N=27  | 20097 -> 5604 |    0s | ants:  1113 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 6 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5604
N=27  | 20097 -> 5466 |    0s | ants:  1140 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 7 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5466
N=27  | 20097 -> 12511 |    1s | ants:  2254 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 8 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 12511
N=27  | 20097 -> 8536 |    1s | ants:  2299 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 9 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_c
ount': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 8536
```

```
N=27  | 20097 -> 12805 |    1s | ants:  2233 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 pheromo
ne_power=1
Possibility: 10 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 12805
N=27  | 20097 -> 7588 |    1s | ants:  2231 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 pheromon
e_power=2
Possibility: 11 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 7588
N=27  | 20097 -> 5630 |    1s | ants:  2151 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 12 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5630
N=27  | 20097 -> 5494 |    1s | ants:  2237 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 13 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 5494
N=27  | 20097 -> 5428 |    1s | ants:  2176 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 14 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5428
N=27  | 20097 -> 5529 |    1s | ants:  2265 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 15 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5529
N=27  | 20097 -> 11490 |    2s | ants:  4468 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 16 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 11490
N=27  | 20097 -> 8768 |    1s | ants:  4520 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 17 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 8768
N=27  | 20097 -> 12526 |    2s | ants:  4510 | trips:   10 | max_round_tri
ps=100 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 pheromo
ne_power=1
Possibility: 18 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 12526
N=27  | 20097 -> 8032 |    1s | ants:  4439 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 pheromon
e_power=2
Possibility: 19 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 8032
```

```
N=27  | 20097 -> 5486 |    1s | ants:  4398 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 20 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5486
N=27  | 20097 -> 5398 |    1s | ants:  4314 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 21 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 5398
N=27  | 20097 -> 5407 |    1s | ants:  4335 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 22 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5407
N=27  | 20097 -> 5370 |    1s | ants:  4541 | trips:   10 | max_round_trip
s=100 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 23 | Params: {'max_round_trips': 100, 'max_ants': 1024, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5370
N=27  | 20097 -> 12078 |    2s | ants:  2482 | trips:   21 | max_round_tri
ps=100 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 24 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 12078
N=27  | 20097 -> 8738 |    2s | ants:  4089 | trips:   33 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 25 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 8738
N=27  | 20097 -> 7438 |    2s | ants:  4096 | trips:   34 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 pheromon
e_power=1
Possibility: 26 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 7438
N=27  | 20097 -> 8427 |    1s | ants:  1568 | trips:   13 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 pheromon
e_power=2
Possibility: 27 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 8427
N=27  | 20097 -> 5446 |    1s | ants:  3681 | trips:   30 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 28 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5446
N=27  | 20097 -> 6181 |    0s | ants:  1147 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 29 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 6181
```

```
N=27  | 20097 -> 5450 |    1s | ants:  2485 | trips:   20 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 30 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5450
N=27  | 20097 -> 5502 |    0s | ants:  1345 | trips:   11 | max_round_trip
s=100 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 31 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5502
N=27  | 20097 -> 10975 |    2s | ants:  4096 | trips:   17 | max_round_tri
ps=100 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 32 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 10975
N=27  | 20097 -> 7715 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 33 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 7715
N=27  | 20097 -> 7434 |    2s | ants:  4096 | trips:   17 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 pheromon
e_power=1
Possibility: 34 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 7434
N=27  | 20097 -> 7208 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 pheromon
e_power=2
Possibility: 35 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 7208
N=27  | 20097 -> 5430 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 36 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5430
N=27  | 20097 -> 5464 |    1s | ants:  3307 | trips:   14 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 37 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 5464
N=27  | 20097 -> 5361 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 38 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5361
N=27  | 20097 -> 5656 |    1s | ants:  2269 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 39 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5656
```

```
N=27  | 20097 -> 11818 |    1s | ants:  4193 | trips:   10 | max_round_tri
ps=100 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 40 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 11818
N=27  | 20097 -> 7873 |    1s | ants:  4502 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 pheromon
e_power=2
Possibility: 41 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 7873
N=27  | 20097 -> 8104 |    2s | ants:  4409 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 pheromon
e_power=1
Possibility: 42 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 8104
N=27  | 20097 -> 7968 |    1s | ants:  4583 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 pheromon
e_power=2
Possibility: 43 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 7968
N=27  | 20097 -> 5407 |    1s | ants:  4288 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pheromon
e_power=1
Possibility: 44 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1}
| Distance: 5407
N=27  | 20097 -> 5366 |    1s | ants:  4540 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pheromon
e_power=2
Possibility: 45 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2}
| Distance: 5366
N=27  | 20097 -> 5481 |    1s | ants:  4468 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pheromon
e_power=1
Possibility: 46 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1}
| Distance: 5481
N=27  | 20097 -> 5559 |    1s | ants:  4500 | trips:   10 | max_round_trip
s=100 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pheromon
e_power=2
Possibility: 47 | Params: {'max_round_trips': 100, 'max_ants': 4096, 'ant_
count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 2}
| Distance: 5559
N=27  | 20097 -> 13287 |    1s | ants:  1116 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 pherom
one_power=1
Possibility: 48 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 13287
N=27  | 20097 -> 10965 |    1s | ants:  1146 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 49 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 10965
```

```
N=27  | 20097 -> 9194 |    1s | ants:  1120 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 pheromo
ne_power=1
Possibility: 50 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 9194
N=27  | 20097 -> 7672 |    1s | ants:  1141 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 51 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 7672
N=27  | 20097 -> 5597 |    0s | ants:  1111 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 52 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5597
N=27  | 20097 -> 5327 |    0s | ants:  1131 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 53 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5327
N=27  | 20097 -> 5852 |    0s | ants:  1115 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 54 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5852
N=27  | 20097 -> 5610 |    0s | ants:  1112 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 55 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5610
N=27  | 20097 -> 12109 |    1s | ants:  2218 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 pherom
one_power=1
Possibility: 56 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12109
N=27  | 20097 -> 7903 |    1s | ants:  2153 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 pheromo
ne_power=2
Possibility: 57 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 7903
N=27  | 20097 -> 12500 |    1s | ants:  2256 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 pherom
one_power=1
Possibility: 58 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 12500
N=27  | 20097 -> 8020 |    1s | ants:  2229 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 59 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 8020
```

```
N=27  | 20097 -> 5411 |    1s | ants:  2152 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 60 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5411
N=27  | 20097 -> 5404 |    1s | ants:  2181 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 61 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5404
N=27  | 20097 -> 5411 |    1s | ants:  2088 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 62 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5411
N=27  | 20097 -> 5625 |    1s | ants:  2294 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 63 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5625
N=27  | 20097 -> 12638 |    2s | ants:  4276 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 pherom
one_power=1
Possibility: 64 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12638
N=27  | 20097 -> 9142 |    1s | ants:  4546 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 pheromo
ne_power=2
Possibility: 65 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 9142
N=27  | 20097 -> 11544 |    2s | ants:  4449 | trips:   10 | max_round_tri
ps=1000 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 pherom
one_power=1
Possibility: 66 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 11544
N=27  | 20097 -> 8539 |    2s | ants:  4475 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 67 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 8539
N=27  | 20097 -> 5439 |    1s | ants:  4363 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 68 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5439
N=27  | 20097 -> 5401 |    1s | ants:  4524 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 69 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5401
```

```
N=27  | 20097 -> 5417 |    1s | ants:  4354 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 70 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5417
N=27  | 20097 -> 5504 |    1s | ants:  4516 | trips:   10 | max_round_trip
s=1000 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 71 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5504
N=27  | 20097 -> 12421 |    1s | ants:  1421 | trips:   12 | max_round_tri
ps=1000 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 pherom
one_power=1
Possibility: 72 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12421
N=27  | 20097 -> 9989 |    2s | ants:  3617 | trips:   29 | max_round_trip
s=1000 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 pheromo
ne_power=2
Possibility: 73 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 9989
N=27  | 20097 -> 13378 |    1s | ants:  1140 | trips:   10 | max_round_tri
ps=1000 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 pherom
one_power=1
Possibility: 74 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 13378
N=27  | 20097 -> 10877 |    2s | ants:  3840 | trips:   31 | max_round_tri
ps=1000 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 75 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 10877
N=27  | 20097 -> 5393 |    1s | ants:  2444 | trips:   21 | max_round_trip
s=1000 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 76 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5393
N=27  | 20097 -> 6170 |    1s | ants:  1744 | trips:   14 | max_round_trip
s=1000 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 77 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 6170
N=27  | 20097 -> 5349 |    1s | ants:  3944 | trips:   32 | max_round_trip
s=1000 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 78 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5349
N=27  | 20097 -> 6131 |    1s | ants:  1662 | trips:   14 | max_round_trip
s=1000 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 79 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 6131
```

```
N=27  | 20097 -> 7965 |    2s | ants:  4096 | trips:   17 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 pheromo
ne_power=1
Possibility: 80 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 7965
N=27  | 20097 -> 9666 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 pheromo
ne_power=2
Possibility: 81 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 9666
N=27  | 20097 -> 12186 |    1s | ants:  2921 | trips:   12 | max_round_tri
ps=1000 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 pherom
one_power=1
Possibility: 82 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 12186
N=27  | 20097 -> 8529 |    1s | ants:  3587 | trips:   15 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 83 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 8529
N=27  | 20097 -> 5430 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 84 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5430
N=27  | 20097 -> 5430 |    1s | ants:  2233 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 85 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5430
N=27  | 20097 -> 5349 |    1s | ants:  3021 | trips:   13 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 86 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5349
N=27  | 20097 -> 5510 |    1s | ants:  2297 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 87 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5510
N=27  | 20097 -> 12478 |    2s | ants:  4547 | trips:   10 | max_round_tri
ps=1000 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 pherom
one_power=1
Possibility: 88 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12478
N=27  | 20097 -> 7955 |    1s | ants:  4429 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 pheromo
ne_power=2
Possibility: 89 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 7955
```

```
N=27  | 20097 -> 11400 |    2s | ants:  4410 | trips:   10 | max_round_tri
ps=1000 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 pherom
one_power=1
Possibility: 90 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 11400
N=27  | 20097 -> 7902 |    2s | ants:  4412 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 pheromo
ne_power=2
Possibility: 91 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 7902
N=27  | 20097 -> 5471 |    1s | ants:  4435 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pheromo
ne_power=1
Possibility: 92 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5471
N=27  | 20097 -> 5393 |    1s | ants:  4540 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pheromo
ne_power=2
Possibility: 93 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5393
N=27  | 20097 -> 5674 |    1s | ants:  4392 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pheromo
ne_power=1
Possibility: 94 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5674
N=27  | 20097 -> 5417 |    1s | ants:  4535 | trips:   10 | max_round_trip
s=1000 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pheromo
ne_power=2
Possibility: 95 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant
_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5417
N=27  | 20097 -> 13328 |    1s | ants:  1128 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 96 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'an
t_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 13328
N=27  | 20097 -> 9474 |    1s | ants:  1139 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 97 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'an
t_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 9474
N=27  | 20097 -> 11637 |    1s | ants:  1130 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 98 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'an
t_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 11637
N=27  | 20097 -> 9737 |    1s | ants:  1141 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 99 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'an
t_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 9737
```

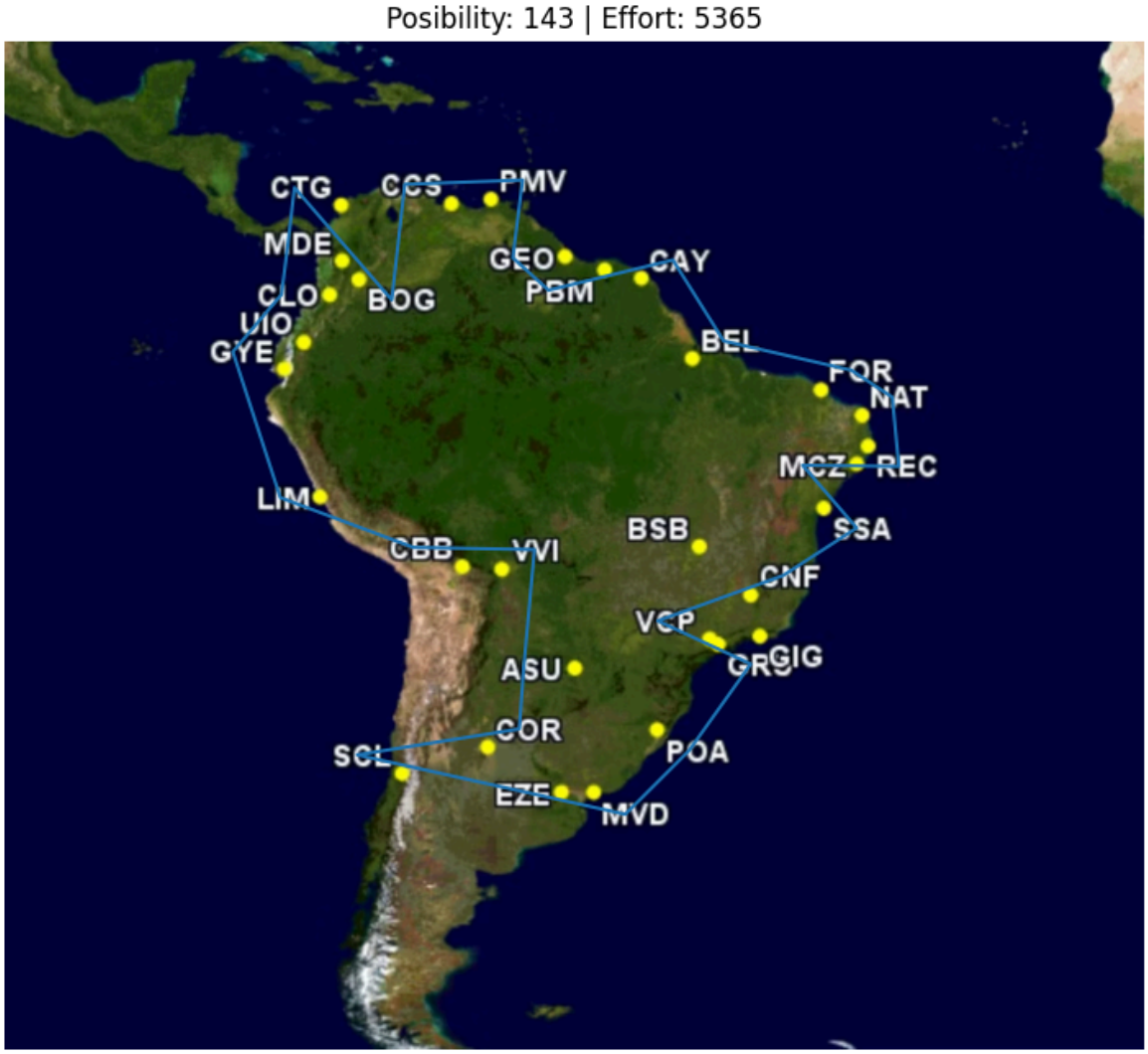
```
N=27  | 20097 -> 5472 |    0s | ants:  1115 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 100 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5472
N=27  | 20097 -> 5550 |    0s | ants:  1137 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 101 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5550
N=27  | 20097 -> 5653 |    0s | ants:  1092 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 102 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5653
N=27  | 20097 -> 5807 |    0s | ants:  1094 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=128 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 103 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5807
N=27  | 20097 -> 12002 |    1s | ants:  2287 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 104 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12002
N=27  | 20097 -> 7712 |    1s | ants:  2272 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 105 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 7712
N=27  | 20097 -> 11751 |    1s | ants:  2254 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 106 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 11751
N=27  | 20097 -> 8204 |    1s | ants:  2260 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 107 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 8204
N=27  | 20097 -> 5357 |    1s | ants:  2198 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 108 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5357
N=27  | 20097 -> 5524 |    1s | ants:  2289 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 109 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5524
```

```
N=27  | 20097 -> 5954 |    1s | ants:  2057 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 110 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5954
N=27  | 20097 -> 5464 |    1s | ants:  2272 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=256 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 111 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5464
N=27  | 20097 -> 12198 |    2s | ants:  4513 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 112 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12198
N=27  | 20097 -> 7679 |    1s | ants:  4427 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 113 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 7679
N=27  | 20097 -> 11660 |    2s | ants:  4430 | trips:   10 | max_round_tri
ps=10000 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 114 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 11660
N=27  | 20097 -> 9145 |    1s | ants:  4560 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 115 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 9145
N=27  | 20097 -> 5327 |    1s | ants:  4208 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 116 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5327
N=27  | 20097 -> 5361 |    1s | ants:  4482 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 117 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5361
N=27  | 20097 -> 5388 |    1s | ants:  4284 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 118 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5388
N=27  | 20097 -> 5432 |    1s | ants:  4525 | trips:   10 | max_round_trip
s=10000 max_ants=1024 ant_count=512 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 119 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5432
```

```
N=27  | 20097 -> 11654 |    2s | ants:  4096 | trips:   33 | max_round_tri
ps=10000 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 120 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 11654
N=27  | 20097 -> 11301 |    1s | ants:  1787 | trips:   15 | max_round_tri
ps=10000 max_ants=4096 ant_count=128 distance_power=0 reward_power=0 phero
mone_power=2
Possibility: 121 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 11301
N=27  | 20097 -> 12500 |    1s | ants:  1901 | trips:   16 | max_round_tri
ps=10000 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 122 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 12500
N=27  | 20097 -> 9884 |    1s | ants:  2342 | trips:   19 | max_round_trip
s=10000 max_ants=4096 ant_count=128 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 123 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 9884
N=27  | 20097 -> 5441 |    1s | ants:  4096 | trips:   35 | max_round_trip
s=10000 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 124 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5441
N=27  | 20097 -> 5731 |    0s | ants:  1266 | trips:   11 | max_round_trip
s=10000 max_ants=4096 ant_count=128 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 125 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5731
N=27  | 20097 -> 5481 |    1s | ants:  1828 | trips:   15 | max_round_trip
s=10000 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 126 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5481
N=27  | 20097 -> 5716 |    0s | ants:  1394 | trips:   11 | max_round_trip
s=10000 max_ants=4096 ant_count=128 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 127 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5716
N=27  | 20097 -> 11422 |    1s | ants:  3104 | trips:   13 | max_round_tri
ps=10000 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 128 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 11422
N=27  | 20097 -> 8104 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 129 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 8104
```

```
N=27  | 20097 -> 10829 |    2s | ants:  4096 | trips:   17 | max_round_tri
ps=10000 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 130 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 10829
N=27  | 20097 -> 7891 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 131 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 7891
N=27  | 20097 -> 5418 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 132 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5418
N=27  | 20097 -> 5525 |    1s | ants:  2268 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 133 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5525
N=27  | 20097 -> 5366 |    1s | ants:  4096 | trips:   17 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 134 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5366
N=27  | 20097 -> 5516 |    1s | ants:  2295 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=256 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 135 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 256, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5516
N=27  | 20097 -> 12465 |    2s | ants:  4557 | trips:   10 | max_round_tri
ps=10000 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 phero
mone_power=1
Possibility: 136 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
1} | Distance: 12465
N=27  | 20097 -> 7867 |    1s | ants:  4504 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=0 reward_power=0 pherom
one_power=2
Possibility: 137 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 0, 'pheromone_power':
2} | Distance: 7867
N=27  | 20097 -> 12102 |    2s | ants:  4454 | trips:   10 | max_round_tri
ps=10000 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 phero
mone_power=1
Possibility: 138 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
1} | Distance: 12102
N=27  | 20097 -> 7278 |    1s | ants:  4519 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=0 reward_power=1 pherom
one_power=2
Possibility: 139 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 0, 'reward_power': 1, 'pheromone_power':
2} | Distance: 7278
```

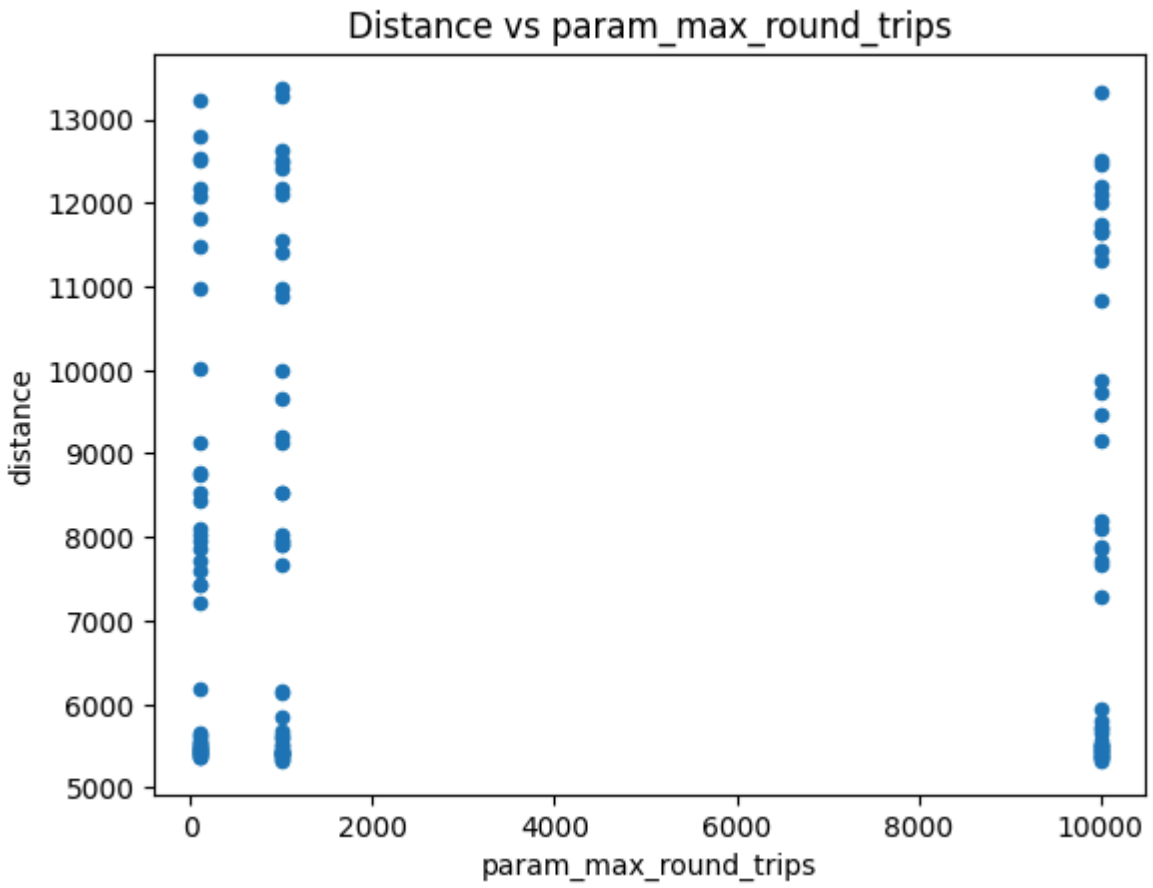
```
N=27  | 20097 -> 5437 |    1s | ants:  4348 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pherom
one_power=1
Possibility: 140 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
1} | Distance: 5437
N=27  | 20097 -> 5401 |    1s | ants:  4579 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=1 reward_power=0 pherom
one_power=2
Possibility: 141 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power':
2} | Distance: 5401
N=27  | 20097 -> 5502 |    1s | ants:  4280 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pherom
one_power=1
Possibility: 142 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
1} | Distance: 5502
N=27  | 20097 -> 5365 |    1s | ants:  4554 | trips:   10 | max_round_trip
s=10000 max_ants=4096 ant_count=512 distance_power=1 reward_power=1 pherom
one_power=2
Possibility: 143 | Params: {'max_round_trips': 10000, 'max_ants': 4096, 'a
nt_count': 512, 'distance_power': 1, 'reward_power': 1, 'pheromone_power':
2} | Distance: 5365
```

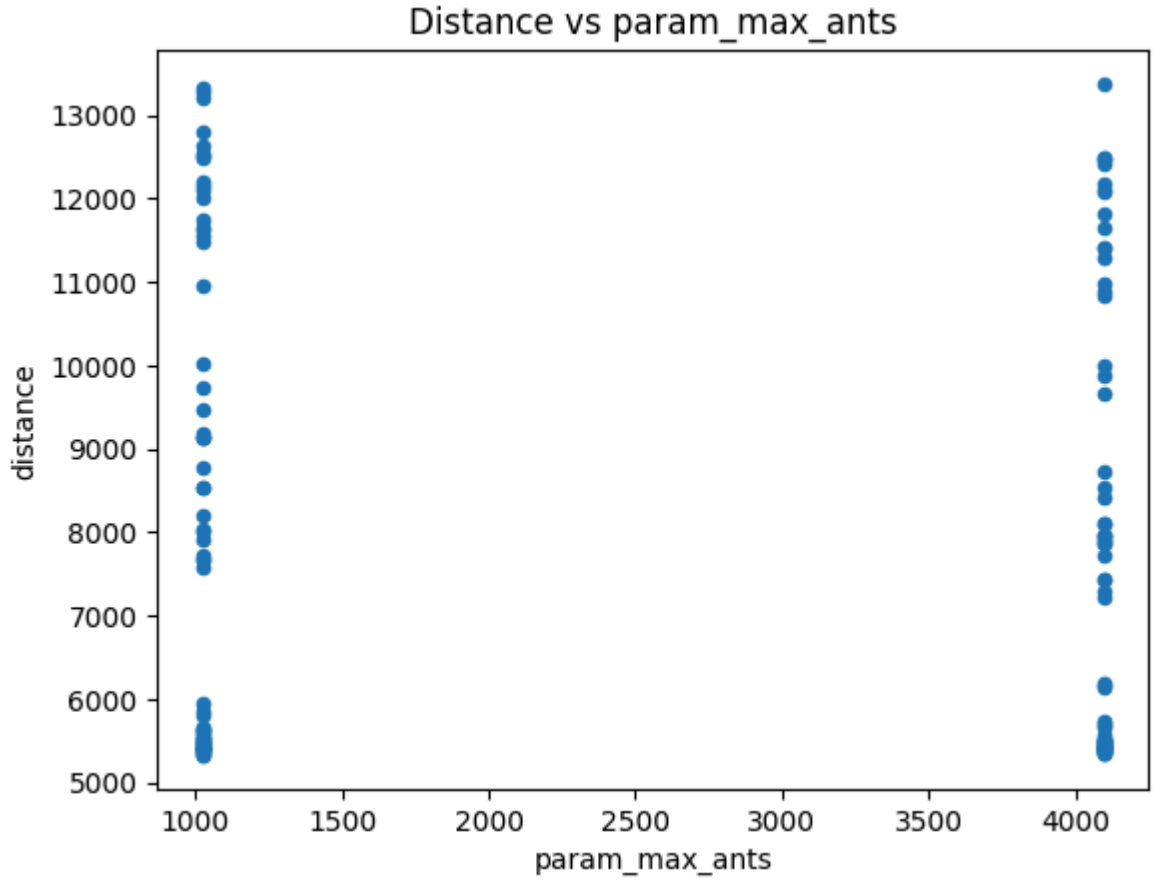


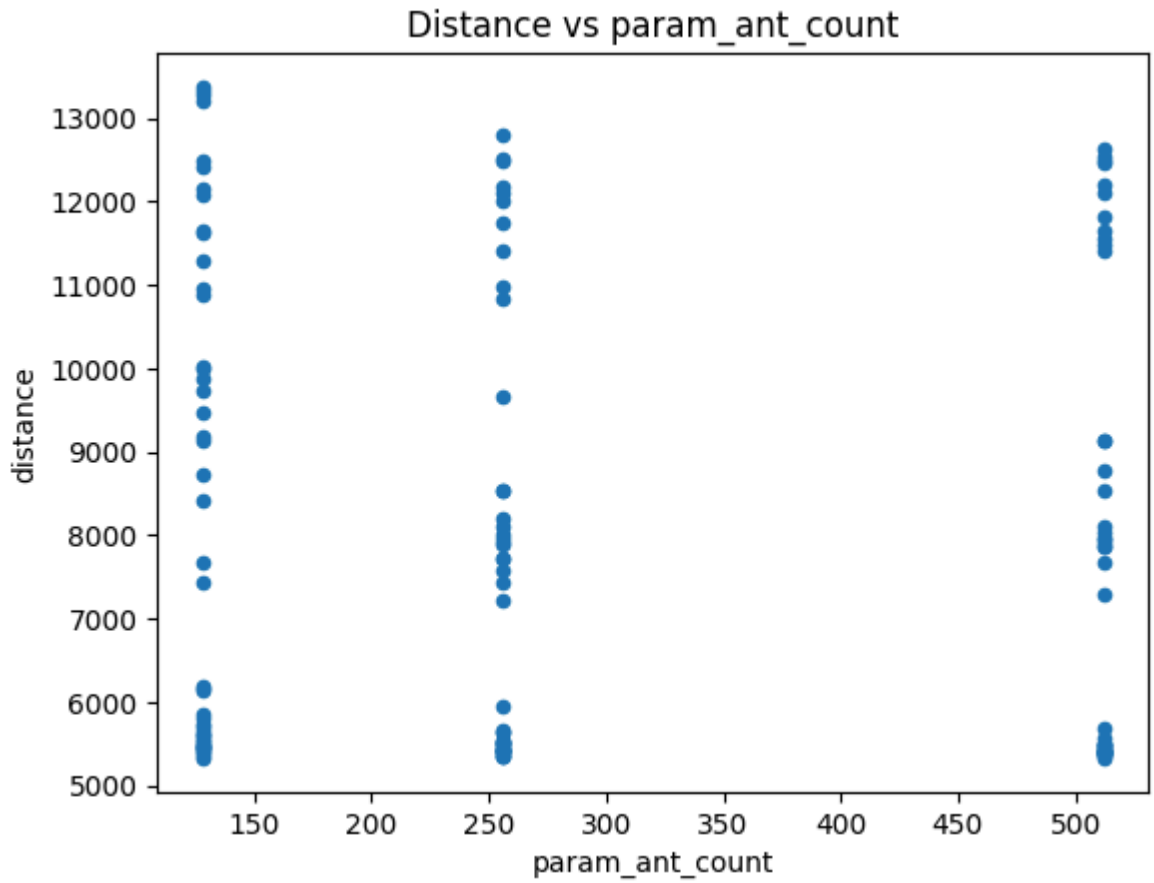
In [13]:
```
tests_results_df = pd.DataFrame(tests_results).T
tests_results_df = tests_results_df.sort_values(by='distance')
tests_results_df.to_csv('results.csv')
```

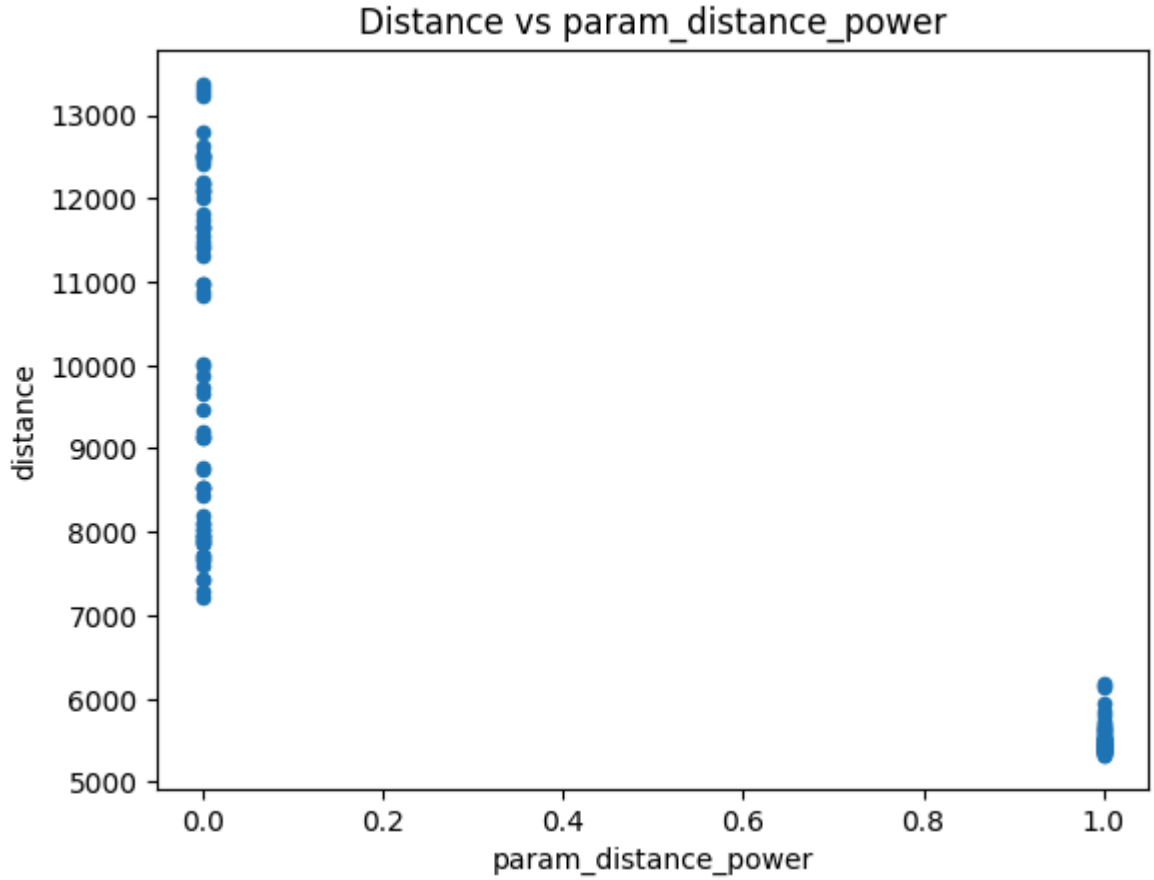
Comparando resultados:

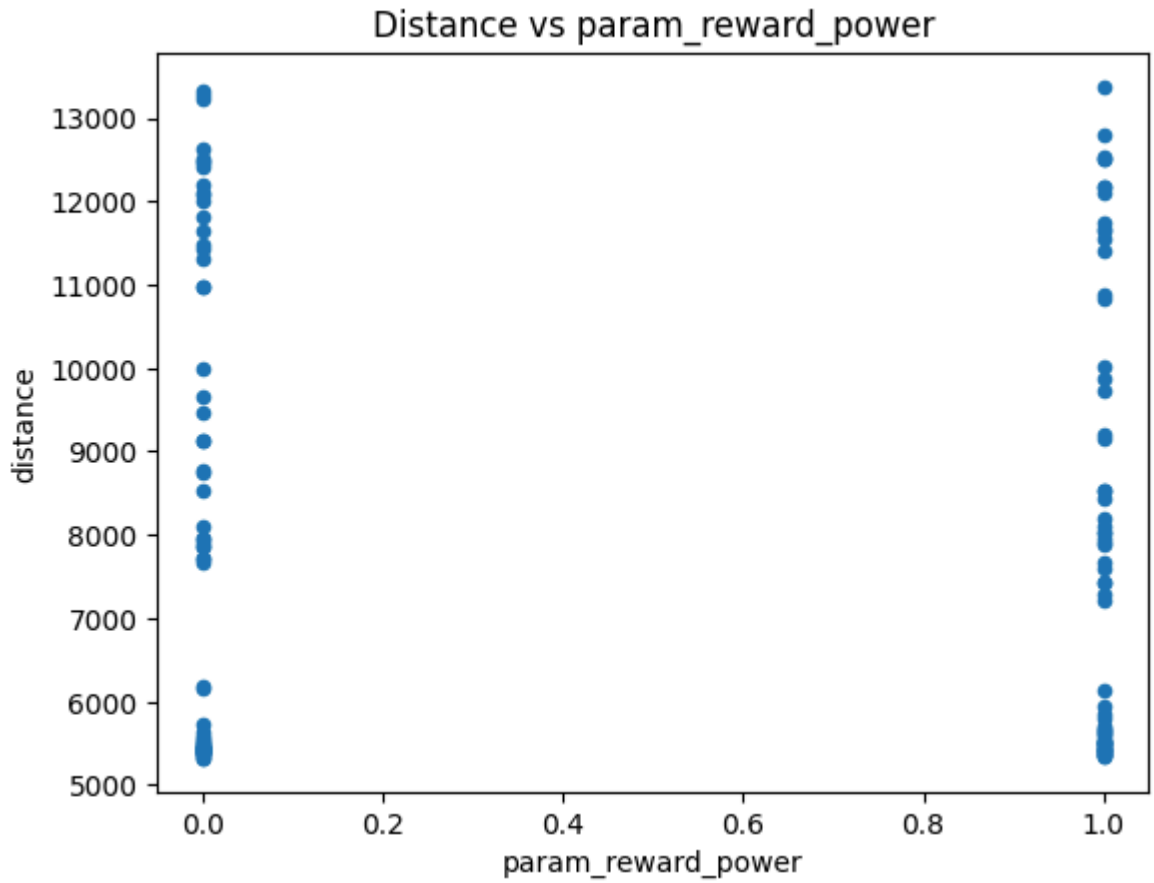
In [18]:
```
tests_results_df = tests_results_df.sort_values(by='distance')

# extra params from params colunm with each item as col of DF with patter

for index, row in tests_results_df.iterrows():
    for param_name, param_value in row['params'].items():
        tests_results_df.loc[index, f'param_{param_name}'] = param_value

tests_results_df.to_csv('results-params-extracted.csv')

# plot a chart with correlation between distance and each param

colunms_of_params = [col for col in tests_results_df.columns if 'param_'

for param_name in colunms_of_params:
    tests_results_df.plot.scatter(x=param_name, y='distance', title=f"Dis
```

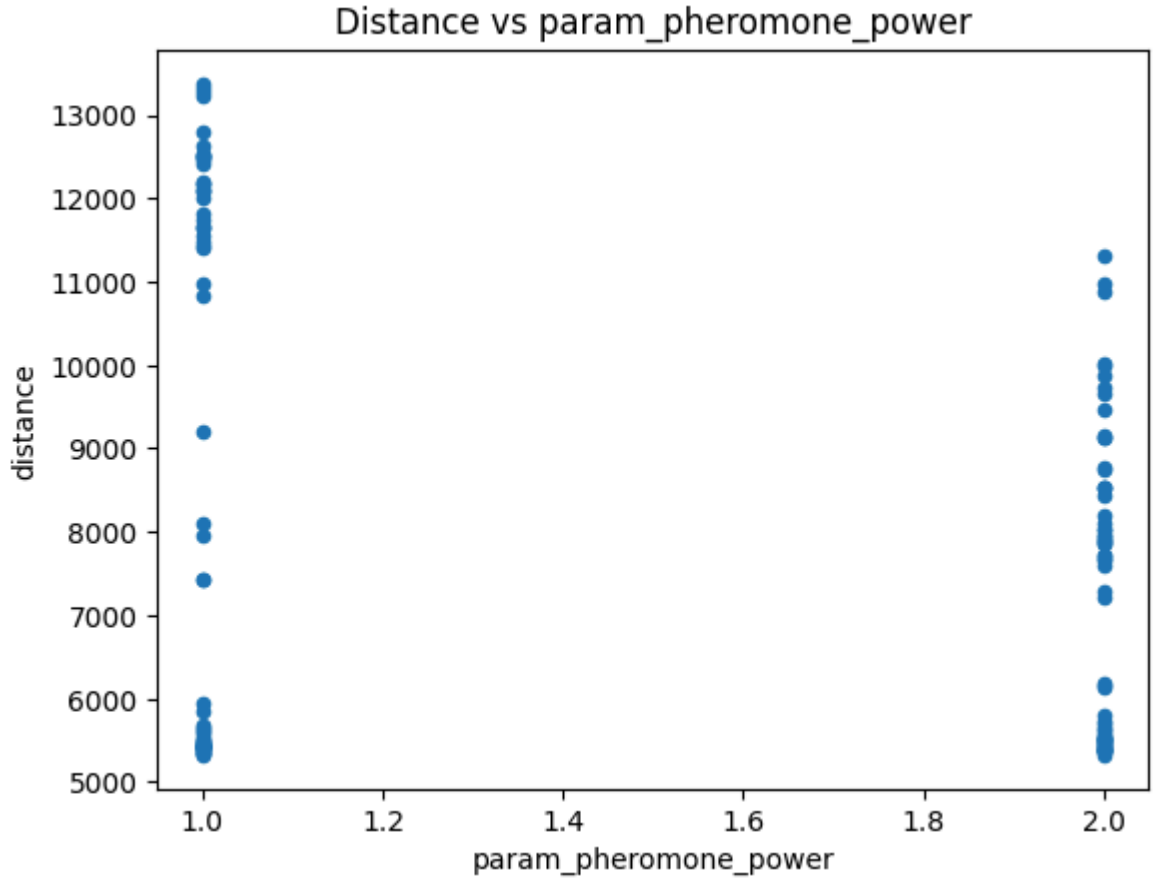



Distance vs param_max_ants
distance
13000
12000
11000
10000
9000
8000
7000
6000
5000
1000
1500
2000
2500
3000
3500
4000
param_max_ants


Distance vs param_ant_count
distance
13000
12000
11000
10000
9000
8000
7000
6000
5000
150
200
250
300
350
400
450
500
param_ant_count

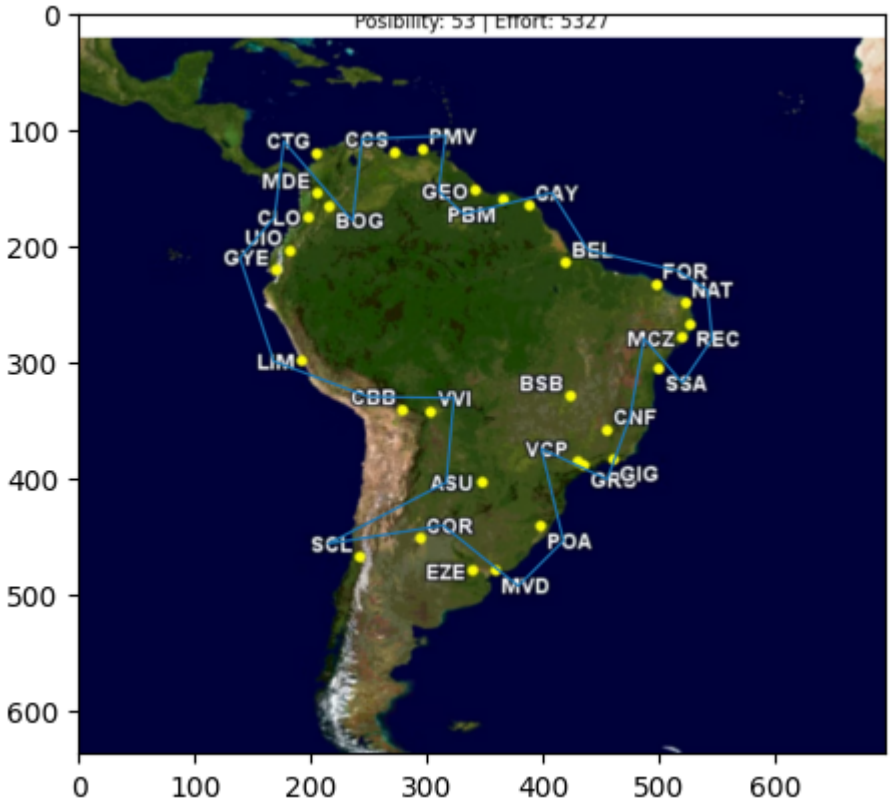

Distance vs param_distance_power
distance
13000
12000
11000
10000
9000
8000
7000
6000
5000
0.0
0.2
0.4
0.6
0.8
1.0
param_distance_power


Distance vs param_reward_power
distance
13000
12000
11000
10000
9000
8000
7000
6000
5000
0.0
0.2
0.4
0.6
0.8
1.0
param_reward_power

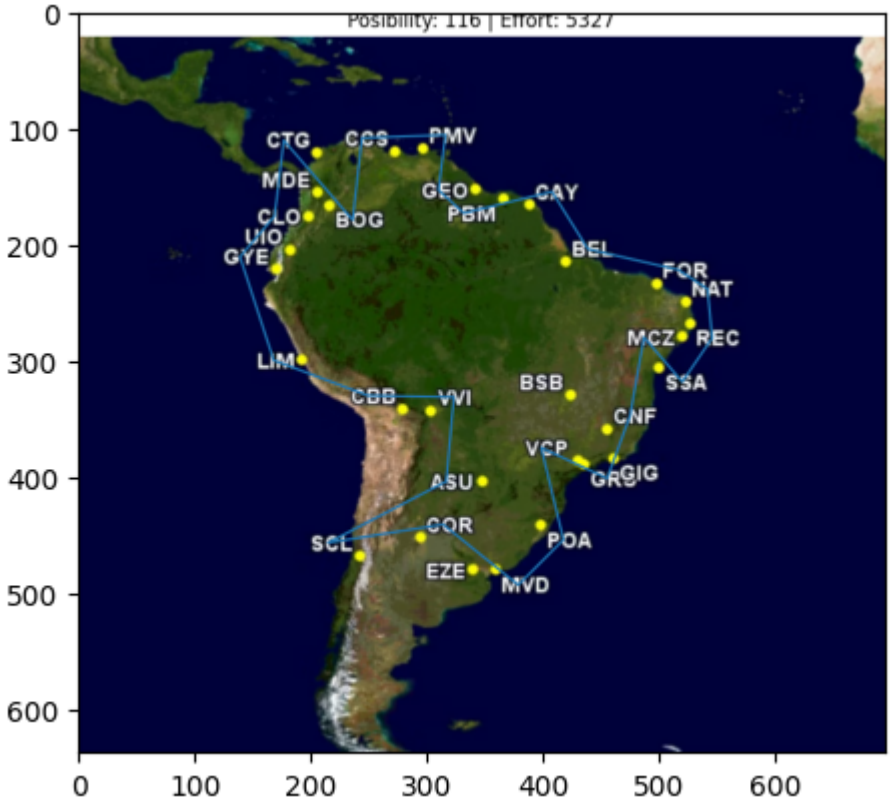


In [15]:
```
best_results = tests_results_df.head(3)

for index, row in best_results.iterrows():
    print(f"Index: {index} | Params: {row['params']} | Distance: {row['di

    image = Image.open(row['image_result'])
    plt.imshow(image)
    plt.show()
```

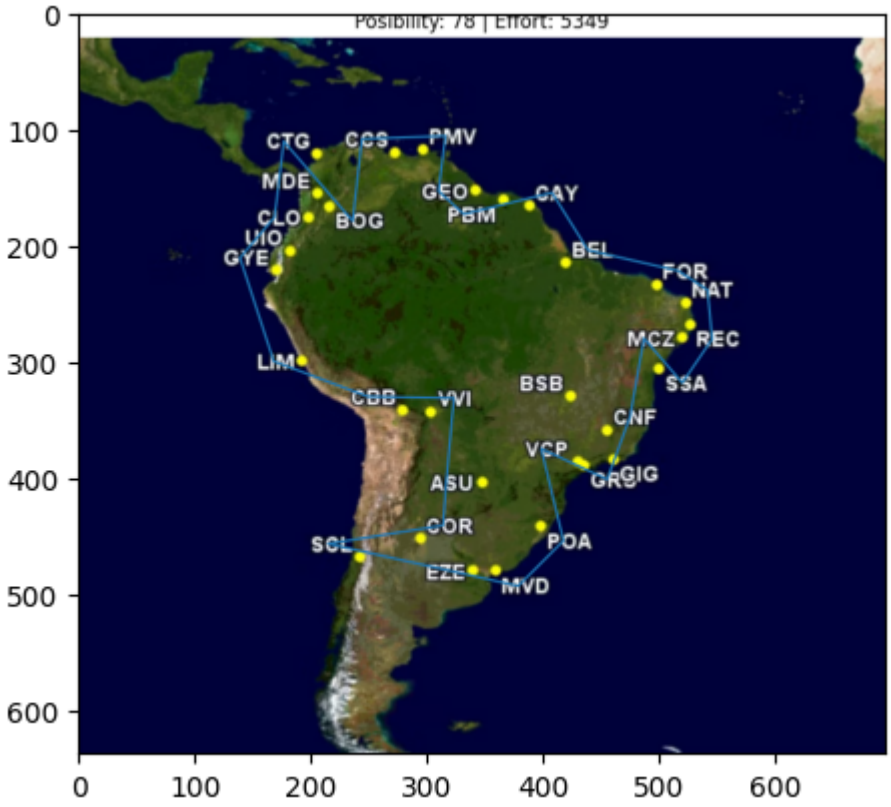
```
Index: 53 | Params: {'max_round_trips': 1000, 'max_ants': 1024, 'ant_coun
t': 128, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 2} | D
istance: 5327
```



Index: 116 | Params: {'max_round_trips': 10000, 'max_ants': 1024, 'ant_count': 512, 'distance_power': 1, 'reward_power': 0, 'pheromone_power': 1} | Distance: 5327



Index: 78 | Params: {'max_round_trips': 1000, 'max_ants': 4096, 'ant_count': 128, 'distance_power': 1, 'reward_power': 1, 'pheromone_power': 1} | Distance: 5349



Resposta: Podemos notar que todos os que tiveram melhor resultado utilizaram distance power = 1, o que indica que a distância é o fator mais importante para a otimização da rota. Além disso o número máximo de formigas não influenciou tanto no resultado, mas o pheromone power influenciou bastante, quanto maior o valor, melhor o resultado.