

Package: `arcgisrouting` (via `r-universe`)

July 7, 2026

Title Access the ArcGIS Routing and Network Analysis Services

Version 1.0.0

Description Bindings to the ArcGIS Routing REST API (<https://developers.arcgis.com/rest/routing/>) for solving network analysis problems. Plan routes and generate driving directions, measure travel time and distance with origin-destination cost matrices, build service areas, find the closest facilities, route fleets of vehicles, and snap GPS tracks to roads. Both synchronous requests and asynchronous geoprocessing jobs are supported, returning simple features ('sf') objects ready for analysis and mapping.

License Apache License (>= 2)

Encoding UTF-8

Language en

RoxygenNote 7.3.3

Imports `arcgisutils`, `cli`, `httr2`, `sf`, `yyjsonr`, `vctrs`, `RcppSimdJson`, `rlang`, `hms`

Depends R (>= 4.3)

Suggests `readr`, `heck`, `brio`, `wk`, `testthat` (>= 3.0.0)

Config/testthat/edition 3

URL <http://r.esri.com/arcgisrouting/>,
<https://github.com/r-arcgis/arcgisrouting>

BugReports <https://github.com/r-arcgis/arcgisrouting/issues>

NeedsCompilation yes

Author Josiah Parry [aut, cre] (ORCID:
<https://orcid.org/0000-0001-9910-865X>)

Maintainer Josiah Parry <josiah.parry@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-07 09:30:21 UTC

RemoteUrl <https://github.com/cran/arcgisrouting>

RemoteRef HEAD

RemoteSha f982eb49809084ecfcde0fb8b2ade807d878a255

Contents

decode_compressed_geometry	2
download_od_results	3
download_service_area_results	4
find_closest_facilities	5
find_closest_facilities_job	9
find_routes	13
find_routes_job	17
find_service_areas	20
find_service_areas_job	23
get_travel_modes	26
last_mile_delivery	27
location_allocation_job	31
od_cost_matrix	35
od_cost_matrix_job	38
route_vehicles	42
route_vehicles_job	46
snap_to_roads	49

Index **52**

decode_compressed_geometry

Decode Compressed Geometry

Description

Decodes ArcGIS compressed geometry strings and converts them to an sfc object.

Usage

```
decode_compressed_geometry(geometry)
```

Arguments

geometry Character vector containing compressed geometry strings.

Value

An sfc object with LIENSTRING geometries in EPSG:4326.

References

[maslke/arcgis-compressed-geometry](#)

download_od_results *Download Origin-Destination Cost Matrix Results*

Description

Downloads and unzips the result file produced by a completed origin-destination cost matrix geo-processing job, reading each CSV output into a named list of data frames with snake_case column names.

Usage

```
download_od_results(job)
```

Arguments

job A completed origin-destination cost matrix job object.

Value

A named list of data frames, one per CSV output, with snake_case names derived from the output file names.

See Also

Other async: [download_service_area_results\(\)](#), [find_closest_facilities_job\(\)](#), [find_routes_job\(\)](#), [find_service_areas_job\(\)](#), [last_mile_delivery\(\)](#), [location_allocation_job\(\)](#), [od_cost_matrix_job\(\)](#), [route_vehicles_job\(\)](#)

Other od: [od_cost_matrix\(\)](#), [od_cost_matrix_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

origins <- st_sfc(
  st_point(c(-122.4194, 37.7749)),
  st_point(c(-122.4313, 37.7793)),
  crs = 4326
)

destinations <- st_sfc(
  st_point(c(-122.4083, 37.7858)),
  st_point(c(-122.4000, 37.7900)),
  crs = 4326
)
```

```
job <- od_cost_matrix_job(origins, destinations)
job$start()
job$await()

download_od_results(job)

## End(Not run)
```

download_service_area_results

Download Service Area Results

Description

Downloads and unzips the result file produced by a completed service area geoprocessing job, parsing each JSON output into a named list of data frames with snake_case column names.

Usage

```
download_service_area_results(job)
```

Arguments

job A completed service area job object.

Value

A named list of data frames, one per JSON output, with snake_case names derived from the output file names.

See Also

Other async: [download_od_results\(\)](#), [find_closest_facilities_job\(\)](#), [find_routes_job\(\)](#), [find_service_areas_job\(\)](#), [last_mile_delivery\(\)](#), [location_allocation_job\(\)](#), [od_cost_matrix_job\(\)](#), [route_vehicles_job\(\)](#)

Other service area: [find_service_areas\(\)](#), [find_service_areas_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

facilities <- st_sfc(
```

```
    st_point(c(-122.4194, 37.7749)),
    st_point(c(-122.0312, 37.3318)),
    crs = 4326
  )

  job <- find_service_areas_job(facilities, break_values = c(5, 10, 15))
  job$start()
  job$await()

  download_service_area_results(job)

  ## End(Not run)
```

find_closest_facilities

Find Closest Facilities

Description

Finds one or more nearby facilities from incidents based on travel time or distance.

Usage

```
find_closest_facilities(
  incidents,
  facilities,
  travel_mode = NULL,
  default_target_facility_count = NULL,
  travel_direction = NULL,
  default_cutoff = NULL,
  time_of_day = NULL,
  time_of_day_usage = NULL,
  u_turns = NULL,
  use_hierarchy = NULL,
  impedance = NULL,
  accumulate_impedance = NULL,
  restrictions = NULL,
  attribute_parameter_values = NULL,
  point_barriers = NULL,
  line_barriers = NULL,
  polygon_barriers = NULL,
  return_geometry = c("facilities", "incidents"),
  directions_language = NULL,
  directions_output_type = NULL,
  directions_style = NULL,
  directions_length_units = NULL,
  directions_time_attribute = NULL,
```

```

output_lines = "true_shape",
ignore_invalid_locations = TRUE,
preserve_object_id = FALSE,
return_empty_results = FALSE,
output_geometry_precision = 10,
output_geometry_precision_units = "meters",
geometry_precision = NULL,
geometry_precision_m = NULL,
locate_settings = NULL,
crs = 4326,
token = arcgisutils::arc_token()
)

```

Arguments

<code>incidents</code>	an <code>sf</code> or <code>sfc</code> object containing point geometries representing the locations to search from.
<code>facilities</code>	an <code>sf</code> or <code>sfc</code> object containing point geometries representing the facilities to search for.
<code>travel_mode</code>	Character. The name of the travel mode to use. See <code>get_travel_modes()</code> for available options. Default: <code>NULL</code> .
<code>default_target_facility_count</code>	default <code>NULL</code> . An integer scalar. The number of closest facilities to find per incident.
<code>travel_direction</code>	default <code>"away"</code> . A scalar character. One of <code>"away"</code> (away from facility) or <code>"towards"</code> (toward facility).
<code>default_cutoff</code>	default <code>NULL</code> . A numeric scalar. The travel time or distance value at which to stop searching for facilities.
<code>time_of_day</code>	default <code>NULL</code> . A scalar date-time. Either a <code>POSIXt</code> scalar or a character string parseable by <code>as.POSIXlt()</code> . The time and date at which travel begins.
<code>time_of_day_usage</code>	default <code>NULL</code> . A scalar character. One of <code>"start_time"</code> or <code>"end_time"</code> . Specifies whether <code>time_of_day</code> represents departure or arrival time.
<code>u_turns</code>	default <code>NULL</code> . A scalar character. U-turn policy at junctions. One of <code>"allow_backtrack"</code> , <code>"deadend_intersection"</code> , <code>"deadend"</code> , <code>"no_backtrack"</code> .
<code>use_hierarchy</code>	Logical. Whether to use hierarchy when finding routes. Default: <code>NULL</code> .
<code>impedance</code>	default <code>NULL</code> . A scalar character. The impedance to minimize. One of <code>"travel_time"</code> , <code>"minutes"</code> , <code>"truck_travel_time"</code> , <code>"truck_minutes"</code> , <code>"walk_time"</code> , <code>"miles"</code> , <code>"kilometers"</code> .
<code>accumulate_impedance</code>	default <code>NULL</code> . A character vector. Additional impedance values to accumulate.
<code>restrictions</code>	Character vector. Restriction names to honor. Default: <code>NULL</code> .
<code>attribute_parameter_values</code>	default <code>NULL</code> . A list of objects. Additional values required by an attribute or restriction.

`point_barriers` default NULL. An `sf` or `sfc` object of point geometries representing barriers to restrict or add cost to travel.

`line_barriers` default NULL. An `sf` or `sfc` object of line geometries representing barriers to restrict or add cost to travel.

`polygon_barriers`
Polygon barriers as `sf` or `sfc` object. Default: NULL.

`return_geometry`
default `c("facilities", "incidents")`. A character vector. Valid values: `"cf_routes"`, `"facilities"`, `"incidents"`, `"directions"`, `"barriers"`, `"polyline_barriers"`, `"polygon_barriers"`, `"traversed_edges"`, `"traversed_junctions"`, `"traversed_turns"`. Use `"everything"` to return all.

`directions_language`
Character. Language code for directions (e.g., `"en"`). Default: `"en"`.

`directions_output_type`
default NULL. A scalar character. One of `"standard"`, `"complete"`, `"complete_no_events"`, `"instructions_only"`, `"summary_only"`, `"feature_sets"`.

`directions_style`
default NULL. A scalar character. One of `"desktop"`, `"navigation"`, `"campus"`.

`directions_length_units`
default NULL. A scalar character. One of `"miles"`, `"kilometers"`, `"feet"`, `"meters"`, `"yards"`, `"nautical_miles"`.

`directions_time_attribute`
default NULL. A scalar character. The time-based impedance attribute used for direction durations.

`output_lines` default NULL (no lines). A scalar character or NULL. One of `"true_shape"` or `"with_measure"`.

`ignore_invalid_locations`
Logical. Whether to ignore invalid locations. Default: TRUE.

`preserve_object_id`
default FALSE. A logical scalar. Preserves object IDs from input locations in the output.

`return_empty_results`
default FALSE. A logical scalar. Returns empty results instead of an error on failure.

`output_geometry_precision`
default 10. A numeric scalar. Simplification tolerance applied to output geometry.

`output_geometry_precision_units`
default `"meters"`. A scalar character. Units for `output_geometry_precision`. Same valid values as `trim_polygon_distance_units`.

`geometry_precision`
default NULL. An integer scalar. Decimal places for x and y values in response geometries.

`geometry_precision_m`
default NULL. A scalar character. Decimal places for m-values in response geometries.

<code>locate_settings</code>	default NULL. A list controlling how inputs are located on the network.
<code>crs</code>	default 4326. The coordinate reference system of the output geometries. Passed to <code>arcgisutils::as_spatial_reference()</code> .
<code>token</code>	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A named list. Elements present depend on `return_geometry`:

- `cf_routes`: route features between incidents and facilities
- `facilities`: facility features
- `incidents`: incident features
- `direction_points`: point features for direction maneuvers
- `direction_lines`: line features for route segments
- `barriers`: point barrier features
- `polyline_barriers`: polyline barrier features
- `polygon_barriers`: polygon barrier features
- `traversed_edges`: traversed edge features
- `traversed_junctions`: traversed junction features
- `traversed_turns`: traversed turn features
- `messages`: status and warning messages from the service

References

[API Reference](#)

See Also

Other direct: [find_routes\(\)](#), [find_service_areas\(\)](#), [od_cost_matrix\(\)](#), [route_vehicles\(\)](#), [snap_to_roads\(\)](#)

Other closest facility: [find_closest_facilities_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

incidents <- st_sfc(st_point(c(-122.4496, 37.7467)), crs = 4326)

facilities <- st_sf(
  name = c("Station 11", "Station 20", "Station 24", "Station 39"),
  geometry = st_sfc(
```

```

        st_point(c(-122.4267, 37.7486)),
        st_point(c(-122.4561, 37.7513)),
        st_point(c(-122.4409, 37.7533)),
        st_point(c(-122.4578, 37.7407)),
        crs = 4326
    )
)

result <- find_closest_facilities(
  incidents = incidents,
  facilities = facilities,
  default_target_facility_count = 2,
  travel_direction = "away",
  default_cutoff = 3,
  return_geometry = "cf_routes",
  directions_length_units = "miles",
  crs = 3857
)

result

## End(Not run)

```

```

find_closest_facilities_job
      Find Closest Facilities (Async)

```

Description

Submits an asynchronous geoprocessing job that finds one or more nearby facilities from incidents based on travel time or travel distance using the ArcGIS /FindClosestFacilities GP service.

Usage

```

find_closest_facilities_job(
  incidents,
  facilities,
  travel_mode = NULL,
  number_of_facilities_to_find = NULL,
  cutoff = NULL,
  travel_direction = NULL,
  measurement_units = NULL,
  analysis_region = NULL,
  time_of_day = NULL,
  time_of_day_usage = NULL,
  uturn_at_junctions = NULL,
  use_hierarchy = NULL,
  restrictions = NULL,

```

```

attribute_parameter_values = NULL,
time_impedance = NULL,
distance_impedance = NULL,
impedance = NULL,
point_barriers = NULL,
line_barriers = NULL,
polygon_barriers = NULL,
route_shape = NULL,
route_line_simplification_tolerance = NULL,
populate_directions = NULL,
directions_language = NULL,
directions_distance_units = NULL,
directions_style_name = NULL,
save_route_data = NULL,
save_output_network_analysis_layer = NULL,
output_format = "feature_set",
ignore_invalid_locations = TRUE,
token = arcgisutils::arc_token()
)

```

Arguments

<code>incidents</code>	An <code>sf</code> or <code>sfc</code> object containing point geometries representing the locations from which the nearby facilities are searched.
<code>facilities</code>	An <code>sf</code> or <code>sfc</code> object containing point geometries representing the locations that are searched for when finding the closest location.
<code>travel_mode</code>	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: <code>NULL</code> .
<code>number_of_facilities_to_find</code>	Integer. The number of closest facilities to find per incident. Default: <code>NULL</code> (API default: 1).
<code>cutoff</code>	Numeric. The travel time or travel distance value at which to stop searching for facilities for a given incident. Units are determined by <code>measurement_units</code> . Default: <code>NULL</code> (API default: no cutoff).
<code>travel_direction</code>	Character. Direction the closest facility search is measured. One of: <code>"facility"</code> (from facilities to incidents) or <code>"incident"</code> (from incidents to facilities). Default: <code>NULL</code> (API default: <code>"incident"</code>).
<code>measurement_units</code>	Character. Units for reporting total travel time or distance. One of: <code>"meters"</code> , <code>"kilometers"</code> , <code>"feet"</code> , <code>"yards"</code> , <code>"miles"</code> , <code>"nautical_miles"</code> , <code>"seconds"</code> , <code>"minutes"</code> , <code>"hours"</code> , <code>"days"</code> . Default: <code>NULL</code> (API default: <code>"minutes"</code>).
<code>analysis_region</code>	Character. Region for the analysis. One of: <code>"europe"</code> , <code>"japan"</code> , <code>"korea"</code> , <code>"middle_east_and_africa"</code> , <code>"north_america"</code> , <code>"south_america"</code> , <code>"south_asia"</code> , <code>"thailand"</code> . Default: <code>NULL</code> (auto-detected).

time_of_day	POSIXct. The departure time for the routes. When NULL, static average speeds are used. Default: NULL.
time_of_day_usage	Character. Whether time_of_day represents the departure or arrival time of the routes. One of: "start" (departure) or "end" (arrival). Default: NULL (API default: "start").
uturn_at_junctions	Character. U-turn policy at junctions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
use_hierarchy	Logical. Whether to use the street hierarchy when finding routes. Default: NULL (API default: TRUE).
restrictions	Character vector. Restriction names to apply. Default: NULL.
attribute_parameter_values	List. Additional values for attributes or restrictions, passed through as a JSON object. Default: NULL.
time_impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
distance_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
impedance	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
route_shape	Character. Shape of the output route features. One of: "true_shape", "true_shape_with_measures", "straight_line", "none". Default: NULL (API default: "true_shape").
route_line_simplification_tolerance	List with elements distance (numeric) and units (character). Simplification tolerance for the output route geometry. Default: NULL (API default: 10 meters).
populate_directions	Logical. Generate driving directions. Default: NULL (API default: FALSE).
directions_language	Character. Language code for directions. Default: NULL (API default: "en").
directions_distance_units	Character. Units for distances in directions. One of: "feet", "kilometers", "meters", "miles", "nautical_miles", "yards". Default: NULL.
directions_style_name	Character. Formatting style for directions. One of: "desktop", "navigation". Default: NULL.
save_route_data	Logical. Whether the route data is saved as a .zip file. Default: NULL (API default: FALSE).

`save_output_network_analysis_layer`
 Logical. Whether to save the analysis as a network analysis layer package file.
 Default: NULL (API default: FALSE).

`output_format` Character. Format for output features. One of: "feature_set", "json_file",
 "geojson_file". Default: "feature_set".

`ignore_invalid_locations`
 Logical. Whether to ignore invalid input locations. Default: TRUE.

`token` Authorization token. Default: `arcgisutils::arc_token()`.

Value

A `find_closest_facilities_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: `download_od_results()`, `download_service_area_results()`, `find_routes_job()`,
`find_service_areas_job()`, `last_mile_delivery()`, `location_allocation_job()`, `od_cost_matrix_job()`,
`route_vehicles_job()`

Other closest facility: `find_closest_facilities()`

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

incidents <- st_sfc(st_point(c(-122.4496, 37.7467)), crs = 4326)

facilities <- st_sf(
  name = c("Station 11", "Station 20", "Station 24", "Station 39"),
  geometry = st_sfc(
    st_point(c(-122.4267, 37.7486)),
    st_point(c(-122.4561, 37.7513)),
    st_point(c(-122.4409, 37.7533)),
    st_point(c(-122.4578, 37.7407)),
    crs = 4326
  )
)

job <- find_closest_facilities_job(
  incidents,
  facilities,
```

```

    number_of_facilities_to_find = 2,
    travel_direction = "facility",
    cutoff = 5,
    measurement_units = "minutes",
    populate_directions = TRUE
  )
  job$start()
  result <- job$results

  ## End(Not run)

```

 find_routes

Find Routes

Description

Finds the best routes between multiple stops using the ArcGIS routing service.

Usage

```

find_routes(
  stops,
  travel_mode = NULL,
  start_time = NULL,
  find_best_sequence = FALSE,
  preserve_first_stop = TRUE,
  preserve_last_stop = TRUE,
  restrict_u_turns = NULL,
  use_hierarchy = NULL,
  impedance_attribute_name = NULL,
  accumulate_attribute_names = NULL,
  restrictions = NULL,
  barriers = NULL,
  polyline_barriers = NULL,
  polygon_barriers = NULL,
  directions_language = "en",
  directions_type = "standard",
  return_geometry = c("routes", "directions"),
  ignore_invalid_locations = TRUE,
  token = arcgisutils::arc_token()
)

```

Arguments

stops	An sf or sfc object containing point geometries representing the stops to visit.
travel_mode	Character. The name of the travel mode to use. See get_travel_modes() for available options. Default: NULL.

start_time	POSIXct or character. The time at which travel begins. Can be "now" for current time, a POSIXct datetime, or NULL for static speeds. Default: NULL.
find_best_sequence	Logical. Whether to reorder stops to find the optimized route. Default: FALSE.
preserve_first_stop	Logical. Whether to keep the first stop fixed when reordering. Only applies if find_best_sequence = TRUE. Default: TRUE.
preserve_last_stop	Logical. Whether to keep the last stop fixed when reordering. Only applies if find_best_sequence = TRUE. Default: TRUE.
restrict_u_turns	Character. Specifies U-turn restrictions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
use_hierarchy	Logical. Whether to use hierarchy when finding routes. Default: NULL.
impedance_attribute_name	Character. The impedance to use. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
accumulate_attribute_names	Character vector. Additional impedance values to accumulate. Default: NULL.
restrictions	Character vector. Restriction names to honor. Default: NULL.
barriers	Point barriers as sf or sfc object. Default: NULL.
polyline_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
directions_language	Character. Language code for directions (e.g., "en"). Default: "en".
directions_type	Character. Specifies the content and verbosity of driving directions (directionsOutputType in REST API). One of: "complete", "complete_no_events", "instructions_only", "standard", "summary_only", "feature_sets". Default: "standard".
return_geometry	Character vector. Specifies which features to return in the output. Valid values: "routes", "directions", "stops", "barriers", "polyline_barriers", "polygon_barriers", "traversed_edges", "traversed_junctions", "traversed_turns". Default: c("routes", "directions").
ignore_invalid_locations	Logical. Whether to ignore invalid locations. Default: TRUE.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A list containing the routing resps. The elements returned depend on the return_geometry parameter. Possible elements include:

- routes: Route features

- directions: Driving directions. Each element contains a compress_geometry column with per-maneuver segment geometry in ArcGIS compressed format. Use [decode_compressed_geometry\(\)](#) to decode these into sf geometries.
- stops: Stop features
- barriers: Barrier features
- polyline_barriers: Polyline barrier features
- polygon_barriers: Polygon barrier features
- traversed_edges: Traversed edge features
- traversed_junctions: Traversed junction features
- traversed_turns: Traversed turn features
- messages: Status and warning messages from the service

When directions_type = "feature_sets", the response includes:

- direction_points: sf object with point features for direction maneuvers
- direction_lines: sf object with line features for route segments

References

[API Reference](#)

See Also

Other direct: [find_closest_facilities\(\)](#), [find_service_areas\(\)](#), [od_cost_matrix\(\)](#), [route_vehicles\(\)](#), [snap_to_roads\(\)](#)

Other routing: [find_routes_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

# Simple route between 3 stops
stops <- st_sf(
  name = c("Start", "Middle", "End"),
  geometry = st_sfc(
    st_point(c(-122.4194, 37.7749)), # San Francisco
    st_point(c(-122.0312, 37.3318)), # Cupertino
    st_point(c(-121.8863, 37.3382)), # San Jose
    crs = 4326
  )
)

resp <- find_routes(stops)
```

```

# Route with time windows for deliveries
delivery_stops <- st_sf(
  name = c("Warehouse", "Customer A", "Customer B", "Customer C"),
  time_window_start = as.POSIXct(c(
    "2024-01-15 08:00:00",
    "2024-01-15 09:00:00",
    "2024-01-15 11:00:00",
    "2024-01-15 14:00:00"
  )),
  time_window_end = as.POSIXct(c(
    "2024-01-15 08:30:00",
    "2024-01-15 10:00:00",
    "2024-01-15 13:00:00",
    "2024-01-15 16:00:00"
  )),
  geometry = st_sfc(
    st_point(c(-122.4194, 37.7749)),
    st_point(c(-122.4083, 37.7858)),
    st_point(c(-122.4313, 37.7793)),
    st_point(c(-122.4000, 37.7900)),
    crs = 4326
  )
)

delivery_route <- find_routes(
  delivery_stops,
  start_time = as.POSIXct("2024-01-15 08:00:00"),
  travel_mode = "Driving Time"
)

# Find optimal order for stops
sales_stops <- st_sf(
  name = c("Office", "Client 1", "Client 2", "Client 3", "Office"),
  geometry = st_sfc(
    st_point(c(-122.4194, 37.7749)),
    st_point(c(-122.4083, 37.7858)),
    st_point(c(-122.4313, 37.7793)),
    st_point(c(-122.4000, 37.7900)),
    st_point(c(-122.4194, 37.7749)),
    crs = 4326
  )
)

optimized_route <- find_routes(
  sales_stops,
  find_best_sequence = TRUE,
  preserve_first_stop = TRUE,
  preserve_last_stop = TRUE
)

# Route with barriers
barrier_pts <- st_sfc(

```

```
    st_point(c(-122.4150, 37.7800)),
    crs = 4326
  )

route_with_barriers <- find_routes(
  stops,
  barriers = barrier_pts
)

# Accumulate multiple impedances
multi_impedance_route <- find_routes(
  stops,
  impedance_attribute_name = "travel_time",
  accumulate_attribute_names = c("miles", "kilometers")
)

## End(Not run)
```

find_routes_job	<i>Find Routes (Async)</i>
-----------------	----------------------------

Description

Submits an asynchronous geoprocessing job to find the best routes between stops using the ArcGIS /FindRoutes GP service.

Usage

```
find_routes_job(
  stops,
  travel_mode = NULL,
  measurement_units = NULL,
  analysis_region = NULL,
  reorder_stops = FALSE,
  preserve_terminal_stops = NULL,
  return_to_start = NULL,
  use_time_windows = NULL,
  time_of_day = NULL,
  uturn_at_junctions = NULL,
  use_hierarchy = NULL,
  restrictions = NULL,
  impedance = NULL,
  time_impedance = NULL,
  distance_impedance = NULL,
  route_shape = NULL,
  populate_route_edges = NULL,
  populate_directions = NULL,
```

```

directions_language = NULL,
directions_distance_units = NULL,
directions_style_name = NULL,
output_format = "feature_set",
ignore_invalid_locations = TRUE,
point_barriers = NULL,
line_barriers = NULL,
polygon_barriers = NULL,
token = arcgisutils::arc_token()
)

```

Arguments

stops	An sf or sfc object containing point geometries. Recognized attribute columns such as route_name, time_window_start, and time_window_end are used when present.
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.
measurement_units	Character. Units for reporting total travel time or distance. One of: "meters", "kilometers", "feet", "yards", "miles", "nautical_miles", "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
analysis_region	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).
reorder_stops	Logical. Reorder stops to find the optimal route (TSP). Default: FALSE.
preserve_terminal_stops	Character. Which terminal stops to preserve when reorder_stops = TRUE. One of: "first", "last", "first_and_last", "none". Default: NULL (API default: "first").
return_to_start	Logical. Whether the route should return to its starting location. Default: NULL (API default: TRUE).
use_time_windows	Logical. Whether to honour time windows on stops. Default: NULL (auto-detected from stops attributes).
time_of_day	POSIXct. The departure time for the routes. When NULL, static average speeds are used. Default: NULL.
uturn_at_junctions	Character. U-turn policy at junctions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
use_hierarchy	Logical. Whether to use the street hierarchy when finding routes. Default: NULL (API default: TRUE).
restrictions	Character vector. Restriction names to apply. Default: NULL.
impedance	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.

time_impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
distance_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
route_shape	Character. Shape of the output route features. One of: "true_shape", "true_shape_with_measures", "straight_line", "none". Default: NULL (API default: "true_shape").
populate_route_edges	Logical. Generate edges for each route. Default: NULL (API default: FALSE).
populate_directions	Logical. Generate driving directions. Default: NULL (API default: FALSE).
directions_language	Character. Language code for directions. Default: NULL (API default: "en").
directions_distance_units	Character. Units for distances in directions. One of: "feet", "kilometers", "meters", "miles", "nautical_miles", "yards". Default: NULL.
directions_style_name	Character. Formatting style for directions. One of: "desktop", "navigation". Default: NULL.
output_format	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
ignore_invalid_locations	Logical. Whether to ignore invalid input locations. Default: TRUE.
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A `find_routes_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: `download_od_results()`, `download_service_area_results()`, `find_closest_facilities_job()`, `find_service_areas_job()`, `last_mile_delivery()`, `location_allocation_job()`, `od_cost_matrix_job()`, `route_vehicles_job()`

Other routing: `find_routes()`

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

stops <- st_sf(
  name = c("Stop 1", "Stop 2", "Stop 3"),
  geometry = st_sfc(
    st_point(c(145.066, -37.865)),
    st_point(c(145.105, -37.819)),
    st_point(c(145.120, -37.800)),
    crs = 4326
  )
)

job <- find_routes_job(stops)
job$start()
result <- job$results

## End(Not run)
```

find_service_areas	<i>Solve Service Area</i>
--------------------	---------------------------

Description

Solve Service Area

Usage

```
find_service_areas(
  facilities,
  default_breaks = c(5, 10, 15),
  travel_mode = NULL,
  travel_direction = "away",
  time_of_day = NULL,
  output_polygons = "simplified",
  split_polygons_at_breaks = TRUE,
  overlap_polygons = TRUE,
  merge_similar_polygon_ranges = FALSE,
  trim_outer_polygons = TRUE,
  output_lines = NULL,
  split_lines_at_breaks = TRUE,
  overlap_lines = TRUE,
```

```

return_geometry = "service_areas",
u_turns = NULL,
use_hierarchy = NULL,
impedance = NULL,
accumulate_impedance = NULL,
restrictions = NULL,
trim_polygon_distance = 100,
trim_polygon_distance_units = "meters",
point_barriers = NULL,
line_barriers = NULL,
polygon_barriers = NULL,
ignore_invalid_locations = TRUE,
output_geometry_precision = 10,
output_geometry_precision_units = "meters",
crs = 4326,
token = arcgisutils::arc_token()
)

```

Arguments

facilities an sf or sfc object containing point geometries representing the facilities around which service areas are generated.

default_breaks default c(5, 10, 15). A numeric vector specifying the size and number of service areas to generate for each facility. Units are determined by the impedance parameter.

travel_mode Character. The name of the travel mode to use. See [get_travel_modes\(\)](#) for available options. Default: NULL.

travel_direction default "away". A scalar character. One of "away" (away from facility) or "towards" (toward facility).

time_of_day default NULL. A scalar date-time. Either a POSIXt scalar or a character string parseable by `as.POSIXlt()`. The time and date at which travel begins.

output_polygons default "simplified". A scalar character or NULL (no polygons). One of "simplified" or "detailed".

split_polygons_at_breaks default TRUE. A logical scalar. When TRUE, service areas appear as rings between breaks. When FALSE, each area is a disk from the facility to the break.

overlap_polygons default TRUE. A logical scalar. Whether service area polygons from different facilities can overlap.

merge_similar_polygon_ranges default FALSE. A logical scalar. Whether service area polygons from different facilities with the same break value are merged into a single polygon.

trim_outer_polygons default TRUE. A logical scalar. Whether service areas are trimmed to lie within a distance of the network. Ignored when `use_hierarchy = TRUE`.

output_lines	default NULL (no lines). A scalar character or NULL. One of "true_shape" or "with_measure".
split_lines_at_breaks	default TRUE. A logical scalar. Whether service area lines are split at break values.
overlap_lines	default TRUE. A logical scalar. Whether service area lines from different facilities can overlap.
return_geometry	default "service_areas". A character vector. Valid values: "service_areas", "sa_lines", "facilities", "barriers", "polyline_barriers", "polygon_barriers".
u_turns	default NULL. A scalar character. U-turn policy at junctions. One of "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack".
use_hierarchy	Logical. Whether to use hierarchy when finding routes. Default: NULL.
impedance	default NULL. A scalar character. The impedance to minimize. One of "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers".
accumulate_impedance	default NULL. A character vector. Additional impedance values to accumulate.
restrictions	Character vector. Restriction names to honor. Default: NULL.
trim_polygon_distance	default 100. An integer scalar. The distance within which the service area polygon extends from the network.
trim_polygon_distance_units	default "meters". A scalar character. Units for trim_polygon_distance. One of "meters", "kilometers", "feet", "miles", "nautical_miles", "yards".
point_barriers	default NULL. An sf or sfc object of point geometries representing barriers to restrict or add cost to travel.
line_barriers	default NULL. An sf or sfc object of line geometries representing barriers to restrict or add cost to travel.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
ignore_invalid_locations	Logical. Whether to ignore invalid locations. Default: TRUE.
output_geometry_precision	default 10. A numeric scalar. Simplification tolerance applied to output geometry.
output_geometry_precision_units	default "meters". A scalar character. Units for output_geometry_precision. Same valid values as trim_polygon_distance_units.
crs	default 4326. The coordinate reference system of the output geometries. Passed to <code>arcgisutils::as_spatial_reference()</code> .
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A list containing the service area results.

References

[API Reference](#)

See Also

Other direct: [find_closest_facilities\(\)](#), [find_routes\(\)](#), [od_cost_matrix\(\)](#), [route_vehicles\(\)](#), [snap_to_roads\(\)](#)

Other service area: [download_service_area_results\(\)](#), [find_service_areas_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

facility <- st_sfc(st_point(c(-122.253, 37.757)), crs = 4326)
find_service_areas(facility)

## End(Not run)
```

find_service_areas_job

Generate Service Areas (Async)

Description

Submits an asynchronous geoprocessing job to generate service areas around facilities using the ArcGIS /GenerateServiceAreas GP service.

Usage

```
find_service_areas_job(
  facilities,
  break_values = NULL,
  break_units = NULL,
  travel_mode = NULL,
  travel_direction = NULL,
  time_of_day = NULL,
  use_hierarchy = NULL,
  uturn_at_junctions = NULL,
  polygons_for_multiple_facilities = NULL,
  polygon_overlap_type = NULL,
  detailed_polygons = NULL,
```

```

polygon_trim_distance = NULL,
polygon_simplification_tolerance = NULL,
polygon_detail = NULL,
output_type = NULL,
analysis_region = NULL,
restrictions = NULL,
impedance = NULL,
time_impedance = NULL,
distance_impedance = NULL,
ignore_invalid_locations = TRUE,
output_format = "feature_set",
point_barriers = NULL,
line_barriers = NULL,
polygon_barriers = NULL,
token = arcgisutils::arc_token()
)

```

Arguments

facilities	An sf or sfc object containing point geometries representing the facilities around which service areas are generated.
break_values	Numeric vector. Service area sizes. Units are determined by break_units. Default: NULL (API default: 5 10 15).
break_units	Character. Units for break_values. One of: "meters", "kilometers", "feet", "yards", "miles", "nautical_miles", "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.
travel_direction	Character. Direction of travel relative to facilities. One of: "away", "towards". Default: NULL (API default: "away").
time_of_day	POSIXct. The departure time for the routes. When NULL, static average speeds are used. Default: NULL.
use_hierarchy	Logical. Whether to use the street hierarchy when finding routes. Default: NULL (API default: TRUE).
uturn_at_junctions	Character. U-turn policy at junctions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
polygons_for_multiple_facilities	Character. How service area polygons from multiple facilities are generated. One of: "overlapping", "not_overlapping", "merge". Default: NULL (API default: "overlapping").
polygon_overlap_type	Character. Whether polygons are rings or disks. One of: "rings", "disks". Default: NULL (API default: "rings").
detailed_polygons	Logical. Generate detailed polygons. Default: NULL (API default: FALSE).

polygons	List with elements distance (numeric) and units (character). Trim polygons to within this distance of the network. Default: NULL.
polygons_trim_distance	List with elements distance (numeric) and units (character). Simplification tolerance for output polygons. Default: NULL.
polygons_simplification_tolerance	Character. Level of detail for output polygons. One of: "standard", "generalized", "high". Default: NULL (API default: "standard").
polygons_detail	Character. Type of output to generate. One of: "polygons", "lines", "polygons_and_lines". Default: NULL (API default: "polygons").
output_type	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).
analysis_region	Character vector. Restriction names to apply. Default: NULL.
restrictions	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
time_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
distance_impedance	Logical. Whether to ignore invalid input locations. Default: TRUE.
ignore_invalid_locations	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
output_format	Point barriers as sf or sfc object. Default: NULL.
point_barriers	Line barriers as sf or sfc object. Default: NULL.
line_barriers	Polygon barriers as sf or sfc object. Default: NULL.
polygon_barriers	Authorization token. Default: <code>arcgisutils::arc_token()</code> .
token	

Value

A `find_service_areas_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: `download_od_results()`, `download_service_area_results()`, `find_closest_facilities_job()`, `find_routes_job()`, `last_mile_delivery()`, `location_allocation_job()`, `od_cost_matrix_job()`, `route_vehicles_job()`

Other service area: `download_service_area_results()`, `find_service_areas()`

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

facilities <- st_sfc(
  st_point(c(-122.4194, 37.7749)),
  st_point(c(-122.0312, 37.3318)),
  crs = 4326
)

job <- find_service_areas_job(facilities, break_values = c(5, 10, 15))
job$start()
result <- job$results

## End(Not run)
```

get_travel_modes *Get available travel modes*

Description

Returns the names of the travel modes supported by the routing services associated with the provided token.

Usage

```
get_travel_modes(token = arc_token())
```

Arguments

token Authorization token. Default: `arcgisutils::arc_token()`.

Value

A character vector of supported travel mode names.

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(arcgisutils)
set_arc_token(auth_user())
```

```
get_travel_modes()  
  
## End(Not run)
```

last_mile_delivery *Solve Last Mile Delivery (Async)*

Description

Submits an asynchronous geoprocessing job to solve a last-mile delivery problem using the ArcGIS /SolveLastMileDelivery GP service. Last Mile Delivery is a specialized vehicle routing problem that creates geographically clustered delivery routes to minimize fleet operating costs.

Usage

```
last_mile_delivery(  
  orders,  
  depots,  
  routes = NULL,  
  order_specialties = NULL,  
  route_specialties = NULL,  
  zones = NULL,  
  travel_mode = NULL,  
  analysis_region = NULL,  
  ignore_network_location_fields = FALSE,  
  ignore_invalid_order_locations = FALSE,  
  earliest_route_start_date = NULL,  
  earliest_route_start_time = NULL,  
  time_zone_usage_for_time_fields = NULL,  
  max_route_total_time = NULL,  
  sequence_gap = NULL,  
  time_units = NULL,  
  distance_units = NULL,  
  route_shape = NULL,  
  populate_directions = FALSE,  
  directions_language = NULL,  
  save_route_data = FALSE,  
  save_output_network_analysis_layer = FALSE,  
  output_format = "feature_set",  
  point_barriers = NULL,  
  line_barriers = NULL,  
  polygon_barriers = NULL,  
  token = arcgisutils::arc_token()  
)
```

Arguments

orders	An sf object containing point geometries representing delivery and pickup locations.
depots	An sf or sfc object containing point geometries representing depot locations.
routes	A data.frame describing vehicle and driver characteristics. Default: NULL.
order_specialties	A data.frame mapping orders to the specialties they require, with columns order_name and specialty_name. Default: NULL.
route_specialties	A data.frame mapping routes to the specialties they support, with columns route_name and specialty_name. Default: NULL.
zones	An sf or sfc polygon object delineating work territories, each carrying a name attribute. Default: NULL.
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.
analysis_region	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).
ignore_network_location_fields	Logical. Whether network location fields on the inputs are ignored. Default: FALSE.
ignore_invalid_order_locations	Logical. Whether to ignore invalid orders instead of failing. Default: FALSE.
earliest_route_start_date	Date. Default earliest start date applied to routes whose earliest_start_date attribute is missing. Default: NULL.
earliest_route_start_time	An hms or character "hh:mm:ss" value giving the default earliest start time applied to routes whose earliest_start_time attribute is missing. Default: NULL.
time_zone_usage_for_time_fields	Character. Time zone used for all date-time input fields. One of: "geo_local", "utc". Default: NULL (API default: "geo_local").
max_route_total_time	Numeric. Default maximum allowed total time for each route, applied to routes whose max_total_time attribute is missing. Interpreted in time_units. Default: NULL.
sequence_gap	Integer. Gap in numerical values to leave in the Sequence attribute between adjacent orders when the analysis is solved. Default: NULL (API default: 1).
time_units	Character. Units for all time-based attribute values. One of: "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
distance_units	Character. Units for all distance-based attribute values. One of: "miles", "kilometers", "meters", "feet", "yards", "nautical_miles". Default: NULL (API default: "miles").

route_shape	Character. Geometry type for output route lines. One of: "true_shape_with_measures", "straight_line", "none". Default: NULL (API default: "straight_line").
populate_directions	Logical. Whether to generate driving directions for each route. Default: FALSE.
directions_language	Character. Language code for directions (e.g. "en"). Default: NULL.
save_route_data	Logical. Whether to save results as a .zip file geodatabase. Default: FALSE.
save_output_network_analysis_layer	Logical. Whether the analysis settings are saved as a network analysis layer package file. Default: FALSE.
output_format	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A `last_mile_delivery_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: `download_od_results()`, `download_service_area_results()`, `find_closest_facilities_job()`, `find_routes_job()`, `find_service_areas_job()`, `location_allocation_job()`, `od_cost_matrix_job()`, `route_vehicles_job()`

Other vrp: `route_vehicles()`, `route_vehicles_job()`

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

orders <- st_as_sf(
  data.frame(
    name = c("Order 1", "Order 2"),
    service_time = c(5, 5),
```

```

time_window_start = as.POSIXct(
  c(NA, 1706860800),
  origin = "1970-01-01",
  tz = "UTC"
),
time_window_end = as.POSIXct(
  c(1706868000, 1706868000),
  origin = "1970-01-01",
  tz = "UTC"
),
max_violation_time = c(0, 30),
delivery_quantity_1 = c(2000, 1500),
delivery_quantity_2 = c(100, 75),
x = c(-117, -117.5),
y = c(34, 34.5)
),
coords = c("x", "y"),
crs = 4326
)

depots <- st_as_sf(
  data.frame(name = "Depot 1", x = -117.2, y = 34.2),
  coords = c("x", "y"),
  crs = 4326
)

routes <- data.frame(
  name = c("Truck 1", "Truck 2"),
  start_depot_name = c("Depot 1", "Depot 1"),
  end_depot_name = c("Depot 1", "Depot 1"),
  earliest_start_time = c("6:00:00", "6:00:00"),
  capacity_1 = c(40000, 30000),
  capacity_2 = c(2000, 2500),
  cost_per_unit_time = c(0.5, 0.5),
  cost_per_unit_distance = c(1.5, 1.5)
)

order_specialties <- data.frame(
  order_name = c("Order 1", "Order 2"),
  specialty_name = c("Refrigerated", "Hazmat")
)

route_specialties <- data.frame(
  route_name = c("Truck 1", "Truck 2"),
  specialty_name = c("Refrigerated", "Hazmat")
)

zone1 <- st_polygon(list(rbind(
  c(-97.0634, 32.8442),
  c(-97.0554, 32.84),
  c(-97.0558, 32.8327),
  c(-97.0638, 32.83),
  c(-97.0634, 32.8442)

```

```

)))

zone2 <- st_multipolygon(list(
  list(rbind(
    c(-97.0803, 32.8235),
    c(-97.0776, 32.8277),
    c(-97.074, 32.8254),
    c(-97.0767, 32.8227),
    c(-97.0803, 32.8235)
  )),
  list(rbind(
    c(-97.0871, 32.8311),
    c(-97.0831, 32.8292),
    c(-97.0853, 32.8259),
    c(-97.0892, 32.8279),
    c(-97.0871, 32.8311)
  ))
))

zones <- st_cast(
  st_sf(
    name = c("Zone 1", "Zone 2"),
    geometry = st_sfc(zone1, zone2, crs = 4326)
  ),
  "MULTIPOLYGON"
)

job <- last_mile_delivery(
  orders = orders,
  depots = depots,
  routes = routes,
  order_specialties = order_specialties,
  route_specialties = route_specialties,
  zones = zones,
  earliest_route_start_date = as.Date("2024-02-02"),
  max_route_total_time = 480,
  sequence_gap = 3,
  time_units = "minutes",
  route_shape = "true_shape_with_measures",
  populate_directions = TRUE
)

job$start()
job$results

## End(Not run)

```

Description

Submits an asynchronous geoprocessing job that chooses the set of facilities which best serve demand from surrounding areas, simultaneously locating facilities and allocating demand points to them, using the ArcGIS /SolveLocationAllocation GP service.

Usage

```
location_allocation_job(
    facilities,
    demand_points,
    travel_mode = NULL,
    locate_settings = NULL,
    measurement_units = NULL,
    analysis_region = NULL,
    problem_type = NULL,
    number_of_facilities_to_find = NULL,
    default_measurement_cutoff = NULL,
    default_capacity = NULL,
    target_market_share = NULL,
    measurement_transformation_model = NULL,
    measurement_transformation_factor = NULL,
    travel_direction = NULL,
    time_of_day = NULL,
    uturn_at_junctions = NULL,
    use_hierarchy = NULL,
    restrictions = NULL,
    attribute_parameter_values = NULL,
    allocation_line_shape = NULL,
    time_impedance = NULL,
    distance_impedance = NULL,
    impedance = NULL,
    point_barriers = NULL,
    line_barriers = NULL,
    polygon_barriers = NULL,
    save_output_network_analysis_layer = NULL,
    output_format = "feature_set",
    ignore_invalid_locations = TRUE,
    token = arcgisutils::arc_token()
)
```

Arguments

facilities	An sf or sfc object containing point geometries representing the locations that serve as facilities.
demand_points	An sf or sfc object containing point geometries representing the demand points.
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.

locate_settings	List. Settings that affect how input data are located, passed through as a JSON object. Default: NULL.
measurement_units	Character. Units for reporting total travel time or distance. One of: "meters", "kilometers", "feet", "yards", "miles", "nautical_miles", "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
analysis_region	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).
problem_type	Character. Objective of the location-allocation analysis. One of: "maximize_attendance", "maximize_capacitated_coverage", "maximize_coverage", "maximize_market_share", "minimize_facilities", "minimize_impedance", "target_market_share". Default: NULL (API default: "minimize_impedance").
number_of_facilities_to_find	Integer. The number of facilities the task should choose. Default: NULL (API default: 1).
default_measurement_cutoff	Numeric. The maximum travel time or distance allowed between a demand point and the facility to which it is allocated. Units are determined by measurement_units. Default: NULL (API default: no cutoff).
default_capacity	Numeric. The default capacity assigned to all facilities. Only applicable to the "maximize_capacitated_coverage" problem type. Default: NULL (API default: 1).
target_market_share	Numeric. The percentage of total demand weight to capture. Only applicable to the "target_market_share" problem type. Default: NULL (API default: 10).
measurement_transformation_model	Character. Equation for transforming the network cost between facilities and demand points. One of: "linear", "power", "exponential". Default: NULL (API default: "linear").
measurement_transformation_factor	Numeric. The impedance parameter value (λ) for measurement_transformation_model. Ignored when the model is linear. Default: NULL (API default: 1).
travel_direction	Character. Direction in which to measure travel. One of: "facility_to_demand" or "demand_to_facility". Default: NULL (API default: "facility_to_demand").
time_of_day	POSIXct. The departure time for the routes. When NULL, static average speeds are used. Default: NULL.
uturn_at_junctions	Character. U-turn policy at junctions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
use_hierarchy	Logical. Whether to use the street hierarchy when finding routes. Default: NULL (API default: TRUE).

restrictions	Character vector. Restriction names to apply. Default: NULL.
attribute_parameter_values	List. Additional values for attributes or restrictions, passed through as a JSON object. Default: NULL.
allocation_line_shape	Character. Type of line features output by the request. One of: "straight_line" or "none". Default: NULL (API default: "straight_line").
time_impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
distance_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
impedance	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
save_output_network_analysis_layer	Logical. Whether to save the analysis as a network analysis layer package file. Default: NULL (API default: FALSE).
output_format	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
ignore_invalid_locations	Logical. Whether to ignore invalid input locations. Default: TRUE.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A `location_allocation_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: `download_od_results()`, `download_service_area_results()`, `find_closest_facilities_job()`, `find_routes_job()`, `find_service_areas_job()`, `last_mile_delivery()`, `od_cost_matrix_job()`, `route_vehicles_job()`

Examples

```

## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

facilities <- st_sf(
  name = c("Facility A", "Facility B"),
  facility_type = c(0L, 0L),
  curb_approach = c(0L, 0L),
  geometry = st_sfc(
    st_point(c(-58.557329417999938, -34.587693706999971)),
    st_point(c(-58.460247408999976, -34.683348039999942)),
    crs = 4326
  )
)

demand_points <- st_sf(
  name = c("Household 4", "Household 3", "Household 2", "Household 1"),
  group_name = c("A", "A", NA, NA),
  weight = c(2, 2, 3, 5),
  curb_approach = c(0L, 0L, 0L, 1L),
  geometry = st_sfc(
    st_point(c(-58.664405163999959, -34.614819562999969)),
    st_point(c(-58.514499119999982, -34.496322404999944)),
    st_point(c(-58.541624975999998, -34.788996107999935)),
    st_point(c(-58.405995697999997, -34.637662387999967)),
    crs = 4326
  )
)

job <- location_allocation_job(facilities, demand_points)
job$start()
result <- job$results

## End(Not run)

```

od_cost_matrix

Origin-Destination Cost Matrix

Description

Creates an origin-destination cost matrix containing the travel cost between every origin and destination.

Usage

```

od_cost_matrix(
  origins,
  destinations = origins,
  travel_mode = NULL,
  default_cutoff = NULL,
  default_target_destination_count = NULL,
  output_type = NULL,
  time_of_day = NULL,
  u_turns = NULL,
  use_hierarchy = NULL,
  impedance = NULL,
  accumulate_impedance = NULL,
  restrictions = NULL,
  attribute_parameter_values = NULL,
  point_barriers = NULL,
  line_barriers = NULL,
  polygon_barriers = NULL,
  return_geometry = character(0),
  ignore_invalid_locations = TRUE,
  return_empty_results = FALSE,
  geometry_precision = NULL,
  locate_settings = NULL,
  crs = 4326,
  token = arcgisutils::arc_token()
)

```

Arguments

<code>origins</code>	an <code>sf</code> or <code>sfc</code> object containing point geometries representing the starting points.
<code>destinations</code>	default <code>origins</code> . An <code>sf</code> or <code>sfc</code> object containing point geometries representing the ending points.
<code>travel_mode</code>	Character. The name of the travel mode to use. See <code>get_travel_modes()</code> for available options. Default: <code>NULL</code> .
<code>default_cutoff</code>	default <code>NULL</code> . A numeric scalar. The travel time or distance value at which to stop searching for facilities.
<code>default_target_destination_count</code>	default <code>NULL</code> . An integer scalar. The maximum number of destinations to find per origin.
<code>output_type</code>	default <code>NULL</code> . A scalar character. One of <code>"no_lines"</code> , <code>"straight_lines"</code> , <code>"sparse_matrix"</code> . Controls whether route geometry is returned.
<code>time_of_day</code>	default <code>NULL</code> . A scalar date-time. Either a <code>POSIXt</code> scalar or a character string parseable by <code>as.POSIXlt()</code> . The time and date at which travel begins.
<code>u_turns</code>	default <code>NULL</code> . A scalar character. U-turn policy at junctions. One of <code>"allow_backtrack"</code> , <code>"deadend_intersection"</code> , <code>"deadend"</code> , <code>"no_backtrack"</code> .
<code>use_hierarchy</code>	Logical. Whether to use hierarchy when finding routes. Default: <code>NULL</code> .

impedance	default NULL. A scalar character. The impedance to minimize. One of "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers".
accumulate_impedance	default NULL. A character vector. Additional impedance values to accumulate.
restrictions	Character vector. Restriction names to honor. Default: NULL.
attribute_parameter_values	default NULL. A list of objects. Additional values required by an attribute or restriction.
point_barriers	default NULL. An sf or sfc object of point geometries representing barriers to restrict or add cost to travel.
line_barriers	default NULL. An sf or sfc object of line geometries representing barriers to restrict or add cost to travel.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
return_geometry	default character(0). A character vector. Valid values: "origins", "destinations", "barriers", "polyline_barriers", "polygon_barriers".
ignore_invalid_locations	Logical. Whether to ignore invalid locations. Default: TRUE.
return_empty_results	default FALSE. A logical scalar. Returns empty results instead of an error on failure.
geometry_precision	default NULL. An integer scalar. Decimal places for x and y values in response geometries.
locate_settings	default NULL. A list controlling how inputs are located on the network.
crs	default 4326. The coordinate reference system of the output geometries. Passed to <code>arcgisutils::as_spatial_reference()</code> .
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A named list. Elements present depend on `return_geometry` and `output_type`:

- `od_cost_matrix`: nested cost matrix returned when `output_type = "sparse_matrix"`. A list with `costAttributeName`s and, per origin ID, a list mapping destination ID to its cost values.
- `od_lines`: OD cost matrix features
- `origins`: origin features
- `destinations`: destination features
- `barriers`: point barrier features
- `polyline_barriers`: polyline barrier features
- `polygon_barriers`: polygon barrier features
- `messages`: status and warning messages from the service

References

[API Reference](#)

See Also

Other direct: [find_closest_facilities\(\)](#), [find_routes\(\)](#), [find_service_areas\(\)](#), [route_vehicles\(\)](#), [snap_to_roads\(\)](#)

Other od: [download_od_results\(\)](#), [od_cost_matrix_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

origins <- st_sfc(
  st_point(c(-122.4194, 37.7749)),
  st_point(c(-122.4313, 37.7793)),
  crs = 4326
)

destinations <- st_sfc(
  st_point(c(-122.4083, 37.7858)),
  st_point(c(-122.4000, 37.7900)),
  st_point(c(-122.4561, 37.7513)),
  crs = 4326
)

result <- od_cost_matrix(
  origins = origins,
  destinations = destinations
)

result

## End(Not run)
```

od_cost_matrix_job *Generate Origin-Destination Cost Matrix (Async)*

Description

Submits an asynchronous geoprocessing job that creates an origin-destination (OD) cost matrix from multiple origins to multiple destinations using the ArcGIS /GenerateOriginDestinationCostMatrix GP service. The matrix reports the travel time or travel distance from every origin to every destination.

Usage

```

od_cost_matrix_job(
  origins,
  destinations = NULL,
  travel_mode = NULL,
  time_units = NULL,
  distance_units = NULL,
  analysis_region = NULL,
  n_dests = NULL,
  cutoff = NULL,
  time_of_day = NULL,
  uturn_at_junctions = NULL,
  use_hierarchy = NULL,
  restrictions = NULL,
  attribute_parameter_values = NULL,
  origin_destination_line_shape = NULL,
  impedance = NULL,
  time_impedance = NULL,
  distance_impedance = NULL,
  point_barriers = NULL,
  line_barriers = NULL,
  polygon_barriers = NULL,
  save_output_network_analysis_layer = NULL,
  output_format = "feature_set",
  ignore_invalid_locations = TRUE,
  token = arcgisutils::arc_token()
)

```

Arguments

origins	An sf or sfc object containing point geometries that function as starting points in generating the paths to destinations.
destinations	An sf or sfc object containing point geometries that function as ending points in generating the paths from origins. Default: NULL.
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.
time_units	Character. Units used to measure and report the total travel time between each origin-destination pair. One of: "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
distance_units	Character. Units used to measure and report the total travel distance between each origin-destination pair. One of: "meters", "kilometers", "feet", "yards", "miles", "nautical_miles". Default: NULL (API default: "miles").
analysis_region	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).

n_dests	Integer. The maximum number of destinations to find per origin. Default: NULL (API default: every destination).
cutoff	Numeric. The travel time or travel distance value at which to stop searching for destinations from a given origin. Default: NULL.
time_of_day	POSIXct. The departure time for the routes. When NULL, static average speeds are used. Default: NULL.
uturn_at_junctions	Character. U-turn policy at junctions. One of: "allow_backtrack", "deadend_intersection", "deadend", "no_backtrack". Default: NULL.
use_hierarchy	Logical. Whether to use the street hierarchy when finding routes. Default: NULL (API default: TRUE).
restrictions	Character vector. Restriction names to apply. Default: NULL.
attribute_parameter_values	List. Additional values for attributes or restrictions, passed through as a JSON object. Default: NULL.
origin_destination_line_shape	Character. Shape of the line feature connecting each origin-destination pair in the output matrix. One of: "straight_line" or "none". Default: NULL (API default: "none").
impedance	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
time_impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
distance_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
save_output_network_analysis_layer	Logical. Whether to save the analysis as a network analysis layer package file. Default: NULL (API default: FALSE).
output_format	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
ignore_invalid_locations	Logical. Whether to ignore invalid input locations. Default: TRUE.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

An `od_cost_matrix_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: [download_od_results\(\)](#), [download_service_area_results\(\)](#), [find_closest_facilities_job\(\)](#), [find_routes_job\(\)](#), [find_service_areas_job\(\)](#), [last_mile_delivery\(\)](#), [location_allocation_job\(\)](#), [route_vehicles_job\(\)](#)

Other od: [download_od_results\(\)](#), [od_cost_matrix\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

origins <- st_sf(
  name = c("Origin 1", "Origin 2"),
  n_dests = c(2L, 3L),
  cutoff = c(120, 90),
  curb_approach = c(0L, 0L),
  geometry = st_sfc(
    st_point(c(-0.1891, 51.5254)),
    st_point(c(-0.1744, 51.5353)),
    crs = 4326
  )
)

destinations <- st_sf(
  name = c("Destination 1", "Destination 2"),
  curb_approach = c(0L, 0L),
  geometry = st_sfc(
    st_point(c(-0.1991, 51.5354)),
    st_point(c(-0.1844, 51.5458)),
    crs = 4326
  )
)

job <- od_cost_matrix_job(origins, destinations)
job$start()
result <- job$results

## End(Not run)
```

route_vehicles	<i>Route Vehicles</i>
----------------	-----------------------

Description

Solves a vehicle routing problem (VRP) to find the most effective routes for a fleet of vehicles visiting a set of orders.

Usage

```
route_vehicles(  
  orders,  
  depots,  
  routes = NULL,  
  travel_mode = NULL,  
  locate_settings = NULL,  
  analysis_region = NULL,  
  time_zone_usage_for_time_fields = NULL,  
  default_date = NULL,  
  breaks = NULL,  
  time_units = NULL,  
  distance_units = NULL,  
  time_window_factor = NULL,  
  spatially_cluster_routes = TRUE,  
  route_zones = NULL,  
  route_renewals = NULL,  
  order_pairs = NULL,  
  excess_transit_factor = NULL,  
  uturn_policy = NULL,  
  point_barriers = NULL,  
  line_barriers = NULL,  
  polygon_barriers = NULL,  
  use_hierarchy_in_analysis = TRUE,  
  restrictions = NULL,  
  attribute_parameter_values = NULL,  
  time_impedance = NULL,  
  distance_impedance = NULL,  
  impedance = NULL,  
  populate_route_lines = TRUE,  
  route_line_simplification_tolerance = NULL,  
  populate_directions = FALSE,  
  directions_language = NULL,  
  directions_style_name = NULL,  
  save_route_data = FALSE,  
  save_output_layer = FALSE,  
  populate_stop_shapes = FALSE,  
  ignore_invalid_order_locations = FALSE,
```

```

    token = arcgisutils::arc_token()
  )

```

Arguments

orders	an sf or sfc object containing point geometries representing locations to visit.
depots	an sf or sfc object containing point geometries representing depot locations.
routes	default NULL. A data.frame describing vehicle and driver characteristics.
travel_mode	Character. The name of the travel mode to use. See <code>get_travel_modes()</code> for available options. Default: NULL.
locate_settings	default NULL. A list controlling how inputs are located on the network.
analysis_region	default NULL. A scalar character. One of "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Speeds up analysis when specified.
time_zone_usage_for_time_fields	default NULL. A scalar character. One of "geo_local" or "utc". Time zone used for all date-time input fields.
default_date	default NULL. A POSIXt or character scalar. The date on which all routes start. Only the date portion is used.
breaks	default NULL. A data.frame of rest period definitions for routes.
time_units	default NULL. A scalar character. One of "seconds", "minutes", "hours", "days". Units for all time-based attribute values.
distance_units	default NULL. A scalar character. One of "miles", "kilometers", "meters", "feet", "yards", "nautical_miles". Units for all distance-based attribute values.
time_window_factor	default NULL. A scalar character. One of "low", "medium", "high". Importance of honoring time windows.
spatially_cluster_routes	default TRUE. A logical scalar. Whether orders assigned to a route are spatially clustered.
route_zones	default NULL. An sf or sfc polygon object delineating work territories for routes.
route_renewals	default NULL. A data.frame of intermediate depot renewal locations for routes.
order_pairs	default NULL. A data.frame pairing pickup and delivery orders.
excess_transit_factor	default NULL. A scalar character. One of "low", "medium", "high". Importance of reducing excess transit time for order pairs.
uturn_policy	default NULL. A scalar character. One of "allow_uturn", "allow_dead_ends_and_intersections_on", "allow_dead_ends_only", "no_uturns".
point_barriers	default NULL. An sf or sfc object of point geometries representing barriers to restrict or add cost to travel.

`line_barriers` default NULL. An `sf` or `sfc` object of line geometries representing barriers to restrict or add cost to travel.

`polygon_barriers`
Polygon barriers as `sf` or `sfc` object. Default: NULL.

`use_hierarchy_in_analysis`
default TRUE. A logical scalar. Whether to use the street network hierarchy when finding routes.

`restrictions` Character vector. Restriction names to honor. Default: NULL.

`attribute_parameter_values`
default NULL. A list of objects providing additional values required by an attribute or restriction.

`time_impedance` default NULL. A scalar character. Time-based impedance. One of "travel_time", "minutes", "walk_time", "truck_minutes", "truck_travel_time".

`distance_impedance`
default NULL. A scalar character. Distance-based impedance. One of "miles", "kilometers".

`impedance` default NULL. A scalar character. The impedance type. One of "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers".

`populate_route_lines`
default TRUE. A logical scalar. Whether output routes include the exact street shape.

`route_line_simplification_tolerance`
default NULL. A numeric scalar. Simplification tolerance for route geometry.

`populate_directions`
default FALSE. A logical scalar. Whether to generate driving directions for each route.

`directions_language`
default NULL. A scalar character. Language code for directions (e.g. "en").

`directions_style_name`
default NULL. A scalar character. One of "desktop", "navigation", "campus".

`save_route_data`
default FALSE. A logical scalar. Whether to save results as a .zip file geodatabase.

`save_output_layer`
default FALSE. A logical scalar. Whether to save the analysis as a network analysis layer package.

`populate_stop_shapes`
default FALSE. A logical scalar. Whether output stops include point geometries.

`ignore_invalid_order_locations`
default FALSE. A logical scalar. Whether to ignore invalid orders instead of failing.

`token` Authorization token. Default: `arcgisutils::arc_token()`.

Value

A named list:

- unassigned_stops: orders that could not be assigned to any route
- stops: assigned stop features with route and sequence information
- routes: route features with geometry and cost attributes
- directions: driving directions (only when populate_directions = TRUE)
- solve_succeeded: logical indicating whether the solve completed
- usage_cost: list with numObjects and credits used
- messages: status and warning messages from the service

References

[API Reference](#)

See Also

Other direct: [find_closest_facilities\(\)](#), [find_routes\(\)](#), [find_service_areas\(\)](#), [od_cost_matrix\(\)](#), [snap_to_roads\(\)](#)

Other vrp: [last_mile_delivery\(\)](#), [route_vehicles_job\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

orders <- st_sf(
  name = c("Order 1", "Order 2"),
  geometry = st_sfc(
    st_point(c(-0.1891, 51.5254)),
    st_point(c(-0.1744, 51.5353)),
    crs = 4326
  )
)

depots <- st_sf(
  name = "Depot1",
  geometry = st_sfc(st_point(c(-0.2, 51.5)), crs = 4326)
)

routes <- data.frame(
  name = "Truck1",
  start_depot_name = "Depot1",
  end_depot_name = "Depot1",
  capacities = "40000"
```

```
)  
  
result <- route_vehicles(orders, depots, routes)  
  
## End(Not run)
```

route_vehicles_job	<i>Solve Vehicle Routing Problem (Async)</i>
--------------------	--

Description

Submits an asynchronous geoprocessing job to solve a vehicle routing problem using the ArcGIS /SolveVehicleRoutingProblem GP service.

Usage

```
route_vehicles_job(  
  orders,  
  depots,  
  routes = NULL,  
  breaks = NULL,  
  travel_mode = NULL,  
  analysis_region = NULL,  
  time_zone_usage_for_time_fields = NULL,  
  default_date = NULL,  
  time_units = NULL,  
  distance_units = NULL,  
  time_window_factor = NULL,  
  spatially_cluster_routes = TRUE,  
  route_zones = NULL,  
  route_renewals = NULL,  
  order_pairs = NULL,  
  excess_transit_factor = NULL,  
  uturn_policy = NULL,  
  restrictions = NULL,  
  impedance = NULL,  
  time_impedance = NULL,  
  distance_impedance = NULL,  
  use_hierarchy_in_analysis = TRUE,  
  populate_route_lines = TRUE,  
  route_line_simplification_tolerance = NULL,  
  populate_directions = FALSE,  
  directions_language = NULL,  
  directions_style_name = NULL,  
  save_route_data = FALSE,  
  save_output_layer = FALSE,
```

```

    populate_stop_shapes = FALSE,
    output_format = "feature_set",
    ignore_invalid_order_locations = FALSE,
    point_barriers = NULL,
    line_barriers = NULL,
    polygon_barriers = NULL,
    token = arcgisutils::arc_token()
)

```

Arguments

orders	An sf or sfc object containing point geometries representing locations to visit.
depots	An sf or sfc object containing point geometries representing depot locations.
routes	A data.frame describing vehicle and driver characteristics. Default: NULL.
breaks	A data.frame of rest period definitions for routes. Default: NULL.
travel_mode	Character. The name or ID of the travel mode to use. See get_travel_modes() for available options. Default: NULL.
analysis_region	Character. Region for the analysis. One of: "europe", "japan", "korea", "middle_east_and_africa", "north_america", "south_america", "south_asia", "thailand". Default: NULL (auto-detected).
time_zone_usage_for_time_fields	Character. Time zone used for all date-time input fields. One of: "geo_local", "utc". Default: NULL (API default: "geo_local").
default_date	POSIXct. The date on which all routes start. Only the date portion is used. Default: NULL.
time_units	Character. Units for all time-based attribute values. One of: "seconds", "minutes", "hours", "days". Default: NULL (API default: "minutes").
distance_units	Character. Units for all distance-based attribute values. One of: "miles", "kilometers", "meters", "feet", "yards", "nautical_miles". Default: NULL (API default: "miles").
time_window_factor	Character. Importance of honoring time windows. One of: "low", "medium", "high". Default: NULL (API default: "medium").
spatially_cluster_routes	Logical. Whether orders assigned to a route are spatially clustered. Default: TRUE.
route_zones	An sf or sfc polygon object delineating work territories for routes. Default: NULL.
route_renewals	A data.frame of intermediate depot renewal locations. Default: NULL.
order_pairs	A data.frame pairing pickup and delivery orders. Default: NULL.
excess_transit_factor	Character. Importance of reducing excess transit time for order pairs. One of: "low", "medium", "high". Default: NULL (API default: "medium").

uturn_policy	Character. U-turn policy at junctions. One of: "allow_uturn", "allow_dead_ends_and_intersections", "allow_dead_ends_only", "no_uturns". Default: NULL.
restrictions	Character vector. Restriction names to apply. Default: NULL.
impedance	Character. Impedance type. One of: "travel_time", "minutes", "truck_travel_time", "truck_minutes", "walk_time", "miles", "kilometers". Default: NULL.
time_impedance	Character. Time-based impedance. One of: "minutes", "travel_time", "walk_time", "truck_minutes", "truck_travel_time". Default: NULL.
distance_impedance	Character. Distance-based impedance. One of: "miles", "kilometers". Default: NULL.
use_hierarchy_in_analysis	Logical. Whether to use the street network hierarchy when finding routes. Default: TRUE.
populate_route_lines	Logical. Whether output routes include the exact street shape. Default: TRUE.
route_line_simplification_tolerance	Numeric. Simplification tolerance for route geometry. Default: NULL.
populate_directions	Logical. Whether to generate driving directions for each route. Default: FALSE.
directions_language	Character. Language code for directions (e.g. "en"). Default: NULL.
directions_style_name	Character. Formatting style for directions. One of: "desktop", "navigation", "campus". Default: NULL.
save_route_data	Logical. Whether to save results as a .zip file geodatabase. Default: FALSE.
save_output_layer	Logical. Whether to save the analysis as a network analysis layer package. Default: FALSE.
populate_stop_shapes	Logical. Whether output stops include point geometries. Default: FALSE.
output_format	Character. Format for output features. One of: "feature_set", "json_file", "geojson_file". Default: "feature_set".
ignore_invalid_order_locations	Logical. Whether to ignore invalid orders instead of failing. Default: FALSE.
point_barriers	Point barriers as sf or sfc object. Default: NULL.
line_barriers	Line barriers as sf or sfc object. Default: NULL.
polygon_barriers	Polygon barriers as sf or sfc object. Default: NULL.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A `route_vehicles_job` R6 object inheriting from `arcgisutils::arc_gp_job`. Call `$start()` to submit and `$results` to retrieve output.

References

[API Reference](#)

See Also

Other async: [download_od_results\(\)](#), [download_service_area_results\(\)](#), [find_closest_facilities_job\(\)](#), [find_routes_job\(\)](#), [find_service_areas_job\(\)](#), [last_mile_delivery\(\)](#), [location_allocation_job\(\)](#), [od_cost_matrix_job\(\)](#)

Other vrp: [last_mile_delivery\(\)](#), [route_vehicles\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

orders <- st_sf(
  name = c("Order 1", "Order 2"),
  geometry = st_sfc(
    st_point(c(-0.1891, 51.5254)),
    st_point(c(-0.1744, 51.5353)),
    crs = 4326
  )
)

depots <- st_sf(
  name = "Depot1",
  geometry = st_sfc(st_point(c(-0.2, 51.5)), crs = 4326)
)

job <- route_vehicles_job(orders, depots)
job$start()
result <- job$results

## End(Not run)
```

Description

Snaps a set of GPS points to the most likely roads traveled and optionally returns the traversed road segments as lines using the ArcGIS /SnapToRoads service.

Usage

```

snap_to_roads(
  points,
  travel_mode = NULL,
  point_properties = c("oid_routing_streets", "posted_speed_limit_kph",
    "posted_speed_limit_mph", "posted_speed_limit_mps", "posted_truck_speed_limit_kph",
    "posted_truck_speed_limit_mph", "posted_truck_speed_limit_mps"),
  line_properties = c("oid_routing_streets", "length_kilometers", "length_miles",
    "posted_speed_limit_kph", "posted_speed_limit_mph", "posted_speed_limit_mps",
    "posted_truck_speed_limit_kph", "posted_truck_speed_limit_mph",
    "posted_truck_speed_limit_mps"),
  analysis_region = c("europe", "japan", "korea", "middleeastandafrika", "northamerica",
    "southamerica", "southasia", "thailand"),
  return_lines = TRUE,
  token = arc_token()
)

```

Arguments

points	An sf or sfc object containing the point geometries to snap to the road network. Recognized attribute columns such as track_id and location_timestamp are used when present.
travel_mode	Character. The name of the travel mode to use. See <code>get_travel_modes()</code> for available options. Default: NULL.
point_properties	Character vector. Road properties returned on the snapped points output. One or more of: "oid_routing_streets", "posted_speed_limit_kph", "posted_speed_limit_mph", "posted_speed_limit_mps", "posted_truck_speed_limit_kph", "posted_truck_speed_limit_mph", "posted_truck_speed_limit_mps".
line_properties	Character vector. Road properties returned on the output lines. One or more of: "oid_routing_streets", "length_kilometers", "length_miles", "posted_speed_limit_kph", "posted_speed_limit_mph", "posted_speed_limit_mps", "posted_truck_speed_limit_kph", "posted_truck_speed_limit_mph", "posted_truck_speed_limit_mps". Ignored unless return_lines = TRUE.
analysis_region	Character. Region in which to perform the analysis. One of: "europe", "japan", "korea", "middleeastandafrika", "northamerica", "southamerica", "southasia", "thailand". Default: auto-detected from the location of points.
return_lines	Logical. Whether to return output lines representing the roads traversed. Default: TRUE.
token	Authorization token. Default: <code>arcgisutils::arc_token()</code> .

Value

A named list:

- `snapped_points`: the input points snapped to the road network

- snap_lines: the traversed road segments (only when return_lines = TRUE)
- usage_cost: list with numObjects and credits used
- messages: status and warning messages from the service

References

[API Reference](#)

See Also

Other direct: [find_closest_facilities\(\)](#), [find_routes\(\)](#), [find_service_areas\(\)](#), [od_cost_matrix\(\)](#), [route_vehicles\(\)](#)

Examples

```
## Not run:
# This example is not executed since it requires a network connection
# to ArcGIS Online and a valid authentication token
library(sf)
library(arcgisutils)
set_arc_token(auth_user())

pts <- st_sfc(
  st_point(c(-122.43410, 37.80016)),
  st_point(c(-122.43460, 37.80074)),
  st_point(c(-122.43539, 37.80096)),
  crs = 4326
)

snap_to_roads(pts)

## End(Not run)
```

Index

- * **async**
 - download_od_results, 3
 - download_service_area_results, 4
 - find_closest_facilities_job, 9
 - find_routes_job, 17
 - find_service_areas_job, 23
 - last_mile_delivery, 27
 - location_allocation_job, 31
 - od_cost_matrix_job, 38
 - route_vehicles_job, 46
- * **closest facility**
 - find_closest_facilities, 5
 - find_closest_facilities_job, 9
- * **direct**
 - find_closest_facilities, 5
 - find_routes, 13
 - find_service_areas, 20
 - od_cost_matrix, 35
 - route_vehicles, 42
 - snap_to_roads, 49
- * **location-allocation**
 - location_allocation_job, 31
- * **od**
 - download_od_results, 3
 - od_cost_matrix, 35
 - od_cost_matrix_job, 38
- * **routing**
 - find_routes, 13
 - find_routes_job, 17
- * **service area**
 - download_service_area_results, 4
 - find_service_areas, 20
 - find_service_areas_job, 23
- * **travel modes**
 - get_travel_modes, 26
- * **vrp**
 - last_mile_delivery, 27
 - route_vehicles, 42
 - route_vehicles_job, 46
- arcgisutils::arc_token(), 8, 12, 14, 19, 22, 25, 26, 29, 34, 37, 40, 44, 48, 50
- decode_compressed_geometry, 2
- decode_compressed_geometry(), 15
- download_od_results, 3, 4, 12, 19, 25, 29, 34, 38, 41, 49
- download_service_area_results, 3, 4, 12, 19, 23, 25, 29, 34, 41, 49
- find_closest_facilities, 5, 12, 15, 23, 38, 45, 51
- find_closest_facilities_job, 3, 4, 8, 9, 19, 25, 29, 34, 41, 49
- find_routes, 8, 13, 19, 23, 38, 45, 51
- find_routes_job, 3, 4, 12, 15, 17, 25, 29, 34, 41, 49
- find_service_areas, 4, 8, 15, 20, 25, 38, 45, 51
- find_service_areas_job, 3, 4, 12, 19, 23, 29, 34, 41, 49
- get_travel_modes, 26
- get_travel_modes(), 6, 10, 13, 18, 21, 24, 28, 32, 36, 39, 43, 47, 50
- last_mile_delivery, 3, 4, 12, 19, 25, 27, 34, 41, 45, 49
- location_allocation_job, 3, 4, 12, 19, 25, 29, 31, 41, 49
- od_cost_matrix, 3, 8, 15, 23, 35, 41, 45, 51
- od_cost_matrix_job, 3, 4, 12, 19, 25, 29, 34, 38, 38, 49
- route_vehicles, 8, 15, 23, 29, 38, 42, 49, 51
- route_vehicles_job, 3, 4, 12, 19, 25, 29, 34, 41, 45, 46
- snap_to_roads, 8, 15, 23, 38, 45, 49