Geometry-Based Optimization Heuristics for Region Coverage and Pathfinding in Drone-Based Operations

Kemal İhsan Kılıç

Supervisor: Assoc. Prof. Leonardo Mostarda

Computer Science Division, University of Camerino

kemal.kemal@unicam.it

October 21, 2022

Outline:

- **Thesis Research Summary**

- **Static Op. - Things to Cover (e.g. Communication)**
  - **Milestone 1 (p. 6): Single BS Region Coverage case study.**
    A special case of "SCP (Set Cover Problem)".
  - **Milestone 2 (p. 13): Extended Region Coverage.**
    Multi BS, bigger region, Voronoi Tessellation.

- **Dynamic Op. - Things to Visit (e.g. Transportation)**
  - **Milestone 3 (p. 17): Boat Rescue case study.**
    Coverage with CSs, TSP (Traveling Salesman Problem), and SP.

- **Epilogue**

Thesis in a few words:

Can the geometry of entities inspire novel heuristic methods for better optimization?

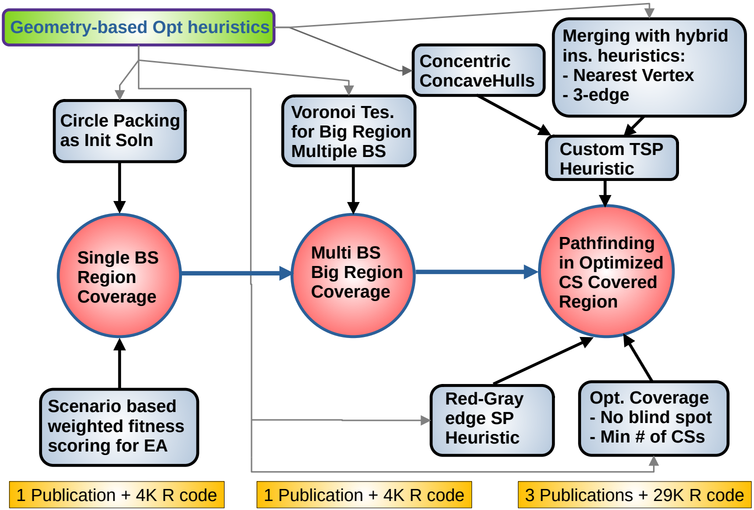
Proposal of **a novel geometry-based approach** for $\mathcal{NP}$**-Hard optimization problems** (case studies in drone operations):

The **geometry of the entities**, **can be exploited and/or regulated for novel approximation heuristics** towards **faster and improved optimizations**.

Research Method in a few words:

- Geometry-based heuristics are **hypothesized**.
- **Novelty/contribution/impact** are checked/evaluated with **Lit. Survey/Research Questions**.
- When possible, **theoretical/probabilistic perf. analysis** is carried out for the proposed heuristics.
- Heuristics are **benchmarked against “Base Case” (no-heuristic)**. When possible, against standard algos.
- When possible, (known soln/for small N), **AR (Approximation Ratio)** is calculated for the proposed heuristics.

Research Summary: “Representation Matters!”



Significance and Impacts of the research:

- **Practically:**
  - **Addressed limited on-board energy issues** for EVs.
  - **Contributed with various novel geometry-based optimization heuristics** to the effective operations (Static/Dynamic) with such vehicles.
  - Will have **practical impacts on Sustainability, Smart Cities, IoT, Industry 4.0, effective use of EVs.**
- **Theoretically:**
  - **Proposed geometry-based approximation heuristics** for $\mathcal{NP}$-Hard optimization problems.
  - **Proposed multi-party multi-objective optimization frameworks**.
  - Will have **theoretical impacts on SCP, TSP, VRP, and Geometry-based optimizations.**

(Vision & Goals: A1, A2 - Research Paradigm: A3)

Drones (Flying IoT A20): Pros and Cons†

Pros
Straight flight over obstacles
Precision (GPS)
Easy deployment
Autonomous mission
Quality aerial imaging
Small size can be useful
Can carry diverse types of sensors

Cons
Limited onboard energy
Little wind can effect flight
Uncertain regulations
Small size can be problem
Privacy problems

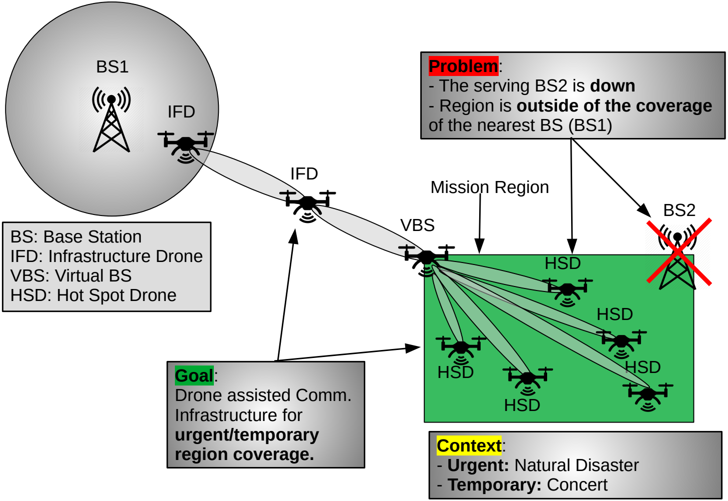
† Partly from: https://onlinemasters.ohio.edu/blog/the-pros-and-cons-of-unmanned-aerial-vehicles-uavs/

Milestone1

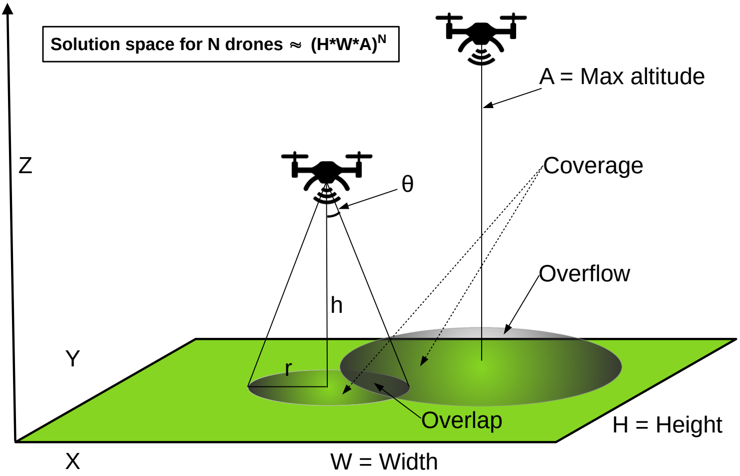
Static Drone Operation: Single BS Region Coverage

Ref:1

Region Coverage: Problem, Goal, and Context



Region Coverage: Drone Parameters



Research in short

Proposal of a multi-party multi-objective optimization framework for drone-based region coverage in which various EAs (Evolutionary Algos) are benchmarked.

Parties and objectives:

- **The operation region:** In need of temporary or urgent connectivity. **Main obj.:** Ideally 100% coverage is required.
- **Operation drones:** Available drones considered in the optimization. **Main obj.:** Energy optimization (distance).

Region Coverage: Research Gaps

Limitations of related work ([1, 14, 27, 4, 2]) and **contributions**:

- (in some) Single-party opt., mostly focused on QoS parameters: **Multi-party multi-objective scheme.**
- (in some) "Coverage Score" with conflicting objectives was not elaborated: **Scenario (A4) based weighted scoring (normalized scores) of novel objectives (A3).**
- (no) Load balancing with multiple BSs was not studied: **Division of the region with Voronoi Tessellation.**
- (no) Accelerating optimization in EAs was not considered: **CP (A8) algo for initial solution to help EA for better opt.**

(RQs: Appendix 1 and 2)
(Process Flow: Appendix 10)

Region Coverage: Results Summary†

EA	Soln Type	Best for
DEoptim	Population	Run time
GA	Population	Sum of drone distances (energy)
GenSA	Single solution	Coverage

(Example coverage: Appendix 9)

† Complete benchmark results in A6 and A7

Region Coverage: Research Impacts

- Base Station deployment for cellular networks (GSM) and wireless networks (IEEE 802.11)
- Energy efficient monitoring in WSNs
- Scheduling
- Optimum industrial cutting
- VLSI testing
- Inspection, Precision Agriculture, Disaster Management
- Strategic deployment of delivery stations (transportation logistics).
- Theoretical study of the “SCP (geometric, score-based)”.

Region Coverage: Publication 1

Kilic K.I., Gemikonakli O., Mostarda L. (2020) **Multi-objective Priority Based Heuristic Optimization for Region Coverage with UAVs**. AINA 2020. Advances in Intelligent Systems and Computing, vol 1151. Springer, Cham.
DOI: `https://doi.org/10.1007/978-3-030-44041-1_68`

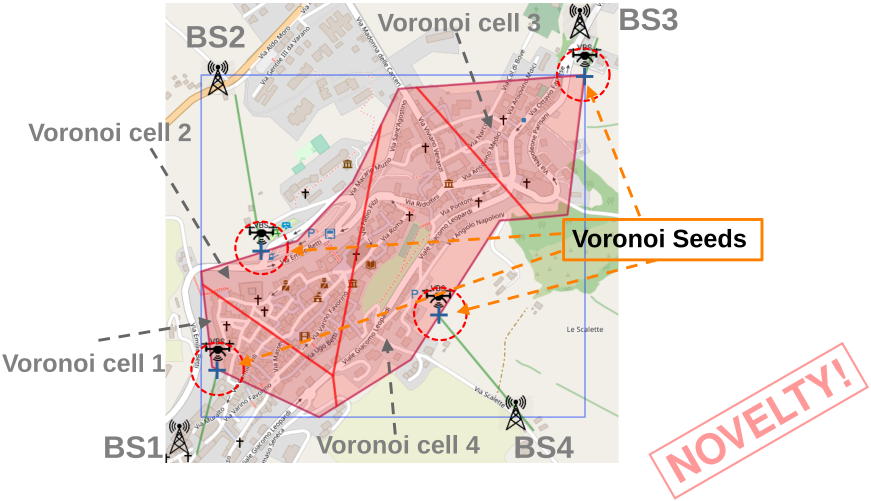
Milestone2

Static Drone Operation: Multi BS Region Coverage

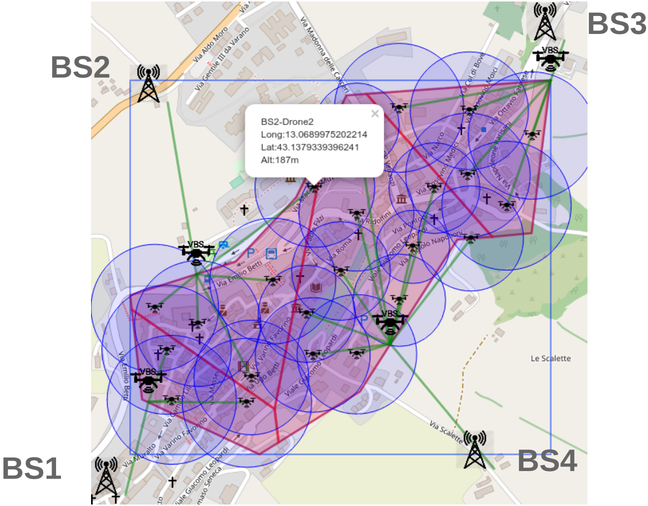
Extensions to Region Coverage

- Web deployable **simulator GUI for selecting real world region** and parameters (Screenshot: A2).
- Utilization of **multiple BSs** (Process Flow: A1).
- **Voronoi Tessellation of the operation region** for load balanced comm.
- Available **drones divided** into sub regions **proportional to the area**.

Example Voronoi Tessellation



Example Coverage



Region Coverage: Publication 2

Kilic, K. I., Gemikonakli, O. and Mostarda, L. (2021), **Voronoi Tesselation-based load-balanced multi-objective priority-based heuristic optimisation for multi-cell region coverage with UAVs**,
International Journal of Web and Grid Services 17(2), 152-178.
DOI: `https://doi.org/10.1504/IJWGS.2021.114574`

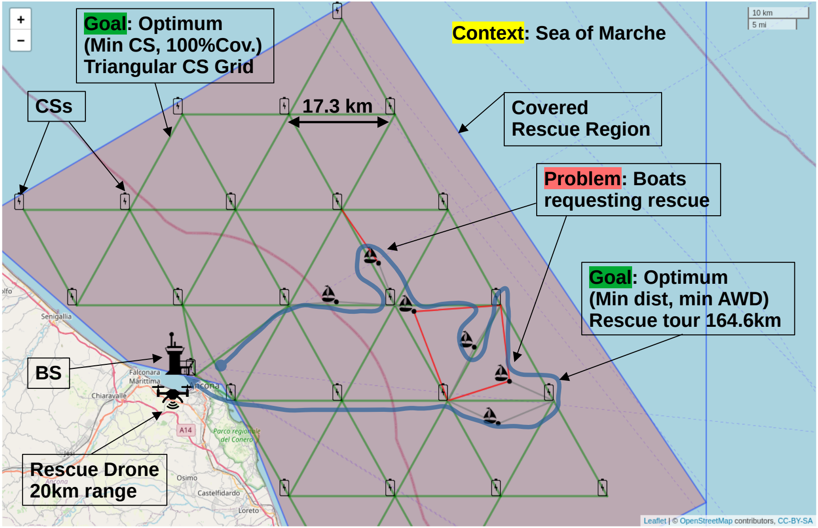
- **Contrib. to Coverage:** **Multi BS coverage** framework for Voronoi Tessellated operation region, utilization of homogeneous/heterogeneous BSs.
- **Contrib. to Voronoi Tess.:** Application of the Voronoi Tessellation with **homogeneous/heterogeneous site points** (A3).

Milestone3

Dynamic Drone Operation: CS Coverage and Pathfinding
1 Drone / 1 BS / N Boats

Ref:1

Boat Rescue: Problem, Goal, and Context(A47)



Boat Rescue Parts:

- **Foundational CS Grid** covering the region (p. 23)
- **Local Pathfinding Heuristic** (boat to boat): **redGraySP** (p. 26)
- **Global Pathfinding Heuristic** (TSP tour of boats): **concaveTSP** (p. 29)

(Process Flow: Appendix 52)

Research in short

Proposal of a multi-party multi-objective optimization framework for drone-based operations in which the synergy between the CS Grid and the Pathfinding is established and exploited:

- **The CS Grid:** Optimized for **region coverage**† through geometry-based heuristics for the **operation region**. **Triangular and Square** Grids are considered. [CS Techs: (A2, A3, A4, A5)]
- **The Pathfinding Algorithm:** Optimized for **energy**† through geometry-based heuristics for the **operation drones**.
  - **Local search (BT2BT), redGraySP:** Augments SP/A* with dynamic constraint edges (Red-Gray).
  - **Global search (All BT), concaveTSP:** A novel TSP for multi boats.

†The main goal. List of objectives: A1

Boat Rescue: Research Gaps - CS Grid

Limitations of related work ([24, 17, 11, 12]) and **contributions**:

- Optimal CS deployment studied **only in the context of adjustable (mobile) CS Grid**: Proposal of **static optimal (min CS - no blind spot) CS grid geometries** (p7) adjusted to drone range (A6) for complete region coverage and the **coverage effectiveness** metric (p10).

- Optimal CS deployment studied only for coverage, **synergy with pathfinding not considered**: Proposal of the **novel TSP** (A17) **+ redGraySP pathfinding heuristics** (p11).

Boat Rescue: Research Gaps - Pathfinding

Limitations of related work ([24, 17, 11, 12]) and **contributions**:

- In pathfinding (Fuel Constrained, UAV Routing Problem (FCURP)) studies, either the **region is assumed to be covered** or the CSs are **assumed to be mobile**: Proposal of the **synergistic CS deployment and the pathfinding benefiting from the regular configuration** of the CS grid.

Boat Rescue: Research Gaps - TSP

Limitations of related work ([22, 7, 8, 15, 18, 16, 13]) and **contributions**:

- For **geometric instances of TSP, concave hull-based heuristics** were not studied: Proposal a **novel concave hull-based TSP heuristic** (A32).
- In **multi objective TSP only generic objectives** are assumed: Proposal of **novel metrics and multi-party/multi-objective optimization** scheme.
- **Mostly TSP datasets are biased**: Proposal of **analyzing various statistical/geometrical properties** of the data sets other than their sizes for better benchmarking. **Custom "Grid" dataset** generation.

Foundation
Optimum CS Deployment

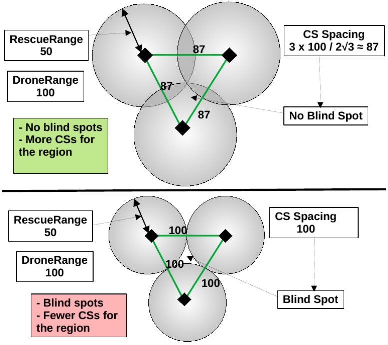
Region coverage with:

Minimum CSs
No blind-spots

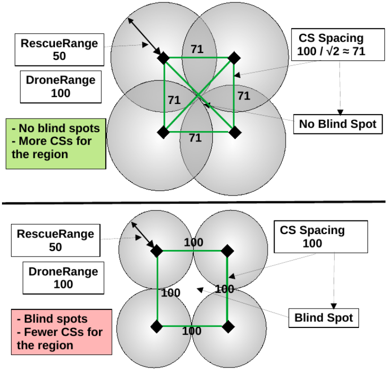
Ref:2

CS Grid design without "blind-spots"

RQ: How to deploy min number of CSs without any "blind-spot"?



Triangular CS Grid

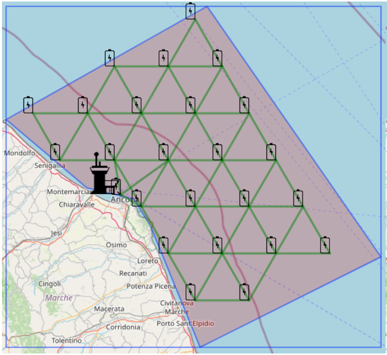


Square CS Grid

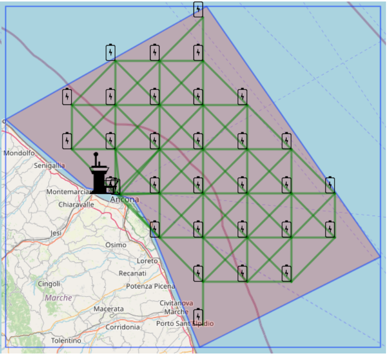
Ref:4

CS Grid design to cover the region I

RQ: How to deploy CSs to cover the region?



Actual **triangular grid CS deployment** **(24 CS)**.
AVG SP length from BS = 2.06 units (unit = Drone range).

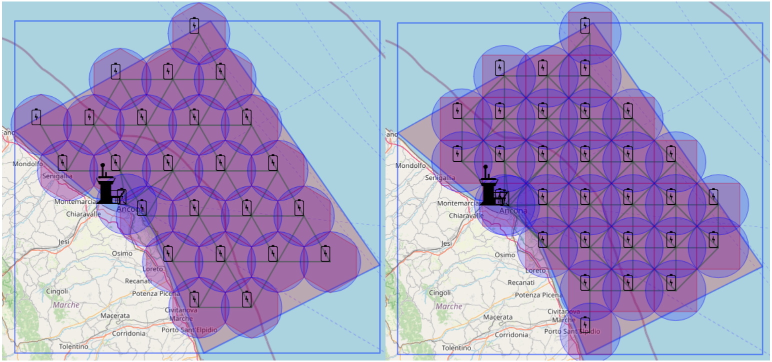


Actual **square grid CS deployment** **(30 CS)**.
AVG SP length from BS = 2.07 units (unit = Drone range).

Actual CS grid deployments on the sea of Marche, Italy. The selected region is 5992 km^2. The bounding box has width: 120.93 km and height: 109.38 km. **Tri Grid wins!**

CS Grid design to cover the region II

RQ: How to deploy CSs to cover the region?



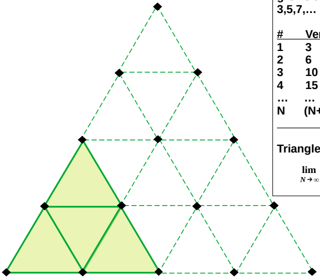
Actual **triangular grid CS deployment (24 CS)**.
AVG SP length from BS = 2.06 units (unit = Drone range).

Actual **square grid CS deployment (30 CS)**.
AVG SP length from BS = 2.07 units (unit = Drone range).

Actual CS grid deployments on the sea of Marche, Italy. The selected region is 5992 km^2. The bounding box has width: 120.93 km and height: 109.38 km. **Tri Grid wins!**

Novel metric: Coverage effectiveness

RQ: Which one has better “coverage” (more area for each CS)?

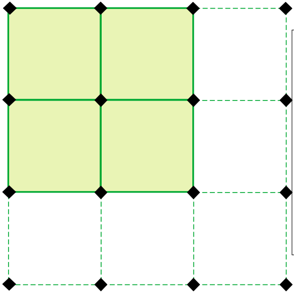


Adding 3,4,5,... vertices grows the area by 3,5,7,... triangles

#	Vertices	Triangles
1	3	1
2	6	4
3	10	9
4	15	16
...	...	...
N	(N+1)(N+2) / 2	N^2

Triangles per vertex limit:

$$\lim_{N\to\infty} \frac{N^2}{\frac{((N+1)*(N+2))}{2}} = 2$$



Adding 5,7,9,... vertices Grows the area by 3,5,7,... squares

#	Vertices	Squares
1	4	1
2	9	4
3	16	9
4	25	16
...	...	...
N	$(N+1)^2$	N^2

Squares per vertex limit:

$$\lim_{N\to\infty} \frac{N^2}{(N+1)^2} = 1$$

Tri: 1.30 $unit^2$

Sq: 0.5 $unit^2$

Coverage effectiveness: Covered unit (=Drone range) area per CS.

Tri Grid wins! (Consider limits as $N \to \infty$!)

NOVELTY

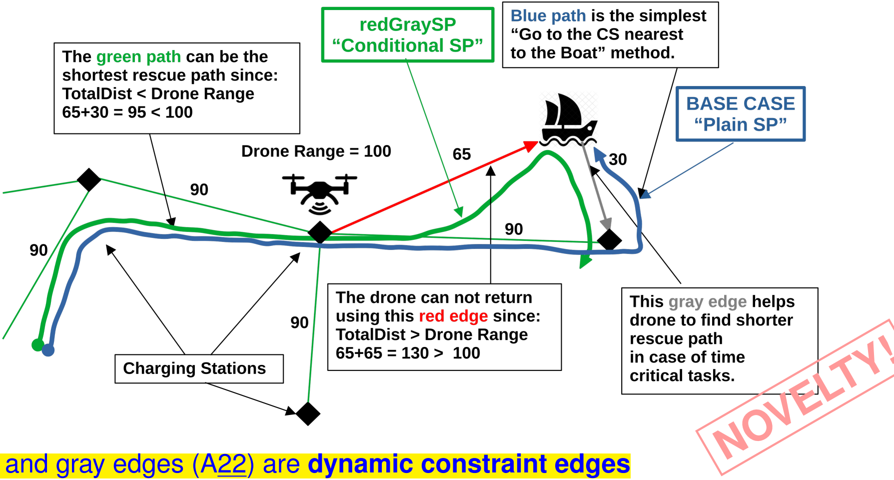
Ref:4

Local Pathfinding Heuristic redGraySP:

Boat to boat pathfinding (Augmented SP/A*)

Red-gray path heuristic: redGraySP

RQ: Can we exploit CS config. for better pathfinding? (Synergy ↑↓)



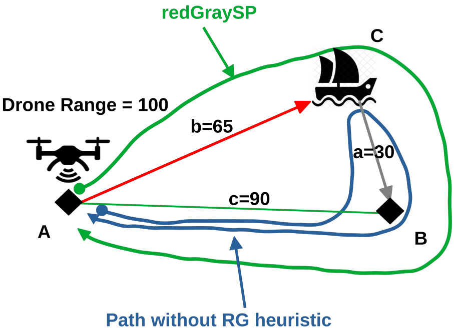
Red and gray edges (A22) are **dynamic constraint edges** (red+gray $\leq$ droneRange). They disappear when the boat is rescued!

Ref:4

Geometric proof for redGraySP

RQ: Can we prove that redGraySP $<$ SP?

Geometric proof for redGraySP is shorter SP



ABC is triangle, inequalities:

a < b + c	(1)
b < a + c	(2)
c < a + b	(3)

redGraySP path:

b + a + c	(4)

SP path (without RG heuristic):

2c + 2a	(5)

Compare:

(b + a + c) ? (2c + 2a)	(6)
(b + $\cancel{a}$ + $\cancel{c}$) ? ($\cancel{2}$c + $\cancel{2}$a)	
(b) ? (c + a)	(7)

From (2):

(b) < (c + a)	(8)

∴ redGraySP (4) < SP (5)

Triangular vs Square CS Grid (Theoretical)

RQ: How redGraySP performs on different CS grid types?

Grid type	Prob. of having a Good RG Path	Prob. of using a Good RG Path	Savings 1-way	Savings Return	Operation Area per CS
Triangular	0.78	0.45	43%	20%	1.30 $unit^{2\dagger}$
Square	0.82	0.31	64%	50%	0.5 $unit^2$

Analysis: p48, p49, p50, p51

Verdict (Theoretical): Tri gives better (less CSs and more area per CS) coverage.

RQ: In practice, how is Tri vs Sq Grid comparison? Multi BT Simulations!

† 1 unit = Drone range

Global Pathfinding Heuristic concaveTSP:

Find optimum rescue order for **multiple boats**

Ref:2

Multiple boats: Design issues

RQ: In which order to rescue boats?

For a **single drone** the optimum order = Shortest rescue tour = TSP tour!

RQ: Can we design a fast algorithm that considers geometric configurations of boats and finds the optimum rescue order?

We need to try Geometric TSP!

Multiple boats: Method

In essence a TSP+SP algo (A23):

1. **TSP** will give the **“optimum” order** to rescue boats.
   - Paths between boats are “dynamic” → Use **Euclidean Dist. between boats for TSP**.
   - Boats have **same priorities**.
   - Tour cost + **novel metrics**: AWD† and minAWD (A26, A27, and A28).
   - **CW and Anti-CW tours** (Same tour len. but diff. AWD).
   - Weights for objectives: **Min tour, min AWD, num chargings**.
2. The red-gray heuristic(A29, A30) will find a **SP between boats**.

concaveTSP [Summary: A32 - Demo: A17 - Algo: A24]

† Average Waiting Distance

Multi Boat Sim Params.

Total 1600 sims:

- **Goals: Benchmark proposed methods** and **compare Tri-Sq**.
- Tri and Sq CS grids compared $\rightarrow \quad \times 2$
- 20, 40, 60, 80, 100 randomly generated Boats $\rightarrow \quad \times 5$
- concaveTSP, Farthest Ins, Nearest Neighbor, 2-OPT benchmarked $\rightarrow \quad \times 4$
- With red-gray heuristic and without (base case=Go to the CS nearest to the Boat!) $\rightarrow \quad \times 2$
- AVG of 20 simulations for each measurement $\rightarrow \quad \times 20$

Simulator GUI (A8)
Examples (1BT: A9, A10, A11, A12, 6BT:A13, A14)

Tri vs Sq: Theo. + Sim Results Summary

Type	Metric	Tri Grid	Sq Grid
Theoretical SingleBoat	Prob. of having a Good RG Path	0.78	0.82
	Prob. of using a Good RG Path	0.45	0.31
	Savings 1-way	43%	64%
	Savings return	20%	50%
	Area per CS	1.30 $unit^{2\dagger}$	0.5 $unit^2$
Simulations MultiBoat	Tour Cost	Higher	Lower
	*Tour Cost Savings%	Higher	Lower
	AWD	Higher	Lower
	*AWD Savings%	Higher	Lower
	Chargings	Not much difference	Not much difference
	*Chargings Savings%	Higher	Lower
	*Number of CSs	Lower (24)	Higher (30)

Verdict (Sim. results): The **redGraySP**, "saves path length" in the range of 10-17% over the "base case".

Tri gives better (over base case) savings and better coverage, but the "tour" is more expensive!

$\dagger$1 unit = Drone range

Perf. of the methods: Sim Results Summary

Plots:

- AVG Runtimes (A31)
- AVG Tour Cost (A32)
- AVG AWD (A33)
- AVG Num. of Chargings (A34)

Tables (With OPT soln):

- 4 boats 20 sims (A40)
- 5 boats 20 sims (A41)
- 6 boats 20 sims (A42)
- 7 boats 20 sims (A43)

Tables (Comparison):

- 20 boats 20 sims (A35)
- 40 boats 20 sims (A36)
- 60 boats 20 sims (A37)
- 80 boats 20 sims (A38)
- 100 boats 20 sims (A39)
- Tour costs big dataset (A44)
- Runtime big dataset (A45)
- AWD costs big dataset (A46)

Verdict (Sim. results): **concaveTSP** is **on par** with other TSP approximation methods for **small number of boats** (<1000 vertices). For **big datasets** (1000+ vertices, regular and hexagonal grid) concaveTSP is **competitive (fast, shorter tour)**.

Pathfinding in CS Grid: Research Impacts

- Coverage framework for general EFV:
  Cost savings and less carbon footprint
- Bigger operation range/area for EFVs
- Optimized delivery with EFVs
- Optimized industrial place inspection with EFVs
- Theoretical impacts on SCP, TSP, and VRP

Boat Rescue: Publications 3-5

- Kilic K.I., Mostarda L. (2021) **Optimum Path Finding Framework for Drone Assisted Boat Rescue Missions**. In: Barolli L., Woungang I., Enokido T. (eds) Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems, vol 227. Springer, Cham.
DOI: `https://doi.org/10.1007/978-3-030-75078-7_23`
- Kilic K.I., Mostarda L. (2021) **Heuristic Drone Pathfinding over Optimised Charging Station Grid**. IEEE Access, vol. 9, pp. 164070-164089,
DOI: `https://doi.org/10.1109/ACCESS.2021.3134459`
- Kilic K.I., Mostarda L. (2022) **Novel Concave Hull-Based Heuristic Algorithm For TSP**, Operations Research Forum, Springer Nature, 3(2):25, DOI: `https://doi.org/10.1007/s43069-022-00137-9`

Epilogue

Idea: Geometry Helps!

How did proposed heuristics utilize Geometry?

- **They exploited the existing geometric configurations of the entities:** Voronoi-Delaunay Structs., Convex/Concave Hulls.
- **They introduced geometric regularities for their configurations:** Grids, Circle Packing.
- **Algebra that "sees" through Geometry!**

> Paraphrasing Sophie Germain:
>
> Geometry draws Algebra and Algebra writes Geometry!

Geometric Techs. Used

Basic novel recipes proposed:

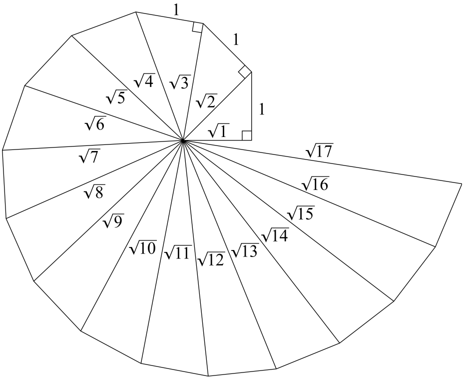
- **Voronoi-Delaunay Structs:** Good for "Neighborhood" discovery and "near-distant region" division.
- **Grids (Special Voronoi-Delaunay Struct):** Regularity can be exploited for pathfinding. Extreme cases with "Manhattan Distances" (A7).
- **Convex/Concave Hulls:** Good for sorting vertices topologically and distance wise.
- **Circle Packing (CP):** Good initial soln for SCP with Evolutionary Algos (EAs).

Ref:1

Representation matters!

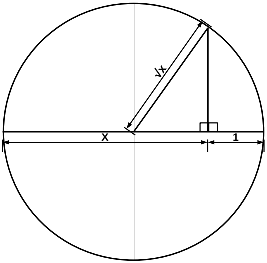
- $1614 \times 564 = 910296$
- Roman Numerals:
  $\text{MDCXIV} \times \text{DLXIV} =$ Error: number too big!

Geometry as Universal Language?



Spiral of Theodorus.

https://en.wikipedia.org/wiki/Spiral_of_Theodorus



$\sqrt{x}$ on a circle.

https://www.youtube.com/watch?v=WD-Ebz5No5Y

https://en.wikipedia.org/wiki/Geometric_mean

Representation matters! Finding $\sqrt{x}$ with Geometry.

Region Coverage: Future Works

- **Heterogeneous/Random drone altitudes** can be tried for generating initial solutions.
- **Heterogeneous BSs** can be tried by utilizing “Weighted Voronoi Tessellation”.

Pathfinding in CS Grid: Future Works

- Decentralized control.
- Single BS - Multi drone scheme (VRP case).
- Multi BS - Single drone per BS scheme with Voronoi Tesselation of the region.
- Multi BS - Multi drone per BS scheme (VRP + Voronoi Tesselation).
- Boat to boat rescue “leaps” through “bin-packing”.

Thanks

Questions?

Appendix

Research Visions, Motivations and Goals

In general:

- “Ground traffic” problems: **Expensive, accidents, high land-use, high carbon footprint**.
- EFVs/EVs are **versatile** and have **great future and benefits**.
- **Optimization for the best usage of the onboard energy**.
- **Smart optimized “charging” infrastructure** for **greater range of operation** with EVs/EFVs.
- **Geometric representations** of entities help for heuristic optimizations.

Ref:4

Research Visions, Motivations and Goals

Specifically:

- Drones are useful **mobile IoT platforms**.
- They help for **temporary conn. away from BSs** - when the **BS is down**.
- **Coverage**(static) and **Pathfinding**(dynamic) are two fundamental operations for drones.
- **Geometry of entities can be exploited/regulated for better optimization heuristics in these operations**.
- **Establish and explore the interdependence and synergy** between CS grid config and the pathfinding for **optimum coverage and optimum flight distance**.

Ref:4

Overall Research Paradigm

- **Quantitative research**: Benchmarking, metrics, measurements.
- **Positivist research paradigm**:
  - **Ontological view**: The "optimum" exist and quantifiable.
  - **Epistemological view**: The "optimum" can be measured, but difficult to find → $\mathcal{NP}$-Hard problem → **Approx. algorithms**. (Von Neumann Arch & binary logic are assumed).
- **Experimental methodology**: Algorithm benchmarks.
- **Statistical verification for measurements**: Results are verified/explained with statistical tests.
- **My default meta RQs:**
  - How can I contribute? Propose novel and useful stuff
  - How can I generalize? Propose paradigm, principle

Ref:4

Research Steps

The activities related to the research process can be listed as:

- Literature Review
- Determination of research gaps
- Discussion of research gaps and research questions for novel contributions
- Theoretical analyses for methods and proofs
- Implementation of proposed models and methods
- Experimental verification of proposed models and methods
- Statistical verification of the results

Region Coverage: Research Questions

1. (Possibility) How difficult is to find optimum (min cost) complete coverage?
   - $\mathcal{NP}$-Hard problem → **Approx. algorithms**.
   - Complex problem formulation → **Heuristic EAs (GA/SA)**.
   - Multi-objective multi-party problem → **Trade-offs** → **Optimization**.
2. (How) How can we effectively optimize the coverage? Parties, essential objectives?
   - Design **fitness func.** for EAs, supply **Init. Soln.** and **benchmark** several algos.
   - **Drones** and **the region** are the parties.
   - Objectives for drones: **Min distance of flight for drones (energy)**.
   - Objectives for the region: **Max coverage, min overlap, min overflow**.

Ref:4

Region Coverage: Research Questions

3. (Technical details) What kind of framework we can design for such optimization?
   - Assumptions for the experiments (A5): **Fixed number of drones, artificial region (400x300px), single BS**.
   - **Weighted sum of objectives** for different scenarios in fitness func.
   - **Different EAs:** Population based/Single soln, Stochastic/Systematic jumps to avoid local min/max.
4. (Extension) How can we extend the optimization framework for “bigger regions”?
   - Opt. needs **more time**, IFDs and VBS can be **overloaded** → **multi-BS, multi region**.
   - Divide and conquer → **Voronoi Tessellation** of the big region.
   - In **parallel optimization** and **load balanced** comm.

Ref:4

Priority based multi-objective scoring

$$\text{Score} = W_c \times CP + W_l \times OlpP + W_f \times OfwP + W_d \times TotDP$$

W_c, CP Coverage weight and percentage (Maximize)

$W_l, OlpP$ Overlap weight and percentage (Minimize)

$W_f, OfwP$ Overflow weight and percentage (Minimize)

$W_d, TotDP$ Distance weight and Normalized % total distance differences from the max distance† from all drones

$$TotDP = 100 \times \frac{(maxDist \times N - sumDist)}{(maxDist \times N)}$$ (Maximize)

NOVELTY!

†: Max 3D distance from the VBS in the operation region.

Ref:4

Scenarios for coverage

Table: Application cases and proposed set of weights for objectives.

Application type (Scenario)	W_c	W_l	W_f	$W_d^{\dagger}$
S1: Max coverage with no compromise	+	0	0	0
S2: Max coverage with only overlap/overflow penalty	+	-	-	0
S3: Max coverage with overlap/overflow penalty and min total distance of drones from VBS	+	-	-	+
S4: Max coverage with only min total distance of drones from VBS	+	0	0	+

†: Normalized total distance differences from the max distance in the mission region from all drones.

NOVELTY

Ref:4

Parameters: Region coverage experiments

Parameter	Value
Region length	400m
Region width	300m
Min height for drones	5m
Max height for drones	150m
Theta angle for drones	30°
Number of drones	2, 4, 6, 8, 10, 12
Algorithms	GA, GenSA, DEoptim

Ref:2

Benchmarking Results

- DEoptim was the **fastest**
- GenSA+InitSoln was the best in finding **highest covering ratio**
- GA with no InitSoln was the best in finding **min tot dist for the drones**

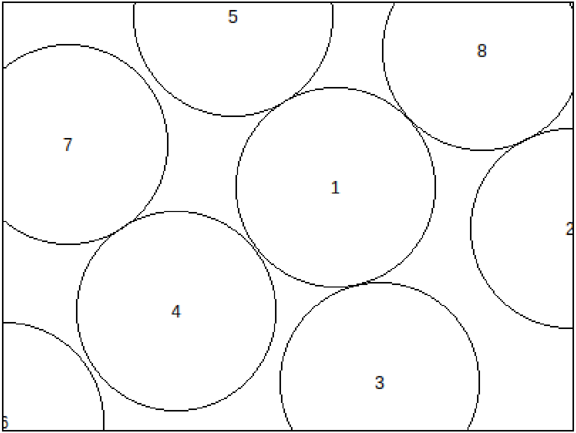
Ref:5

Drones	Scn	Time(sec)					Cov(%)						TDist(m)					
		DO	GA		SA		IS	DO	GA		SA		IS	DO	GA		SA	
			With	No	With	No			With	No	With	No			With	No	With	No
2	S1	13.82	209.3	361.92	743.39	729.64	21.94	40.05	40.05	40.05	40.05	40.05	475.6	565.26	564.74	596.96	666.55	597.95
	S2	15.27	196.86	207.18	712.06	719.62	21.94	40.05	40.05	40.05	40.05	40.05	475.6	652.55	608.76	627.69	636.88	540.39
	S3	39.78	163.48	178.62	716.18	699.62	21.94	40.04	37.74	40.05	40.04	40.04	475.6	508.32	500.19	509.66	507.35	508.7
	S4	30.9	202.51	226.14	711.26	710.78	21.94	39.56	39.36	39.13	39.56	39.56	475.6	493.09	488.73	484.75	493.21	493.42
4	S1	166.32	153.86	153.47	1000.78	1000.75	57.24	76.33	76.12	75.36	76.48	76.48	1322.59	1212.75	1215.67	1185.11	1232.22	1233.37
	S2	155.98	308.28	157.75	1001.01	1000.71	57.24	73.5	74.62	71.82	75.74	75.74	1322.59	1161.65	1177.85	1180.21	1201.64	1223.76
	S3	189.84	200.94	225.61	1000.92	1000.77	57.24	71.23	65.85	66.34	70.9	72.43	1322.59	1177.95	1083.5	1045.09	1097.27	1129.37
	S4	197.45	175.2	153.91	1001.12	1001.65	57.24	76.1	76.11	75.09	76.16	76.14	1322.59	1187.73	1194.66	1160.29	1185.27	1185.03
6	S1	427.51	511.38	577.94	1000.73	1000.75	69.59	95.86	95.8	95.38	96.42	96.43	1760.01	1880.97	1828.03	1851.19	1854.81	1853.23
	S2	378.26	251.65	236.42	1000.74	1000.72	69.59	87.65	83.69	86.43	87.1	89.48	1760.01	1841.02	1859.63	1753.23	1795.94	1793.11
	S3	390.51	325.48	425.98	1000.72	144.12	69.59	84.73	77.55	77.93	72.73	82.43	1760.01	1673.83	1530.65	1517.62	1275.02	1413.54
	S4	416.64	481.24	281.81	1000.76	1002.13	69.59	95.02	95.18	87.69	96.09	96.12	1760.01	1807.89	1796.11	1693.77	1813.75	1812.89
8	S1	269.4	437.34	524.35	1000.76	1000.77	75.61	94.63	98.07	99.24	99.9	99.97	2457.35	2534.07	2483.6	2431.52	2510.69	2499.34
	S2	269.91	458.04	364.65	1000.94	1000.72	75.61	78.96	83.72	83.04	90.68	89.75	2457.35	2343.94	2402.89	2270.75	2462.48	2486.55
	S3	647.31	445.35	502.83	179.23	189.01	75.61	76.98	81.8	80.6	84.85	76.96	2457.35	1688.09	2306.31	2227.58	1985.03	1638.37
	S4	694.71	372.77	394.13	1000.84	1001.92	75.61	96.14	98.37	98.08	98.7	98.32	2457.35	2120.46	2423.06	2293.7	2218.2	2178.36
10	S1	935.31	789.75	1652.59	377.51	363.53	67.29	99.98	100	99.98	100	100	3112.84	3118.6	3109.22	2981.74	3018.26	3136.28
	S2	394.58	609.55	1439.43	1000.79	1000.81	67.29	81.67	84.91	86.08	89.91	89.02	3112.84	2699.96	2796.58	2854.54	3206.29	2998.3
	S3	532	203.99	878.5	110.4	78.42	67.29	85.91	85.37	80.36	85.19	85.29	3112.84	2335.38	2561.08	2501.69	2257.09	2310.3
	S4	742.62	554.54	894.4	604.63	201.47	67.29	98.03	99.08	99.51	99.26	98.49	3112.84	2690.57	2657.05	2763.19	2520.76	2439.56
12	S1	772.97	638.99	1230.49	274.62	240.81	73.98	99.92	100	99.91	100	100	3518.39	3882.51	3617.53	3399.39	3663.7	3655.72
	S2	278.42	657.43	1016.11	1000.92	1002	73.98	69.41	86.77	84.23	89.53	89.46	3518.39	2853.03	3438	3278.39	3367.15	3230
	S3	602.03	1156.61	341.71	145.89	236	73.98	85.36	85.47	82.56	89.66	84.48	3518.39	3011.19	3423.38	2719.09	2776.8	2669.1
	S4	1296.59	744.6	528.75	174.49	192.89	73.98	97.09	99.86	98.67	97.18	98.76	3518.39	2860.62	3268.05	2947.51	2755.3	2943.88
S1 # of 1^{st}		3	0	1	0	2	-	1	3	1	4	6	-	0	2	4	0	0
S2 # of 1^{st}		5	0	1	0	0	-	1	1	1	5	3	-	3	0	2	0	1
S3 # of 1^{st}		2	0	0	2	2	-	2	0	1	2	1	-	0	1	1	2	2
S4 # of 1^{st}		1	1	2	1	1	-	1	1	1	3	2	-	1	0	3	1	1
Tot # of 1^{st}		11	1	4	3	5	-	5	5	4	14	12	-	4	3	10	3	4

The weights (W_c, W_l, W_f, W_d) were, (1, 0, 0, 0), (1, -1, -1, 0), (1, -1, -1, 0.5), (1, 0, 0, 0.5) respectively for scenarios 1, 2, 3, and 4 respectively.

Ref:5

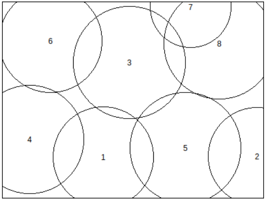
Initial Solution for Region Coverage



Initial solution for 8 drones (Hexagonal CP algorithm). Cov: 75.61%. TDist: 2457.35 m. Time: $<$ 1 sec. (Ref:4)

Ref:4

Region Coverage: Example Optimizations



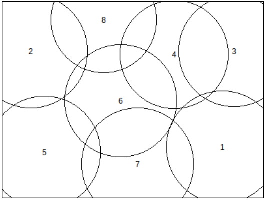
DEoptim [3]

Cov: 94.63%

TDist: 2534.07 m

Time: 269.40 secs

Best in Time



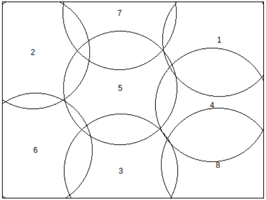
GA [23]

Cov: 99.24%

TDist: 2431.52 m

Time: 524.35 secs

Best in TDist(Energy)



GenSA [25]

Cov: 99.97%

TDist: 2499.34 m

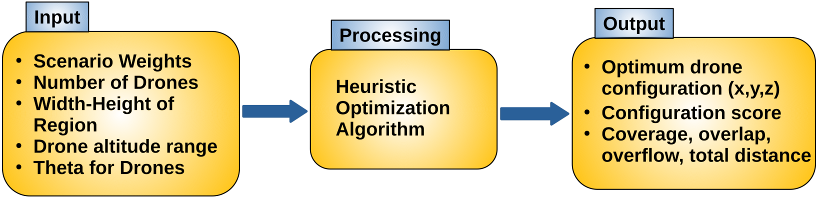
Time: 1000.78 secs

Best in Coverage

Solutions for 8 drones (no initial solution), overflow-overlap ignored, max region coverage. **VBS at UL**. (Good comparisons of EAs in [20])

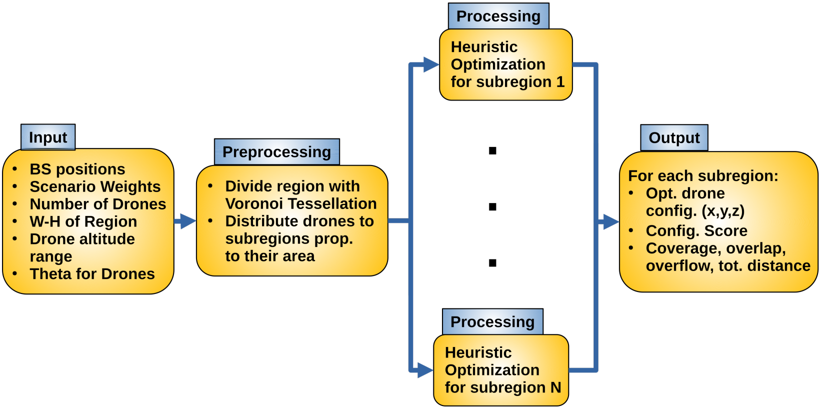
Ref:5

SingleBS Region Coverage: Process Flow



Ref:4

MultiBS Region Coverage: Process Flow



Ref:1

/my_hd1/my_dir/my_prg/Camerino/drone/shiny/droneCoverageWithMapUI - Shiny

http://127.0.0.1:3271 Open in Browser Republish

Select Region Pts OFF | ON Select BS Pos

Clear Selection | Close Polygon

Total drones[1-100]:

22

Avg Drone Alt for Coverage:

15 106 200

15 34 53 72 91 110 129 148 167 186 200

```
22 drones are required at 106 meter.
```

Show Voronoi | Show Init Soln | Find Coverage

Msg:

```
Click Selection: Lat: 43.132 Lng: 13.07 Zoom:
```

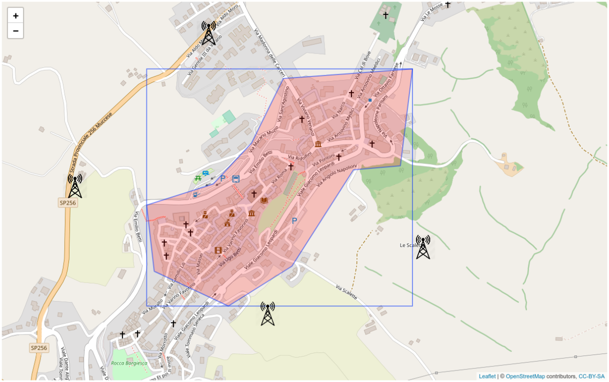
Click msg:

```
BS point[4] at: Lat: 43.132 Lng: 13.07
```

Mouse coord:

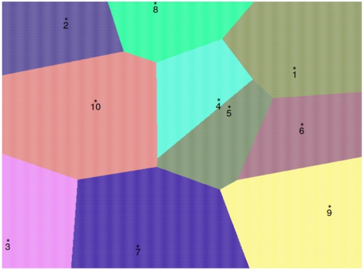
```
Lat: 43.133
Lng: 13.069
```

Save Map | Download Map

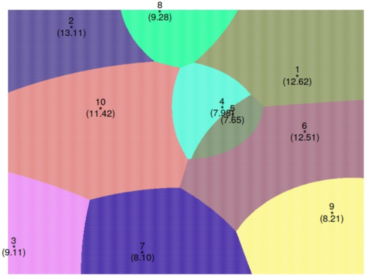


Ref:1

Voronoi Tessellation Types



Tessellation for homogeneous site points.



Tessellation for heterogeneous site points with normalized weights (percentages).

NOVELTY!

Ref:4

Boat Rescue: Objectives, preferences

- CS Grid related objectives:
  - **Min number of CSs**.
  - **No blind-spot** (unreachable spots) in the mission region.
- Pathfinding related objectives:
  - **Shortest Path/Tour** for the drone.
  - **Min Average Waiting Distance** in case of multiple boats.
  - **Min number of chargings** for the drone.
  - Boat **priorities** can be considered.

Ref:3

Research in charging techs for drones

- Contact based [5]
- Battery Swapping [19]
- Drone Swapping [9]
- Contactless:
  - Wireless Power Transfer [26]
  - Laser-Powered UAV wireless communication [21]

Ref:3

Contact-based CS

https://skycharge.de/charging-pad-outdoor

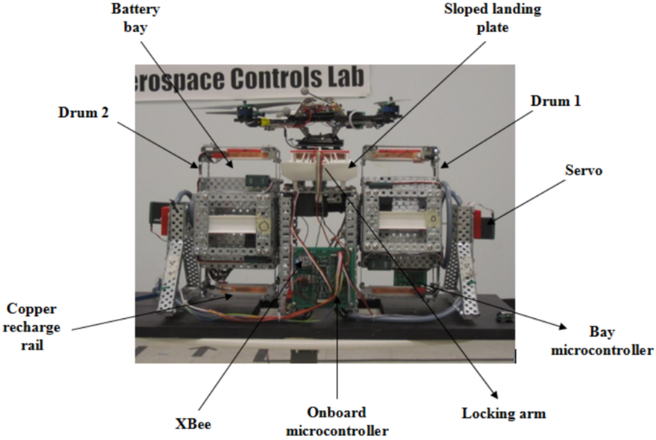
Ref:3

Contact-based CS: Contact points

https://skycharge.de/charging-pad-outdoor

Ref:3

Battery Swapping Station



Michini, B., Toksoz, T., Redding, J., Michini, M., How, J., Vavrina, M. & Vian, J. (2011), Automated Battery Swap and Recharge to Enable Persistent UAV Missions, in 'Infotech@Aerospace 2011'. http://dx.doi.org/10.2514/6.2011-1405

Ref:3

Boat Rescue: Observations

- The **drone range** is the crucial parameter for the framework.
- CS grid should be deployed based on the drone range to prevent “blind-spots”.
- CS grid should have a **regular geometry** covering the mission region.
- This geometry can be exploited for **easier path finding** (Never underestimate the power of regularity!)
- In extreme regularity even polynomial solution!

Ref:4

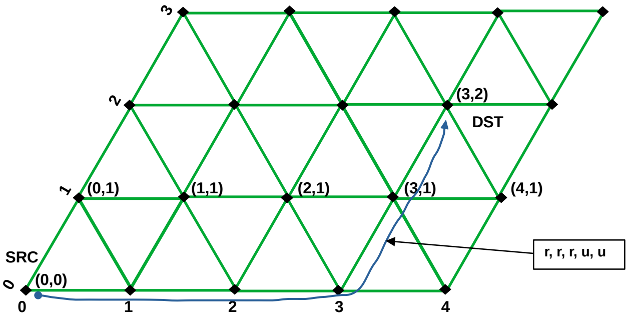
Extreme Regularity → "Easier path finding"

SP from (0,0) to (3,2) involves paths with length of 5 that goes 2 up* and 3 right. Total = $\binom{5}{2} = \binom{5}{3}$ = 10 paths
Ex:
u, u, r, r, r → (0,1), (0,2), (1,2), (2,2), (3,2)
u, r, u, r, r → (0,1), (1,1), (1,2), (2,2), (3,2)
up*: Special up, away from the SRC towards DST

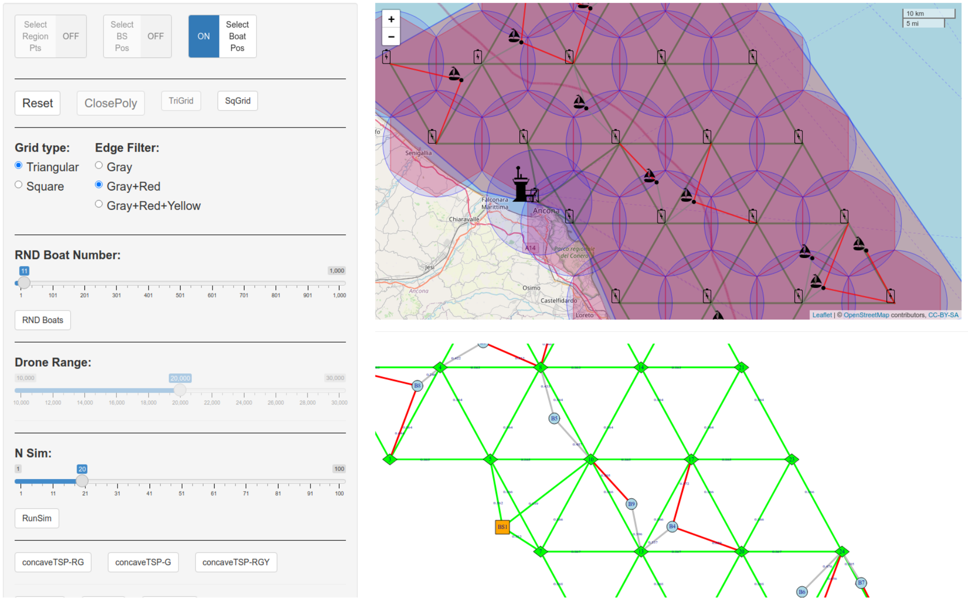
SP from (x_1, y_1) to (x_2, y_2)
Has path length = $|(y_2 - y_1)| + |(x_2 - x_1)|$

Number of paths = $\binom{|(y_2-y_1)|+|(x_2-x_1)|}{|(y_2-y_1)|}$

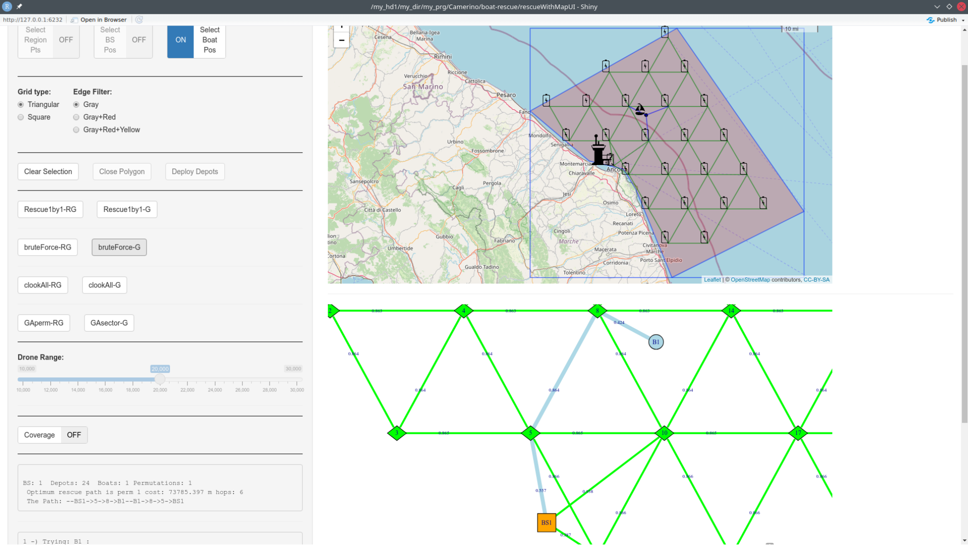
$= \binom{|(y_2-y_1)|+|(x_2-x_1)|}{|(x_2-x_1)|}$



Ref:2

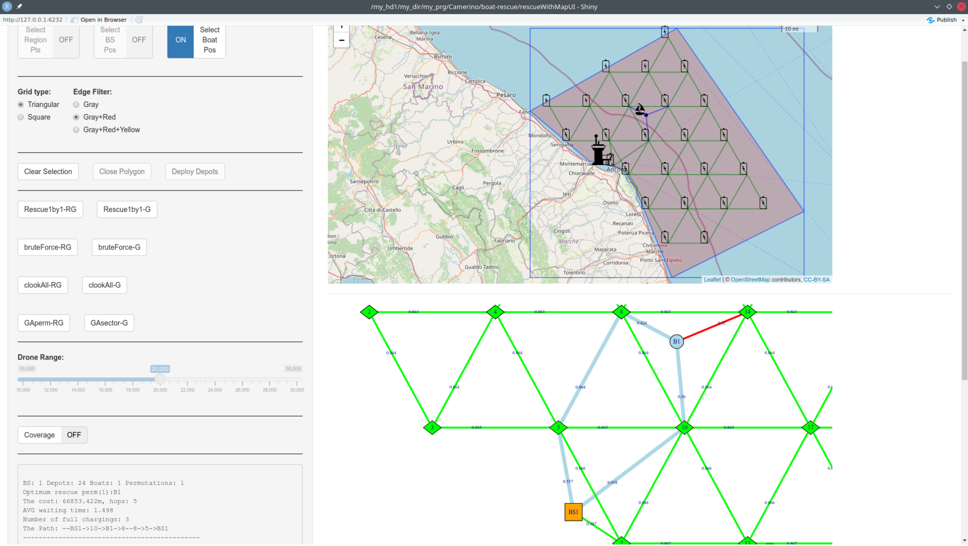

Select Region Pts
OFF
Select BS Pos
OFF
ON
Select Boat Pos
Reset
ClosePoly
TriGrid
SqGrid
Grid type:
Triangular
Square
Edge Filter:
Gray
Gray+Red
Gray+Red+Yellow
RND Boat Number:
11
1,000
RND Boats
Drone Range:
10,000
20,000
30,000
N Sim:
1
20
100
RunSim
concaveTSP-RG
concaveTSP-G
concaveTSP-RGY
10 km
5 mi
Senigallia
Falconara Marittima
Ancona
Chiaravalle
Jesi
Osimo
Castelfidardo
Loreto
Parco regionale del Conero
Leaflet | © OpenStreetMap contributors, CC-BY-SA
BS1
B4
B5
B6
B7
B8
B9

Ref:16



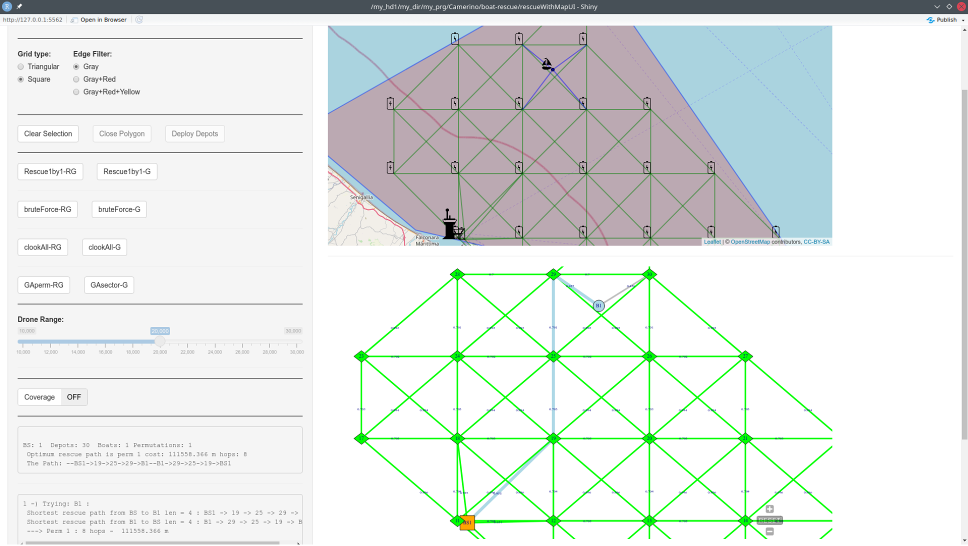
Single boat rescue **without** using Red-Gray heuristics. The rescue path is 73785.397m 6 hops 6 chargings.

Ref:16



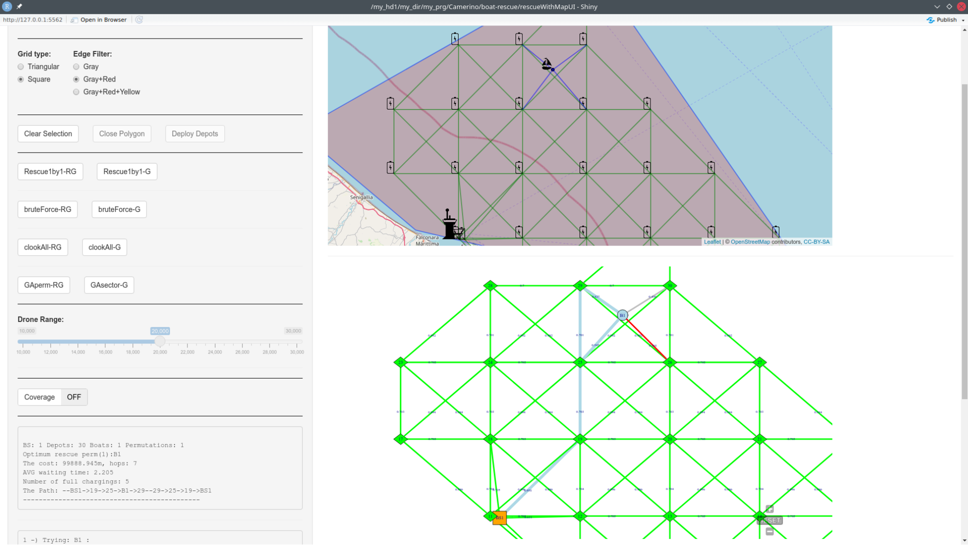
Single boat rescue **with** Red-Gray heuristics. The rescue path is 66853.422m (**9.4% shorter**) 5 hops 5 chargings (**less charging**).

Ref:16



Single boat rescue **without** using Red-Gray heuristics. The rescue path is 111558.366m 8 hops 8 chargings.

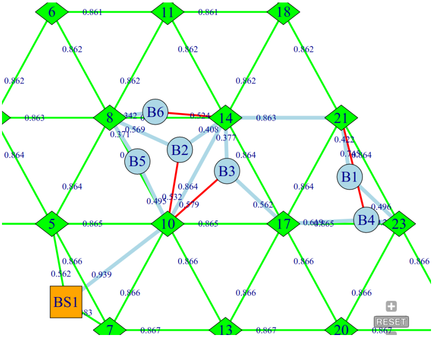
Ref:16

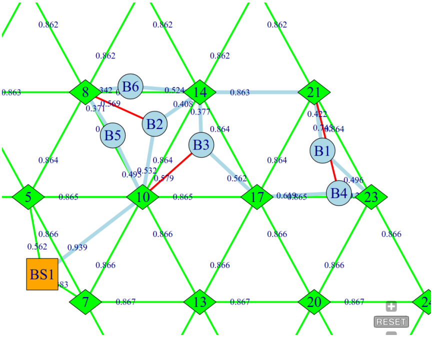


Single boat rescue **with** Red-Gray heuristics. The rescue path is 99888.945m (**10.5% shorter**) 7 hops 7 chargings (**less charging**). Sq has slight better savings! More simulations!

Ref:16

Multiple boats - Boat Rescue sim for 6 rnd boats:





Metric	redGraySP	BF RedGray
Rescue order	B5→B6→B2→B1→B4→B3	B5→B6→B3→B4→B1→B2
Tour cost	177086.641m, hops: 16	162708.808m, hops: 15
AWD	78524.963m	74979.52m
NChargings	9	8
Rescue Path	BS1→10→B5→8→B6→8→B2→14→21→B1→23→B4→17→B3→14→10→BS1	BS1→10→B5→8→B6→14→B3→17→B4→23→B1→21→14→B2→10→BS1

Ref:16

6 Boats Rescue Comparison (Single run!)

Table: Approximation Ratio (AR=$\frac{Other}{BFRedGray}$) over Brute-Force RedGray (Optimum) and Running Time improvements (Imp.=$\frac{BFRedGray}{Other}$).

Metric	redGraySP	GraySP	BF Gray	BF RedGray
Tour Cost(m) (AR)	177086.641 (1.09)	214516.268 (1.32)	197294.387 (1.21)	162708.808
AWD(m) (AR)	78524.963 (1.05)	91786.372 (1.22)	83264.743 (1.11)	74979.52
NChargings (AR)	9 (1.12)	12 (1.5)	11 (1.38)	8
Time(sec) (Imp.)	0.291 (56)	0.031 (530)	13.948 (1.18)	16.437

More simulations are necessary!

Ref:16

Custom TSP algo: concaveTSP (A24, A25)

Observations, design, and properties - I:

- Inspired by the observation that the **optimal TSP tours are non self-crossing, "collapsed" elastic band (x-hull?).**
- Experimented with convex and concave hull **(Topological and distance sorting)**. **Concave hull** gives better results.
- 2-phase algo:
  - **Global opt: Concentric concave hulls.** Uses **geometry** of vtx.
  - **Local opt: Merging** them in to single tour.

(Ref from: 31)

Ref:6

Custom TSP algo: concaveTSP (A24, A25)

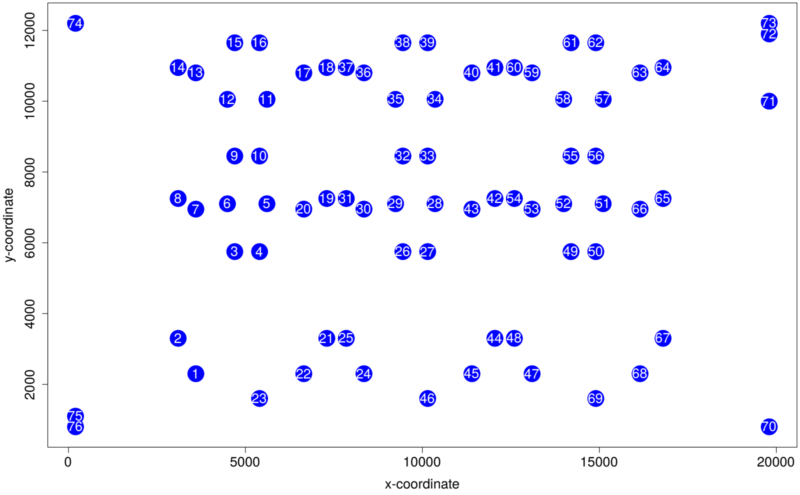
Observations, design, and properties - II:

- Use of concentric ConcaveHulls can **save from storing all the “Dist. MTX” (minHeap)**.
- Novel hybrid heuristics for merging: **Nearest vertex merging and online 3-edge path improvement heuristics (A14).**
- Novel and **fast** ($\mathcal{O}(N \log N)$) **approximation heuristic**: Suited to “rescue” operations.
- Gives **better results for regular “grid” like datasets**.

(Ref from: 31)

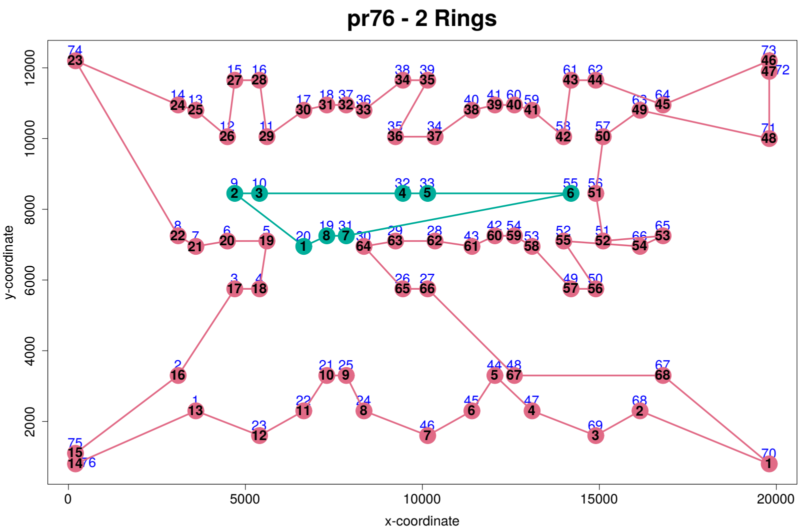
Ref:6

pr76 - 76 Vertices (cities)



(Ref from: 31)

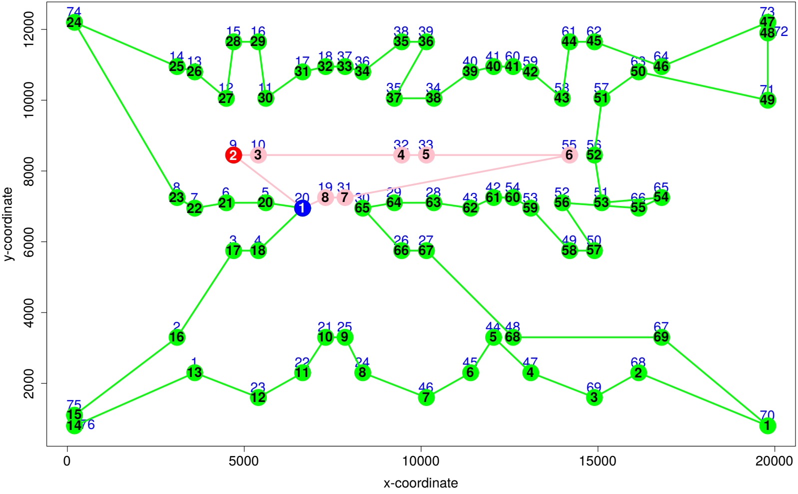
Ref:4



(Ref from: 31)

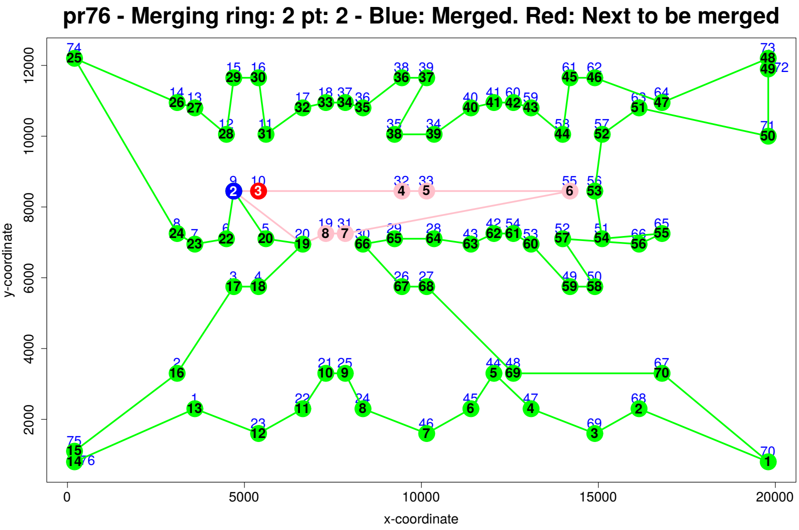
Ref:4

pr76 - Merging ring: 2 pt: 1 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

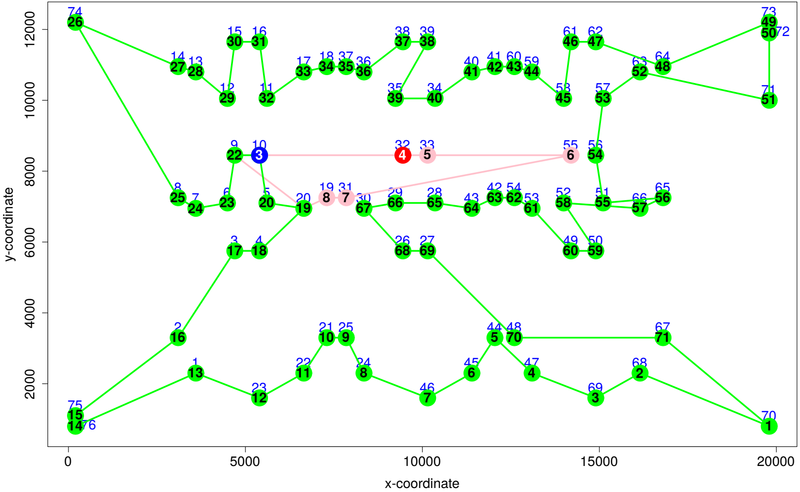
Ref:4



(Ref from: 31)

Ref:4

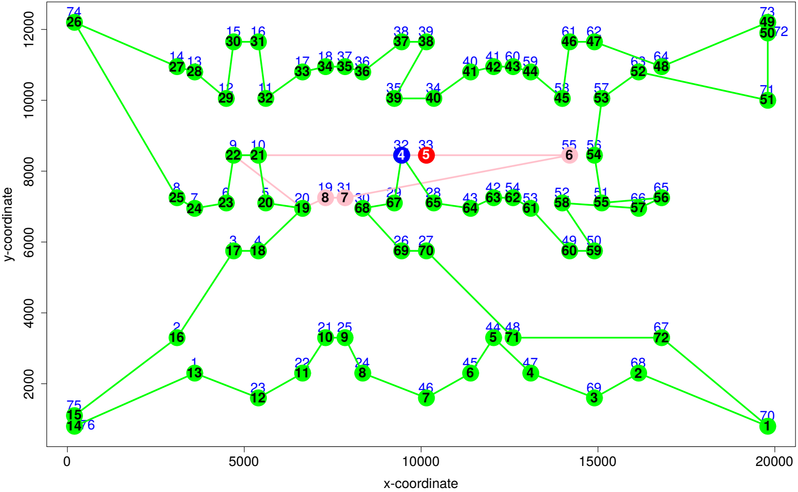
pr76 - Merging ring: 2 pt: 3 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

Ref:4

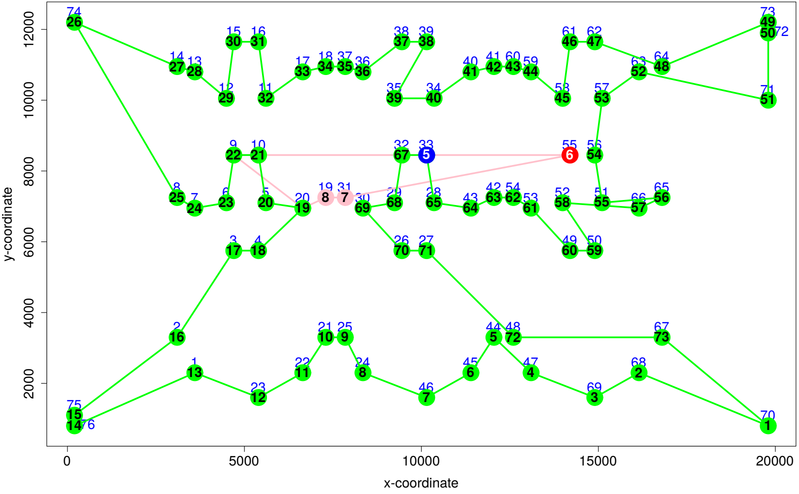
pr76 - Merging ring: 2 pt: 4 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

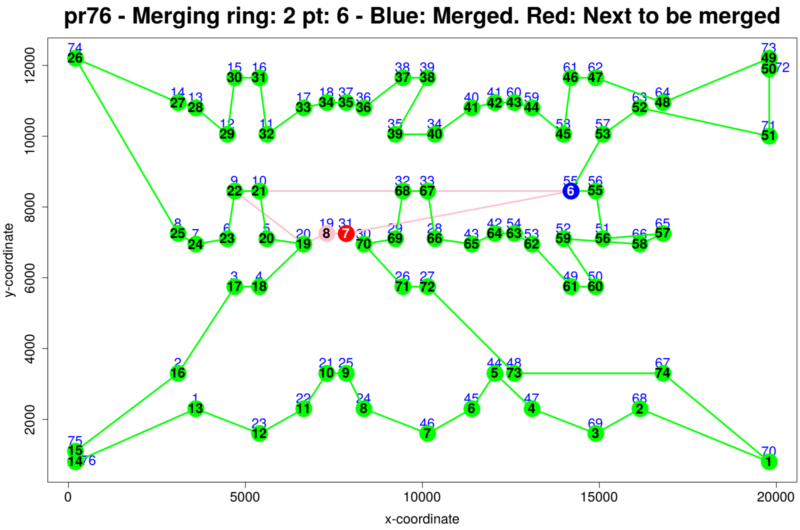
Ref:4

pr76 - Merging ring: 2 pt: 5 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

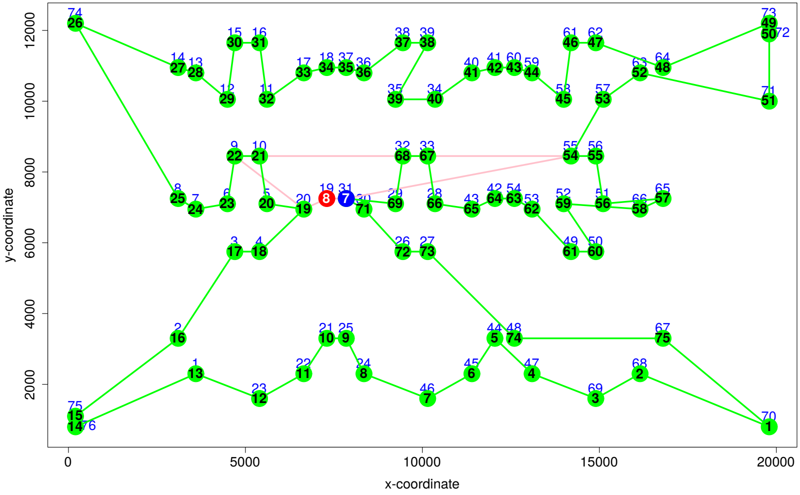
Ref:4



(Ref from: 31)

Ref:4

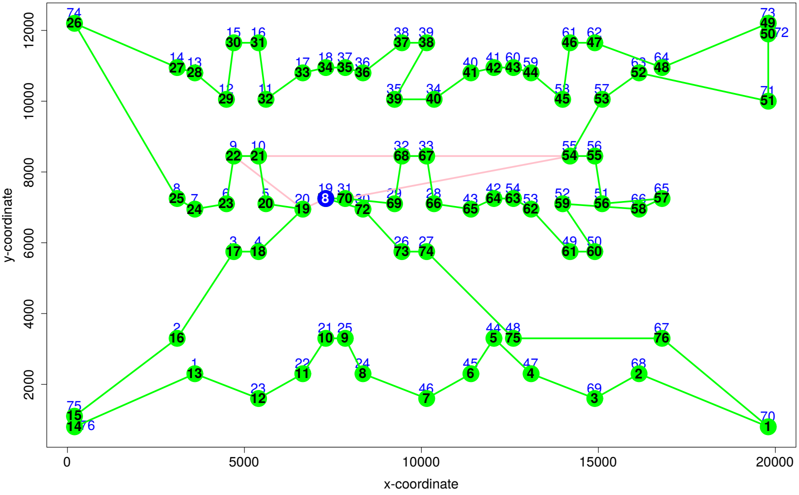
pr76 - Merging ring: 2 pt: 7 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

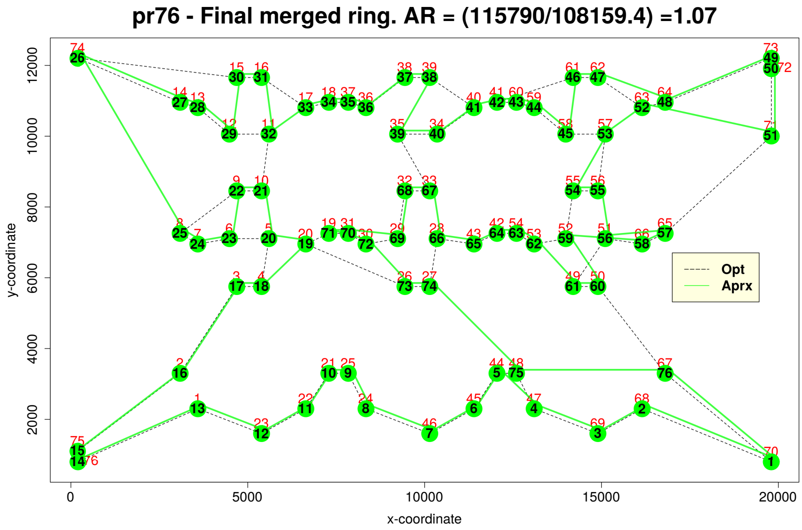
Ref:4

pr76 - Merging ring: 2 pt: 8 - Blue: Merged. Red: Next to be merged



(Ref from: 31)

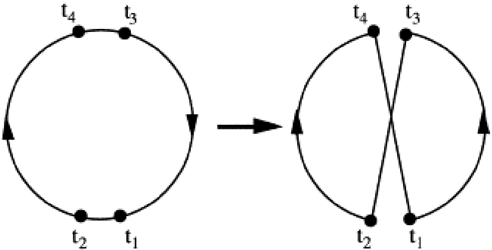
Ref:4



(Ref from: 31)

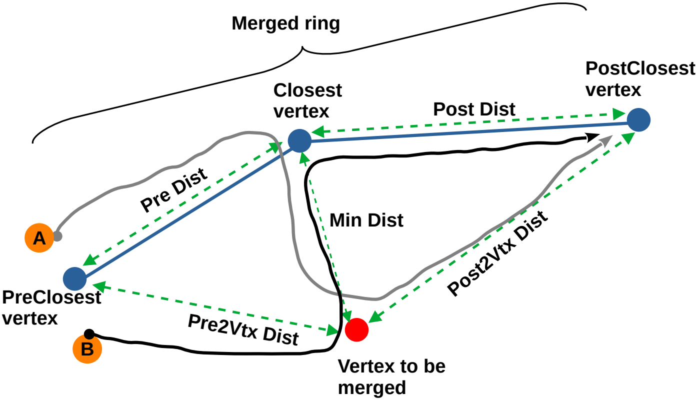
Ref:4

2-Opt: Example 2-edge exchange



Idea of the 2-Opt edge exchanges [10]. Ignore the geometric distances!

Online 3-edge heuristic



3-edge heuristic used during the merging phase of concaveTSP.

Ref:33

Rescue Drone

DJI MATRICE 300 RTK
15 km transmission range
82 kmph max speed
55 min max flight time
10000+ USD

Ref:5

Triangular vs Square: Edge statistics

Table: **Triangular grid**, 208449 (3 edges * 69483 points) **edges** are sampled.

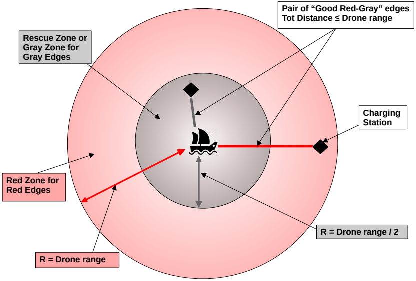
Edges	N	Prob.	Min	Max	Avg
gRG	94118	0.68	0.866	1	0.914
bRG	44838	0.32	1	1.253	1
All-RG	138956	1	0.866	1.253	0.968
gR	94118	0.45	0.502	0.866	0.652
bR	29599	0.14	0.502	0.779	0.677
All-R	123717	0.59	0.502	0.866	0.658
gG	54307	0.26	0	0.496	0.271
bG	30425	0.15	0.364	0.502	0.448
All-G	84732	0.41	0	0.502	0.334
All	208449	1.00	0	0.866	0.527

Table: **Square grid**, 643204 (4 edges * 160801 points) **edges** are sampled.

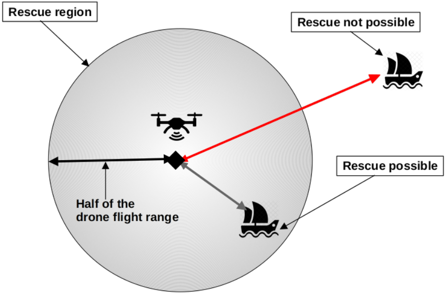
Edges	N	Prob.	Min	Max	Avg
gRG	199504	0.35	0.707	1	0.839
bRG	376064	0.65	1	1.366	1.075
All-RG	575568	1	0.707	1.366	0.993
gR	199504	0.45	0.502	1	0.608
bR	189448	0.14	0.502	0.999	0.749
All-R	388952	0.59	0.502	1	0.677
gG	131280	0.26	0	0.498	0.254
bG	122972	0.15	0.251	0.502	0.419
All-G	254252	0.41	0	0.502	0.334
All	643204	1.00	0	1	0.541

Ref:51

Red-Gray edges and Rescue cond.



Red-gray edges relative to the boat position.



Conditions for rescuing a boat.

Red-gray edges and rescue conditions.

▸ Ref:11

Rescue Framework Pseudocode

Algorithm The proposed rescue heuristic framework.

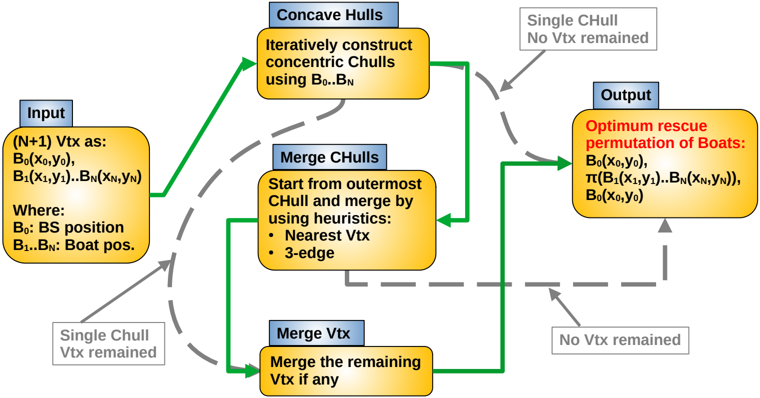
Input1: ▶ RescuePoly: User drawn polygon containing the rescue region.
Input2: ▶ GridType: User selected CS deployment configuration, tri or sq.
Input3: ▶ DroneRange: User selected drone range.
Input4: ▶ BS and BoatPos: User selected BS and Boat coordinates.
Output: ◀ ResTour: The optimum rescue tour: $(V_1 \ldots V_K)$

Grid $\leftarrow$ NULL
VtxList $\leftarrow$ NULL
BS and CS Deployment and Boat position selection
Grid $\leftarrow$ Grid + asVtx(userSelect(BS))
VtxList $\leftarrow$ VtxList + asVtx(userSelect(BS))
Grid $\leftarrow$ Grid + asGraph(deployCS(RescuePoly, userSelect(GridType), DroneRange))
Grid $\leftarrow$ Grid + asGraph(userSelect(BoatPos))
VtxList $\leftarrow$ VtxList + asVtx(userSelect(BoatPos))
Find the Best TSP tour for the BS + Boats for the drone
concaveTSPTour $\leftarrow$ concaveTSP(VtxList)
Find the Best Rescue Path for the drone
ResTour $\leftarrow$ redGraySP(concaveTSPTour, Grid)

Return(ResTour)

Ref:31

concaveTSP Algorithm Flowchart



(Ref from: 31)

Ref:32

concaveTSP Algorithm Pseudocode

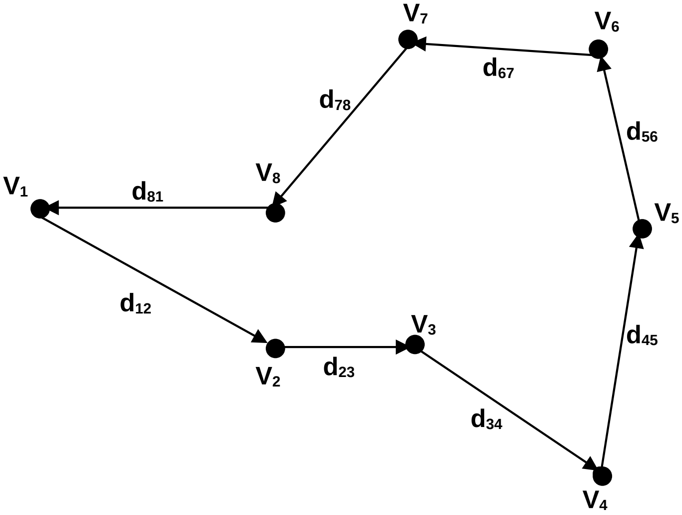
Algorithm The proposed TSP heuristic algorithm.

Input ▶ VtxList: Euclidean vertex coords with numbers, $(i, X_i, Y_i), i = 1 \dots N$
Output ◀ concaveTSPTour: The approximate TSP tour, $(V_1 \dots V_N)$

```
Concave hull construction phase
CHList ← NULL
VtxNotVisited ← VtxList
while (|VtxNotVisited| ≥ 2 ) do
    CH ← concave hull(VtxNotVisited)
    CHList ← CHList ∪ CH
    VtxNotVisited ← VtxNotVisited \ CH
end while
RemainedVtx ← VtxNotVisited

Merging phase: Select the nearest concaveTSPTour vtx and use 3-edge
NSubTour ← |CHList|
concaveTSPTour ← CHList[1]
if (NSubTour ≥ 2) then
    for each CH ∈ CHList[2..NSubTour] do
        NVtx ← |CH|
        for each Vtx ∈ CH[1..NVtx] do
            concaveTSPTour ← Merge(concaveTSPTour, Vtx)
        end for
    end for
end if

Merging RemainedVtx if any
if (|RemainedVtx| ≥ 2) then
    concaveTSPTour ← Merge(concaveTSPTour, RemainedVtx)
end if

Return(concaveTSPTour)
```

Ref:32

TSP tour



The TSP tour with 8 vertices, $\tau = (V_1, \ldots, V_8)$.

Ref:31

Metrics for TSP tour

$$\text{Tour Cost}(\tau) = \sum_{i=1}^{N-1} d_{(i,i+1)} + d_{(N,1)} \quad (1)$$

$$\text{cyclical - AWD}(\tau) = \frac{1}{N}(\sum_{j=2}^{N} \sum_{i=1}^{j-1} d_{(i,i+1)} + d_{(N,1)}) \quad (2)$$

$$\text{non-cyclical - AWD}(\tau) = \frac{1}{N-1}(\sum_{j=2}^{N} \sum_{i=1}^{j-1} d_{(i,i+1)}) \quad (3)$$

$$\text{min AWD}(\tau) = min(min(\rho^{F}_{i=1,\dots,N}(\tau)), min(\rho^{B}_{i=1,\dots,N}(\tau))) \quad (4)$$

$(d_{(i,j)} = Euclidean\ Distance(V_i, V_j))$

Ref:31

Rotations for TSP tour

For a TSP tour $\tau = (V_1, V_2, V_3, V_4, V_5, V_6, V_7)$ with $N = 7$:
Forward (CW):

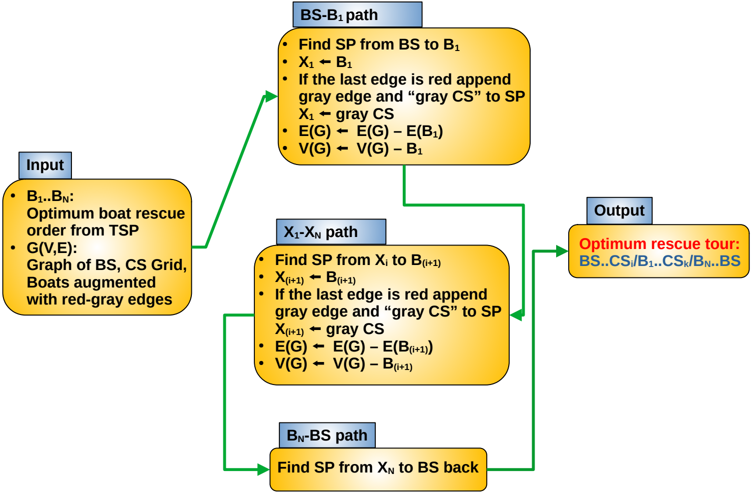
$$\rho_0^F(\tau) = (V_1, V_2, V_3, V_4, V_5, V_6, V_7)$$
$$\rho_1^F(\tau) = (V_2, V_3, V_4, V_5, V_6, V_7, V_1)$$
$$\rho_2^F(\tau) = (V_3, V_4, V_5, V_6, V_7, V_1, V_2)$$

Backward (Anti-CW):

$$\rho_0^B(\tau) = (V_1, V_7, V_6, V_5, V_4, V_3, V_2)$$
$$\rho_1^B(\tau) = (V_7, V_6, V_5, V_4, V_3, V_2, V_1)$$
$$\rho_2^B(\tau) = (V_6, V_5, V_4, V_3, V_2, V_1, V_7)$$

Ref:31

redGraySP Algorithm Flowchart



Ref:31

redGraySP Algorithm Pseudocode

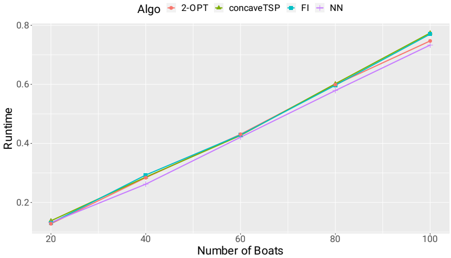
Algorithm The proposed red-gray shortest path algorithm.

Input1: ▶ BoatPerm: The approximate TSP tour of boats, $(B_1 \dots B_N)$
Input2: ▶ Grid: Graph of BS, CSs and boats, $G(V, E)$
Output: ◀ ResTour: The optimum rescue tour, $(V_1 \dots V_K)$

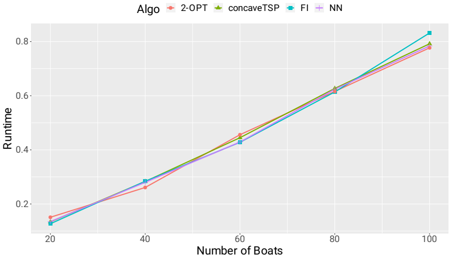
```
From BS to the first boat:
ResTour ← NULL
Dynamically augment Grid with the related boats
E(Grid) ← E(Grid) + redAndgrayEdges(B_1)
V(Grid) ← V(Grid) + asVtx(B_1)
SP ← shortestPath(Grid, src=BS, dst=B_1)
if colour(lastEdge(SP)) == "red") then
    appendEdge(SP) ← E(findAdjacentGoodGrayCS(B_1))
    appendVtx(SP) ← V(findAdjacentGoodGrayCS(B_1))
end if
ResTour ← ResTour + SP
BoatPerm ← BoatPerm - B_1
Dynamically de-augment Grid with the related boats
This is to force the rescue order as in the BoatPerm
E(Grid) ← E(Grid) - redAndgrayEdges(B_1)
V(Grid) ← V(Grid) - asVtx(B_1)
From the first boat to the last boat:
if lastVtx(ResTour)) == BS) then
    Return(ResTour)
else
    prevBoat ← B_1
    while (BoatPerm ≠ NULL) do
        nextBoat ← getNext(BoatPerm)
Dynamically augment Grid with the related boats
        E(Grid) ← E(Grid) + redAndgrayEdges(prevBoat)
        V(Grid) ← V(Grid) + asVtx(prevBoat)
        E(Grid) ← E(Grid) + redAndgrayEdges(nextBoat)
        V(Grid) ← V(Grid) + asVtx(nextBoat)
        SP ← shortestPath(Grid, src=lastVtx(ResTour),
dst=nextBoat)
        if colour(lastEdge(SP)) == "red") then
            appendEdge(SP) ← E(findAdjacentGoodGrayCS(nextBoat))
            appendVtx(SP) ← V(findAdjacentGoodGrayCS(nextBoat))
        end if
        ResTour ← ResTour + SP
        BoatPerm ← BoatPerm - nextBoat
Dynamically de-augment Grid with the related boats
        E(Grid) ← E(Grid) - redAndgrayEdges(prevBoat)
        V(Grid) ← V(Grid) - asVtx(prevBoat)
        E(Grid) ← E(Grid) - redAndgrayEdges(nextBoat)
        V(Grid) ← V(Grid) - asVtx(nextBoat)
        prevBoat ← nextBoat
    end while
end if
From the last boat to the BS:
if lastVtx(ResTour)) == BS) then
    Return(ResTour)
else
    SP ← shortestPath(Grid, src=lastVtx(ResTour), dst=BS)
    ResTour ← ResTour + SP
end if
Return(ResTour)
```

▶ Ref:31

Pathfinding Benchmarks: AVG Runtime



AVG Runtimes for triangular grid.

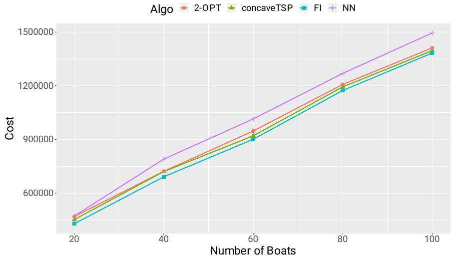


AVG Runtimes for square grid.

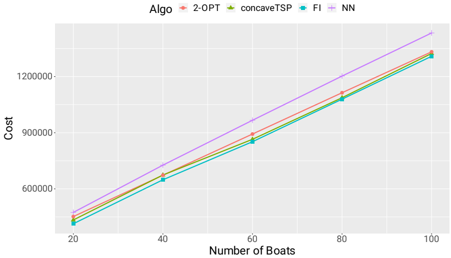
AVG Runtimes in seconds from 20 sims.

Ref:18

Pathfinding Benchmarks: AVG Tour Cost



AVG Tour Cost for triangular grid.

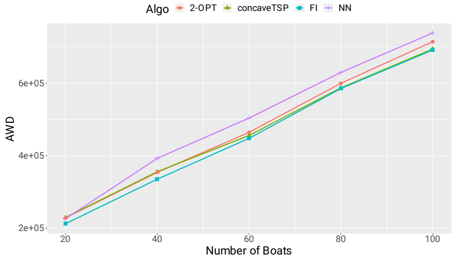


AVG Tour Cost for square grid.

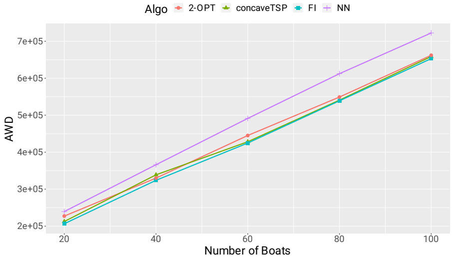
AVG Tour Cost in meters from 20 sims.

Ref:18

Pathfinding Benchmarks: AVG AWD



AVG AWD for triangular grid.

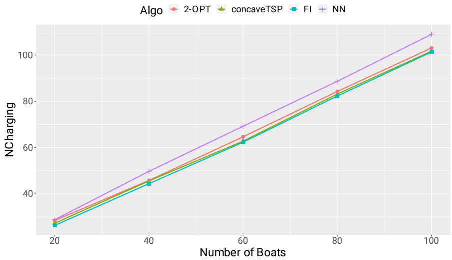


AVG AWD for square grid.

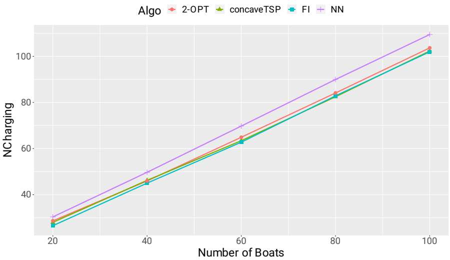
AVG AWD in meters from 20 sims.

Ref:18

Pathfinding Benchmarks: AVG NChargings



AVG Number of Chargings for triangular grid.



AVG Number of Chargings for square grid.

AVG Number of Chargings from 20 sims.

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. **20 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Metric	concaveTSP	FI	NN	2-OPT
Tri	RG Cost	452221.4 ± 45194.8	428986 ± 44761.8	472358.8 ± 44207.8	469513.4 ± 54352.4
	G Cost	548381.874 ± 51218.2	524330.3 ± 42431.6	578961.3 ± 56157.4	556431.1 ± 42991.8
	Cost Saving %	17.5 ± 3.5	18.2 ± 4.6	18 ± 8.3	15.5 ± 8.2
	RG Time	0.138 ± 0.016	0.13 ± 0.017	0.132 ± 0.021	0.128 ± 0.018
	G Time	0.115 ± 0.017	0.111 ± 0.015	0.106 ± 0.014	0.115 ± 0.025
	Time Saving %	-22.5 ± 22.6	-19 ± 24.7	-25.5 ± 25.8	-15.7 ± 26.0
	RG AWD	229580.6 ± 29556.8	212353.9 ± 27475.4	226095.7 ± 27345.6	228387.2 ± 33754.3
	G AWD	260659.2 ± 30823.6	250167.3 ± 25706.1	268733.7 ± 27410.8	264514.2 ± 24029.2
	AWD Saving %	11.4 ± 10.6	15.1 ± 7.6	15.2 ± 12.2	13.4 ± 11.6
	RG NCharging	27.3 ± 2.6	26.4 ± 2.7	28.8 ± 2.5	28.5 ± 3.2
	G NCharging	37 ± 3.0	35.6 ± 2.5	38.8 ± 3.2	37.4 ± 2.6
	NCharging Saving %	26.1 ± 4.6	26.0 ± 5.4	25.3 ± 7.4	23.7 ± 7.5
Sq	RG Cost	433795.7 ± 41318.6	413666.8 ± 39358.3	474235.7 ± 46487.9	452063.7 ± 42498.3
	G Cost	484250.594 ± 45610.3	463520.3 ± 45947.6	503022.3 ± 56043.2	487495.8 ± 44167.4
	Cost Saving %	10.4 ± 2.8	10.6 ± 5.4	5.4 ± 6.7	7.1 ± 5.6
	RG Time	0.134 ± 0.015	0.127 ± 0.012	0.135 ± 0.019	0.151 ± 0.082
	G Time	0.124 ± 0.017	0.12 ± 0.021	0.113 ± 0.015	0.109 ± 0.011
	Time Saving %	-10.5 ± 21.0	-8.0 ± 20.8	-21.2 ± 19.3	-40.0 ± 80.1
	RG AWD	212517.9 ± 24600.3	206507.0 ± 27921.9	239400.5 ± 29827.2	227244.7 ± 31700.8
	G AWD	227655.3 ± 23861.4	219690.2 ± 26102.8	232338 ± 25767.3	228410.8 ± 24335.6
	AWD Saving %	6.6 ± 6.8	5.9 ± 7.094	-4.1 ± 16.6	0.5 ± 9.6
	RG NCharging	28 ± 2.5	26.6 ± 1.9	30.4 ± 2.2	28.6 ± 2.6
	G NCharging	35.2 ± 3.1	33.8 ± 3.2	36.2 ± 3.6	35.2 ± 2.7
	NCharging Saving %	20.2 ± 3.7	21.2 ± 5.4	15.7 ± 5.3	18.7 ± 5.0

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. **40 boats randomly generated for 20 simulations**. Results are in the form of mean $\pm$ standard deviation.

Grid	Metric	concaveTSP	FI	NN	2-OPT
Tri	RG Cost	720751.6 $\pm$ 45793.7	691249.3 $\pm$ 42099.4	790136.0 $\pm$ 59432.8	721864.5 $\pm$ 50161.6
	G Cost	872883.2 $\pm$ 62339.7	840797.8 $\pm$ 66965.2	932849.2 $\pm$ 80359.5	885781.1 $\pm$ 57051.8
	Cost Saving %	17.4 $\pm$ 2.3	17.6 $\pm$ 3.1	15.0 $\pm$ 6.6	18.4 $\pm$ 4.4
	RG Time	0.286 $\pm$ 0.027	0.293 $\pm$ 0.066	0.262 $\pm$ 0.027	0.284 $\pm$ 0.082
	G Time	0.233 $\pm$ 0.014	0.236 $\pm$ 0.024	0.24 $\pm$ 0.023	0.232 $\pm$ 0.021
	Time Saving %	-23.7 $\pm$ 16.6	-26.1 $\pm$ 36.2	-10.7 $\pm$ 20.8	-24.8 $\pm$ 43.4
	RG AWD	355812.2 $\pm$ 28430.5	335152.0 $\pm$ 18936.2	392548.8 $\pm$ 52073.8	353986.5 $\pm$ 25156.5
	G AWD	423383.2 $\pm$ 33027.5	409855.1 $\pm$ 32340.7	441702.0 $\pm$ 37025.5	426331.5 $\pm$ 28120.4
	AWD Saving %	15.8 $\pm$ 5	18.0 $\pm$ 4.9	11.1 $\pm$ 9.5	16.9 $\pm$ 5.0
	RG NCharging	45.6 $\pm$ 2.5	44.4 $\pm$ 2.3	49.8 $\pm$ 3.1	45.8 $\pm$ 2.0
	G NCharging	60.5 $\pm$ 3.8	58.8 $\pm$ 3.0	64.0 $\pm$ 4.5	61.4 $\pm$ 2.9
	NCharging Saving %	24.5 $\pm$ 3.2	24.4 $\pm$ 3.4	21.9 $\pm$ 6.8	25.3 $\pm$ 4.1
Sq	RG Cost	674039.5 $\pm$ 44620.1	648563.9 $\pm$ 47506.8	726830.7 $\pm$ 62542.1	672992.2 $\pm$ 41647.4
	G Cost	752414.9 $\pm$ 45426.4	723336.2 $\pm$ 41700.1	806233.0 $\pm$ 65459.6	760135.2 $\pm$ 38325.9
	Cost Saving %	10.4 $\pm$ 3.0	10.3 $\pm$ 4.2	9.7 $\pm$ 5.6	11.3 $\pm$ 6.2
	RG Time	0.283 $\pm$ 0.025	0.284 $\pm$ 0.076	0.28 $\pm$ 0.021	0.261 $\pm$ 0.016
	G Time	0.271 $\pm$ 0.083	0.244 $\pm$ 0.022	0.255 $\pm$ 0.08	0.248 $\pm$ 0.021
	Time Saving %	-9.4 $\pm$ 21.3	-17.8 $\pm$ 37.9	-15.4 $\pm$ 22.1	-6.2 $\pm$ 13.5
	RG AWD	338778.0 $\pm$ 25650.2	323931.2 $\pm$ 26247.0	366110.5 $\pm$ 44705.3	329932.0 $\pm$ 31020.0
	G AWD	364926.3 $\pm$ 24261.6	347338.3 $\pm$ 23588.8	385326.7 $\pm$ 37316.1	366898.4 $\pm$ 18717.9
	AWD Saving %	7.0 $\pm$ 5.8	6.6 $\pm$ 7.5	4.6 $\pm$ 11.7	9.9 $\pm$ 9.6
	RG NCharging	46.2 $\pm$ 1.9	45 $\pm$ 2.5	49.7 $\pm$ 3.2	46.0 $\pm$ 2.1
	G NCharging	57.7 $\pm$ 3.1	55.8 $\pm$ 2.8	60.6 $\pm$ 3.4	57.8 $\pm$ 2.8
	NCharging Saving %	19.8 $\pm$ 4.5	19.3 $\pm$ 4.9	17.9 $\pm$ 5.1	20.4 $\pm$ 5.8

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. **60 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Metric	concaveTSP	FI	NN	2-OPT
Tri	RG Cost	920434.6 ± 32296.1	901084.1 ± 39776.9	1014695.3 ± 48015.6	947449.5 ± 47647.7
	G Cost	1122678.6 ± 54088.8	1109086.4 ± 58758.0	1223358.7 ± 73658.9	1148255.4 ± 47218.4
	Cost Saving %	17.9 ± 2.3	18.7 ± 3.2	16.9 ± 4.8	17.4 ± 3.9
	RG Time	0.427 ± 0.02	0.429 ± 0.012	0.421 ± 0.019	0.431 ± 0.074
	G Time	0.4 ± 0.07	0.382 ± 0.025	0.374 ± 0.023	0.388 ± 0.068
	Time Saving %	-8.5 ± 11.8	-12.5 ± 6.5	-12.8 ± 6.8	-13.2 ± 23.4
	RG AWD	455620.9 ± 23194.3	447833.4 ± 25049.2	503701.6 ± 46465.3	464278.8 ± 31067.5
	G AWD	547482.7 ± 32740.8	539610.8 ± 30919.0	583976.3 ± 40899.3	548870.2 ± 33839.5
	AWD Saving %	16.7 ± 3.7	16.9 ± 4.8	13.3 ± 10.5	15.3 ± 5.5
	RG NCharging	62.8 ± 2.1	62.4 ± 1.8	69.2 ± 2.4	64.7 ± 2.9
	G NCharging	81.4 ± 3.2	80.6 ± 3.6	87.3 ± 4.4	82.8 ± 3.0
	NCharging Saving %	22.8 ± 3.2	22.5 ± 3.9	20.5 ± 3.6	21.8 ± 3.1
Sq	RG Cost	864874.0 ± 41174.3	852019.4 ± 39644.0	966629.2 ± 54690.3	892968.7 ± 57206.6
	G Cost	977309.5 ± 52123.8	953718.8 ± 58872.9	1052405.6 ± 61678.2	988270.3 ± 61886.8
	Cost Saving %	11.4 ± 2.4	10.5 ± 4.1	8.0 ± 5.6	9.5 ± 5.4
	RG Time	0.445 ± 0.024	0.428 ± 0.019	0.429 ± 0.021	0.456 ± 0.097
	G Time	0.427 ± 0.064	0.395 ± 0.02	0.403 ± 0.077	0.392 ± 0.022
	Time Saving %	-5.5 ± 10.2	-8.6 ± 6.2	-8.5 ± 12.6	-17.0 ± 27.1
	RG AWD	428230.9 ± 25205.0	424685.0 ± 25096.3	491356.2 ± 34682.3	445149.7 ± 30371.0
	G AWD	470800.3 ± 28575.7	463554.5 ± 31222.1	501983.1 ± 38491.1	473394.2 ± 32971.8
	AWD Saving %	8.9 ± 4.5	8.1 ± 6.3	1.8 ± 7.8	5.7 ± 7.4
	RG NCharging	63.4 ± 1.7	62.8 ± 1.7	69.8 ± 3.2	64.9 ± 2.7
	G NCharging	78 ± 2.5	75.9 ± 3.2	81.8 ± 3.8	78.0 ± 3.2
	NCharging Saving %	18.6 ± 3.5	17.2 ± 3.8	14.6 ± 4.2	16.6 ± 4.0

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. **80 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Metric	concaveTSP	FI	NN	2-OPT
Tri	RG Cost	1193873.4 ± 72509.5	1173711.2 ± 65385.8	1269457.7 ± 91281.2	1207666.7 ± 74836.9
	G Cost	1435808.0 ± 76420.3	1406063.3 ± 103009.1	1521348.4 ± 90490.4	1452633.4 ± 85168.4
	Cost Saving %	16.8 ± 2.3	16.4 ± 3.8	16.5 ± 3.6	16.8 ± 3.4
	RG Time	0.602 ± 0.024	0.597 ± 0.059	0.579 ± 0.016	0.599 ± 0.057
	G Time	0.545 ± 0.012	0.54 ± 0.021	0.546 ± 0.045	0.531 ± 0.013
	Time Saving %	-10.5 ± 4.1	-10.5 ± 11.4	-6.6 ± 7.6	-13.0 ± 11.4
	RG AWD	586966.9 ± 29048.0	584999.6 ± 29249.8	629485.6 ± 49728.9	599470.5 ± 41282.2
	G AWD	701345.6 ± 36936.2	684808.4 ± 53810.1	736745.9 ± 42902.6	707968.4 ± 43998.6
	AWD Saving %	16.4 ± 2.6	14.3 ± 5.4	14.4 ± 7.4	15.3 ± 3.3
	RG NCharging	83.2 ± 2.4	82.3 ± 1.7	88.7 ± 3.3	84.3 ± 2.0
	G NCharging	104.5 ± 3.7	103.0 ± 3.9	109.5 ± 4.0	105.8 ± 3.8
	NCharging Saving %	20.4 ± 2.4	20.0 ± 3.0	18.9 ± 3.1	20.2 ± 3.0
Sq	RG Cost	1087114.7 ± 41665.8	1079338.9 ± 41120.3	1202759.3 ± 62191.1	1113741.6 ± 53598.5
	G Cost	1204612.8 ± 53285.4	1202933.5 ± 45489.3	1294414.3 ± 63389.2	1251674.3 ± 75083.5
	Cost Saving %	9.7 ± 2.7	10.2 ± 2.9	6.7 ± 5.2	10.8 ± 5.0
	RG Time	0.627 ± 0.032	0.614 ± 0.02	0.624 ± 0.06	0.616 ± 0.021
	G Time	0.575 ± 0.021	0.579 ± 0.061	0.596 ± 0.077	0.563 ± 0.019
	Time Saving %	-9.1 ± 6.6	-6.8 ± 8.6	-6.2 ± 16.3	-9.6 ± 4.7
	RG AWD	540783.1 ± 27959.0	538915.9 ± 28640.4	612312.0± 46168.5	549134.7 ± 36898.8
	G AWD	589200.9 ± 28054.9	584540.6 ± 21146.8	616422.4 ± 34841.8	607477.8 ± 34821.5
	AWD Saving %	8.2 ± 3.9	7.8 ± 4.2	0.5 ± 7.8	9.4 ± 7.2
	RG NCharging	82.4 ± 2.4	82.8 ± 2.1	90.0 ± 2.9	84.1 ± 2.8
	G NCharging	97.3 ± 4.0	97 ± 4.0	102.7 ± 4.0	100.4 ± 4.9
	NCharging Saving %	15.2 ± 3.6	14.5 ± 3.4	12.3 ± 4.1	16.1 ± 4.2

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. **100 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Metric	concaveTSP	FI	NN	2-OPT
Tri	RG Cost	1396244.1 ± 59737.2	1382759.1 ± 49286.5	1495121.1 ± 58477.8	1411860.7 ± 60752.0
	G Cost	1661578.8 ± 85371.8	1623590.9 ± 88023.8	1740390.5 ± 108055.0	1682288.9 ± 60739.8
	Cost Saving %	15.9 ± 2.2	14.7 ± 3.4	13.9 ± 4.1	16.0 ± 3.0
	RG Time	0.774 ± 0.071	0.77 ± 0.08	0.733 ± 0.014	0.747 ± 0.02
	G Time	0.721 ± 0.078	0.698 ± 0.012	0.693 ± 0.012	0.696 ± 0.011
	Time Saving %	-8.4 ± 14.7	-10.4 ± 13.0	-5.8 ± 2.3	-7.5 ± 3.2
	RG AWD	694099.1 ± 30895.1	691196.4 ± 26685.4	738222.4 ± 48601.1	714325.9 ± 42378.5
	G AWD	813660.4 ± 42993.1	792946.9 ± 47939.9	827134.7 ± 46929.7	821965.4 ± 33998.7
	AWD Saving %	14.6 ± 3.1	12.6 ± 4.5	10.5 ± 7.7	13.1 ± 4.2
	RG NCharging	101.7 ± 2.2	101.4 ± 2.0	109 ± 2.5	103.1 ± 2.6
	G NCharging	125.2 ± 4.008	1 ± 4.2	129.6 ± 4.8	126.2 ± 3.2
	NCharging Saving %	18.7 ± 2.8	17.5 ± 2.8	15.8 ± 3.3	18.3 ± 2.2
Sq	RG Cost	1323625.9 ± 47004.3	1308544.6 ± 36527.9	1433151.1 ± 72536.7	1332355.1 ± 49288.4
	G Cost	1453297.1 ± 66877.8	1436964.1 ± 63498.9	1545871.9 ± 68780.6	1475904.1 ± 47047.5
	Cost Saving %	8.9 ± 2.2	8.8 ± 3.3	7.2 ± 3.6	9.7 ± 3.5
	RG Time	0.792 ± 0.029	0.832 ± 0.13	0.784 ± 0.034	0.777 ± 0.023
	G Time	0.758 ± 0.025	0.741 ± 0.032	0.742 ± 0.028	0.74 ± 0.033
	Time Saving %	-4.6 ± 3.8	-12.6 ± 19.2	-5.8 ± 5.4	-5.1 ± 4.7
	RG AWD	658603.1 ± 24126.3	653044.7 ± 25061.3	722387.9 ± 51718.8	662079.8 ± 38094.0
	G AWD	712229.3 ± 35801.1	702810.9 ± 31125.1	738183.7 ± 31901.1	717612.3 ± 26269.5
	AWD Saving %	7.4 ± 3.2	7.0 ± 3.1	2.1 ± 6.9	7.7 ± 5.4
	RG NCharging	102.3 ± 1.5	102.0 ± 1.4	109.4 ± 3.6	103.6 ± 2.1
	G NCharging	118.1 ± 4.4	116.9 ± 4.4	123.0 ± 4.3	119.2 ± 3.6
	NCharging Saving %	13.3 ± 3.1	12.7 ± 2.8	10.9 ± 3.2	13.0 ± 2.8

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. Approximation ratios and Speed-ups are over Brute-Force method. **4 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Algo	Cost Saving %	AWD Saving %	Num Chargings Saving %	Approx. Ratio RG	Approx. Ratio G	Speed-up Ratio RG	Speed-up Ratio G
Tri	BruteForce	9.141 ± 4.824	6.126 ± 13.672	17.336 ± 9.627	NA	NA	NA	NA
	clook	9.258 ± 4.411	5.106 ± 11.79	17.182 ± 9.052	1.016 ± 0.049	1.016 ± 0.037	11.385 ± 1.868	10.951 ± 2.031
	concaveTSP	9.543 ± 4.491	5.924 ± 12.768	17.67 ± 9.096	1.001 ± 0.002	1.005 ± 0.016	11.895 ± 2.842	11.169 ± 1.923
	nearest_insertion	9.237 ± 4.474	3.172 ± 12.4	17.285 ± 9.356	1.006 ± 0.016	1.007 ± 0.019	13.732 ± 3.014	12.314 ± 3.009
	farthest_insertion	9.543 ± 4.491	5.788 ± 12.714	17.67 ± 9.096	1.001 ± 0.002	1.005 ± 0.016	12.075 ± 3.237	11.692 ± 3.627
	cheapest_insertion	9.237 ± 4.474	3.42 ± 12.533	17.285 ± 9.356	1.006 ± 0.016	1.007 ± 0.019	13.607 ± 3.244	12.904 ± 2.361
	arbitrary_insertion	9.543 ± 4.491	5.788 ± 12.714	17.67 ± 9.096	1.001 ± 0.002	1.005 ± 0.016	12.148 ± 4.512	13.176 ± 1.574
	nn	7.88 ± 9.473	0.581 ± 19.279	15.729 ± 12.721	1.037 ± 0.057	1.026 ± 0.048	13.935 ± 2.958	13.012 ± 2.356
	repetitive_nn	9.543 ± 4.491	5.788 ± 12.714	17.67 ± 9.096	1.001 ± 0.002	1.005 ± 0.016	12.213 ± 3.011	12.124 ± 1.513
	two_opt	5.999 ± 8.718	4.677 ± 15.799	13.952 ± 11.976	1.052 ± 0.064	1.018 ± 0.034	13.96 ± 3.093	12.98 ± 2.369
Sq	BruteForce	6.64 ± 4.526	-0.405 ± 11.464	15.473 ± 8.785	NA	NA	NA	NA
	clook	6.51 ± 4.464	7.566 ± 9.3	15.152 ± 8.676	1.022 ± 0.056	1.02 ± 0.051	10.867 ± 2.023	11.82 ± 3.386
	concaveTSP	6.378 ± 4.574	-1.617 ± 22.252	14.731 ± 8.471	1.005 ± 0.011	1.002 ± 0.008	11.284 ± 1.517	12.135 ± 3.508
	nearest_insertion	6.07 ± 4.571	0.483 ± 13.622	14.346 ± 8.244	1.01 ± 0.023	1.004 ± 0.012	11.415 ± 3.428	13.692 ± 3.853
	farthest_insertion	6.372 ± 4.572	-0.985 ± 22.393	14.731 ± 8.471	1.005 ± 0.011	1.002 ± 0.008	12.282 ± 2.331	12.712 ± 5.212
	cheapest_insertion	6.219 ± 4.595	2.482 ± 10.21	14.731 ± 8.471	1.009 ± 0.023	1.004 ± 0.012	12.648 ± 1.819	13.43 ± 4.728
	arbitrary_insertion	6.358 ± 4.563	1.56 ± 14.266	14.731 ± 8.471	1.005 ± 0.011	1.002 ± 0.008	12.391 ± 2.316	14.109 ± 4.377
	nn	7.056 ± 6.867	1.005 ± 18.681	14.797 ± 9.229	1.024 ± 0.051	1.03 ± 0.058	12.082 ± 3.036	14.026 ± 4.182
	repetitive_nn	6.372 ± 4.572	-0.985 ± 22.393	14.731 ± 8.471	1.005 ± 0.011	1.002 ± 0.008	11.894 ± 1.724	12.624 ± 4.071
	two_opt	5.319 ± 7.807	-3.218 ± 26.245	13.235 ± 10.656	1.035 ± 0.069	1.021 ± 0.038	11.996 ± 2.964	13.04 ± 5.025

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. Approximation ratios and Speed-ups are over Brute-Force method. **5 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Algo	Cost Saving %	AWD Saving %	Num Chargings Saving %	Approx. Ratio RG	Approx. Ratio G	Speed-up Ratio RG	Speed-up Ratio G
Tri	BruteForce	8.868 ± 4.655	1.666 ± 21.291	17.29 ± 6.963	NA	NA	NA	NA
	clook	8.323 ± 4.433	3.41 ± 15.792	16.38 ± 6.543	1.047 ± 0.055	1.041 ± 0.055	54.702 ± 10.672	53.934 ± 10.525
	concaveTSP	8.528 ± 4.594	-1.841 ± 20.413	16.761 ± 6.683	1.026 ± 0.051	1.022 ± 0.045	53.547 ± 14.533	54.361 ± 10.974
	nearest_insertion	8.748 ± 5.491	1.134 ± 18.158	17.304 ± 8.558	1.026 ± 0.043	1.025 ± 0.036	61.307 ± 14.513	61.993 ± 14.073
	farthest_insertion	8.866 ± 5.306	3.989 ± 16.013	17.482 ± 7.09	1.014 ± 0.035	1.015 ± 0.035	65.928 ± 9.297	66 ± 5.592
	cheapest_insertion	8.091 ± 5.301	1.169 ± 15.679	16.756 ± 7.5	1.03 ± 0.044	1.021 ± 0.038	64.686 ± 10.608	58.243 ± 17.337
	arbitrary_insertion	9.193 ± 5.041	4.311 ± 14.599	17.966 ± 7.083	1.017 ± 0.036	1.021 ± 0.039	63.862 ± 10.672	64.673 ± 9.737
	nn	5.798 ± 6.573	-2.258 ± 21.899	13.398 ± 8.536	1.077 ± 0.091	1.042 ± 0.072	66.167 ± 11.465	61.269 ± 15.148
	repetitive_nn	8.482 ± 4.339	-0.964 ± 20.983	16.875 ± 6.748	1.019 ± 0.038	1.015 ± 0.035	54.704 ± 17.652	58.802 ± 5.497
	two_opt	7.574 ± 8.497	2.316 ± 15.387	15.708 ± 8.342	1.056 ± 0.087	1.042 ± 0.051	65.192 ± 7.043	63.589 ± 10.872
Sq	BruteForce	5.934 ± 2.907	-6.853 ± 20.773	15.033 ± 5.709	NA	NA	NA	NA
	clook	5.84 ± 3.163	-10.614 ± 23.46	14.197 ± 6.288	1.037 ± 0.039	1.036 ± 0.041	56.284 ± 5.724	46.617 ± 16.545
	concaveTSP	5.983 ± 2.491	-13.563 ± 24.415	13.953 ± 5.24	1.012 ± 0.031	1.013 ± 0.03	56.867 ± 9.547	50.568 ± 15.28
	nearest_insertion	5.748 ± 2.515	-8.725 ± 23.09	13.898 ± 5.153	1.02 ± 0.029	1.018 ± 0.036	61.962 ± 12.212	63.733 ± 8.442
	farthest_insertion	6.612 ± 3.038	-10.137 ± 24.24	15.208 ± 5.761	1.003 ± 0.008	1.011 ± 0.022	62.768 ± 10.748	61.684 ± 12.187
	cheapest_insertion	6.249 ± 3.019	-9.493 ± 22.342	14.884 ± 5.614	1.011 ± 0.022	1.015 ± 0.034	63.946 ± 10.55	61.21 ± 12.498
	arbitrary_insertion	5.349 ± 3.795	-10.673 ± 23.551	13.985 ± 6.121	1.008 ± 0.019	1.001 ± 0.004	64.039 ± 9.896	63.056 ± 6.296
	nn	6.948 ± 6.303	-4.184 ± 23.056	17.266 ± 8.653	1.042 ± 0.053	1.055 ± 0.046	57.54 ± 17.577	62.768 ± 10.436
	repetitive_nn	6.249 ± 3.062	-11.567 ± 24.903	15.208 ± 5.761	1.01 ± 0.021	1.014 ± 0.025	54.33 ± 14.277	58.342 ± 9.149
	two_opt	7.807 ± 6.095	-5.589 ± 21.513	15.45 ± 9.141	1.04 ± 0.057	1.063 ± 0.058	63.018 ± 10.991	63.305 ± 10.543

Ref:18

Table: Simulation results for triangular and square grid. Savings are for Red-Gray heuristics over only Gray edge usage. Approximation ratios and Speed-ups are over Brute-Force method. **6 boats randomly generated for 20 simulations**. Results are in the form of mean $\pm$ standard deviation.

Grid	Algo	Cost Saving %	AWD Saving %	Num Chargings Saving %	Approx. Ratio RG	Approx. Ratio G	Speed-up Ratio RG	Speed-up Ratio G
Tri	BruteForce	10.991 $\pm$ 6.183	4.798 $\pm$ 13.217	17.802 $\pm$ 7.777	NA	NA	NA	NA
	clook	10.93 $\pm$ 5.343	8.613 $\pm$ 14.575	17.218 $\pm$ 5.876	1.064 $\pm$ 0.067	1.063 $\pm$ 0.074	336.433 $\pm$ 95.01	295.898 $\pm$ 109.473
	concaveTSP	10.543 $\pm$ 6.389	7.797 $\pm$ 13.268	17.604 $\pm$ 7.871	1.022 $\pm$ 0.041	1.017 $\pm$ 0.044	357.732 $\pm$ 78.98	376.644 $\pm$ 29.189
	nearest_insertion	11.776 $\pm$ 5.436	10.466 $\pm$ 11.957	18.918 $\pm$ 6.634	1.036 $\pm$ 0.044	1.045 $\pm$ 0.051	433.548 $\pm$ 22.384	405.666 $\pm$ 78.451
	farthest_insertion	10.326 $\pm$ 6.226	5.694 $\pm$ 14.083	17.497 $\pm$ 7.996	1.01 $\pm$ 0.02	1.003 $\pm$ 0.012	427.932 $\pm$ 62.458	398.882 $\pm$ 89.541
	cheapest_insertion	10.551 $\pm$ 7.536	7.937 $\pm$ 15.716	17.734 $\pm$ 8.257	1.035 $\pm$ 0.039	1.031 $\pm$ 0.041	381.638 $\pm$ 121.093	400.926 $\pm$ 76.278
	arbitrary_insertion	11.969 $\pm$ 7.948	9.276 $\pm$ 11.237	19.002 $\pm$ 9.155	1.015 $\pm$ 0.039	1.029 $\pm$ 0.047	387.22 $\pm$ 112.774	376.091 $\pm$ 102.843
	nn	9.108 $\pm$ 9.199	7.868 $\pm$ 16.067	15.374 $\pm$ 9.216	1.1 $\pm$ 0.087	1.079 $\pm$ 0.074	411.601 $\pm$ 81.716	426.876 $\pm$ 31.193
	repetitive_nn	10.034 $\pm$ 6.079	4.706 $\pm$ 13.76	16.918 $\pm$ 7.583	1.026 $\pm$ 0.041	1.015 $\pm$ 0.035	396.865 $\pm$ 90.325	393.541 $\pm$ 30.383
	two_opt	7.762 $\pm$ 8.984	-1.402 $\pm$ 22.882	15.188 $\pm$ 8.953	1.072 $\pm$ 0.07	1.036 $\pm$ 0.058	413.163 $\pm$ 72.171	362.235 $\pm$ 132.171
Sq	BruteForce	8.037 $\pm$ 4.109	0.78 $\pm$ 11.454	19.775 $\pm$ 8.391	NA	NA	NA	NA
	clook	7.499 $\pm$ 3.399	3.383 $\pm$ 15.292	19.268 $\pm$ 7.381	1.084 $\pm$ 0.059	1.077 $\pm$ 0.05	342.447 $\pm$ 92.399	320.1 $\pm$ 94.281
	concaveTSP	8 $\pm$ 4.041	0.002 $\pm$ 14.457	19.747 $\pm$ 7.831	1.014 $\pm$ 0.025	1.013 $\pm$ 0.026	383.352 $\pm$ 65.676	356.616 $\pm$ 82.259
	nearest_insertion	8.317 $\pm$ 5.231	-3.645 $\pm$ 20.151	19.222 $\pm$ 9.607	1.043 $\pm$ 0.052	1.047 $\pm$ 0.05	385.321 $\pm$ 110.161	407.548 $\pm$ 66.634
	farthest_insertion	7.933 $\pm$ 3.942	1.142 $\pm$ 12.447	19.169 $\pm$ 7.964	1.004 $\pm$ 0.012	1.003 $\pm$ 0.007	397.872 $\pm$ 114.178	405.063 $\pm$ 71.121
	cheapest_insertion	6.832 $\pm$ 5.031	-5.778 $\pm$ 14.923	17.648 $\pm$ 9.011	1.055 $\pm$ 0.058	1.041 $\pm$ 0.046	383.373 $\pm$ 104.393	357.335 $\pm$ 136.662
	arbitrary_insertion	7.676 $\pm$ 6.205	-3.677 $\pm$ 15.141	18.642 $\pm$ 8.981	1.03 $\pm$ 0.051	1.027 $\pm$ 0.042	406.005 $\pm$ 101.385	425.686 $\pm$ 17.989
	nn	4.171 $\pm$ 9.726	-10.233 $\pm$ 28.861	14.711 $\pm$ 12.631	1.105 $\pm$ 0.099	1.062 $\pm$ 0.068	404.158 $\pm$ 94.317	411.081 $\pm$ 72.39
	repetitive_nn	8.181 $\pm$ 3.959	2.497 $\pm$ 12.103	20.155 $\pm$ 8.284	1.015 $\pm$ 0.028	1.016 $\pm$ 0.031	383.738 $\pm$ 86.72	352.204 $\pm$ 105.294
	two_opt	5.988 $\pm$ 8.201	-0.469 $\pm$ 21.566	17.272 $\pm$ 9.147	1.072 $\pm$ 0.063	1.052 $\pm$ 0.072	414.197 $\pm$ 70.865	409.881 $\pm$ 67.041

Ref:18

Table: Simulation results for triangular and square grids. Savings are for Red-Gray heuristics over only Gray edge usage. Approximation ratios and Speed-ups are over Brute-Force method. **7 boats randomly generated for 20 simulations**. Results are in the form of mean ± standard deviation.

Grid	Algo	Cost Saving %	AWD Saving %	Num Chargings Saving %	Approx. Ratio RG	Approx. Ratio G	Speed-up Ratio RG	Speed-up Ratio G
Tri	BruteForce	12.182 ± 5.258	8.565 ± 13.825	18.825 ± 7.611	NA	NA	NA	NA
	clook	10.969 ± 5.487	8.333 ± 10.87	16.709 ± 8.228	1.065 ± 0.089	1.049 ± 0.074	2323.506 ± 1090.313	2519.976 ± 894.221
	concaveTSP	12.09 ± 5.201	4.562 ± 17.664	18.627 ± 7.652	1.026 ± 0.046	1.025 ± 0.042	2846.816 ± 910.073	2707.85 ± 923.289
	nearest_insertion	12.903 ± 4.748	9.577 ± 10.715	19.629 ± 6.968	1.035 ± 0.055	1.043 ± 0.049	3072.458 ± 1125.865	3280.95 ± 666.735
	farthest_insertion	11.494 ± 4.859	5.891 ± 15.457	18.655 ± 7.181	1.018 ± 0.039	1.01 ± 0.024	3499.508 ± 338.037	3283.165 ± 674.529
	cheapest_insertion	11.665 ± 6.103	8.171 ± 13.865	18.531 ± 7.624	1.028 ± 0.039	1.023 ± 0.028	3169.709 ± 920.07	3136 ± 888.583
	arbitrary_insertion	13.199 ± 6.874	6.87 ± 19.079	19.867 ± 8.467	1.016 ± 0.024	1.03 ± 0.045	2850.124 ± 1175.964	3116.143 ± 921.251
	nn	8.691 ± 10.327	4.479 ± 19.718	14.267 ± 11.53	1.092 ± 0.077	1.054 ± 0.059	3252.765 ± 978.095	3280.022 ± 669.766
	repetitive_nn	11.649 ± 5.231	9.235 ± 11.71	17.888 ± 7.777	1.026 ± 0.045	1.019 ± 0.036	2798.433 ± 1200.604	3065.354 ± 617.465
	two_opt	8.94 ± 8.408	5.095 ± 13.874	15.385 ± 7.902	1.104 ± 0.112	1.064 ± 0.065	3428.719 ± 341.631	3125.57 ± 913.372
Sq	BruteForce	7.451 ± 3.61	1.784 ± 11.452	18.634 ± 5.744	NA	NA	NA	NA
	clook	6.704 ± 3.784	6.593 ± 7.946	15.669 ± 6.201	1.062 ± 0.083	1.053 ± 0.077	3423.964 ± 1033.882	3590.896 ± 1381.162
	concaveTSP	7.299 ± 3.705	-1.071 ± 16.145	17.614 ± 6.566	1.015 ± 0.029	1.014 ± 0.028	3555.98 ± 1101.093	3789.317 ± 1458.101
	nearest_insertion	8.283 ± 5.275	-0.304 ± 22.146	18.255 ± 6.84	1.046 ± 0.061	1.056 ± 0.054	3763.436 ± 1462.795	3920.006 ± 1806.181
	farthest_insertion	7.949 ± 3.502	-0.917 ± 17.705	17.77 ± 5.714	1.004 ± 0.008	1.01 ± 0.02	4093.449 ± 923.356	4195.969 ± 1641.237
	cheapest_insertion	7.636 ± 6.393	-2.675 ± 23.218	18.024 ± 8.32	1.032 ± 0.042	1.037 ± 0.054	3669.774 ± 1408.791	4606.16 ± 1007.742
	arbitrary_insertion	8.277 ± 4.423	-0.258 ± 18.75	17.649 ± 6.338	1.015 ± 0.024	1.025 ± 0.033	3721.356 ± 1438.948	4400.094 ± 1373.281
	nn	4.616 ± 11.159	-2.709 ± 16.41	14.559 ± 10.18	1.097 ± 0.1	1.068 ± 0.065	3144.783 ± 1851.757	4589.522 ± 1012.535
	repetitive_nn	7.425 ± 3.603	1.651 ± 14.22	16.981 ± 5.902	1.018 ± 0.03	1.018 ± 0.035	3709.634 ± 1144.465	3681.774 ± 1690.398
	two_opt	5.261 ± 7.635	-1.83 ± 18.827	14.874 ± 7.78	1.08 ± 0.073	1.058 ± 0.071	3910.01 ± 1221.312	4606.714 ± 1009.613

Ref:18

Table: Benchmark results for approximate TSP tour costs in units.

Dataset	concaveTSP	FI	NN	2-Opt
myLattice-25x40-1000	10437.0	10622.7	12225.1	10864.0
myLattice-50x40-2000	20576.3	21256.0	24264.8	21571.5
myLattice-50x60-3000	30601.7	31877.7	36396.3	32275.8
myRNDLattice-29x46-1000	13545.6	11324.5	13122.9	11597.9
myRNDLattice-58x46-2000	28001.3	22720.8	26033.6	23181.4
myRNDLattice-58x69-3000	42777.2	34129.5	38661.3	34664.3
myHexLattice-25x40-1000	10455.2	10494.8	12280.5	10730.5
myHexLattice-50x40-2000	20439.9	20534.8	21364.9	20863.0
myHexLattice-50x60-3000	30667.5	30815.9	31839.5	31193.8
myRNDHexLattice-29x46-1000	12255.9	11092.5	12798.8	11233.1
myRNDHexLattice-58x46-2000	23334.8	21695.8	24520.3	21858.7
myRNDHexLattice-58x69-3000	35494.9	32452.0	36658.8	32582.3

Ref:18

Table: Benchmark results for running time in seconds.

Dataset	concaveTSP	FI	NN	2-Opt
myLattice-25x40-1000	0.279	3.088	0.060	2.063
myLattice-50x40-2000	0.644	20.307	0.176	31.207
myLattice-50x60-3000	0.988	67.366	0.360	153.377
myRNDLattice-29x46-1000	0.241	3.093	0.063	2.213
myRNDLattice-58x46-2000	0.527	20.298	0.169	32.087
myRNDLattice-58x69-3000	0.858	67.522	0.365	164.172
myHexLattice-25x40-1000	0.078	3.072	0.063	2.191
myHexLattice-50x40-2000	0.117	20.378	0.189	36.102
myHexLattice-50x60-3000	0.147	68.024	0.388	160.825
myRNDHexLattice-29x46-1000	0.152	3.072	0.052	2.305
myRNDHexLattice-58x46-2000	0.273	21.087	0.182	36.376
myRNDHexLattice-58x69-3000	0.448	69.281	0.353	185.657

Ref:18

Table: Benchmark results for approximate TSP tour AWD (from vertex 1) costs in units.

Dataset	concaveTSP	FI	NN	2-Opt
myLattice-25x40-1000	5300.1	5320.5	5948.3	5440.4
myLattice-50x40-2000	10395.3	10632.3	12007.0	10810.2
myLattice-50x60-3000	15389.2	15953.1	18339.8	16139.2
myRNDLattice-29x46-1000	6863.2	5668.0	6575.0	5810.1
myRNDLattice-58x46-2000	13972.7	11363.8	13010.3	11615.3
myRNDLattice-58x69-3000	21385.0	17064.1	19581.6	17348.2
myHexLattice-25x40-1000	5229.7	5254.2	6203.6	5380.4
myHexLattice-50x40-2000	10225.1	10274.0	10819.5	10469.0
myHexLattice-50x60-3000	15341.0	15415.2	16102.9	15631.5
myRNDHexLattice-29x46-1000	6257.2	5552.3	6501.7	5622.2
myRNDHexLattice-58x46-2000	11477.9	10856.9	12388.4	10930.5
myRNDHexLattice-58x69-3000	17782.4	16225.2	18310.3	16295.5

Ref:18

Statistics from Italian Gov.(?) *:
"Every year, boating activities require to deal with a high number of rescuing calls. **In Italy in 2013, 6166 boats called for help.** During the summer period (35 days) about 1152 people called for aid in Marche region. . . . false alarms. More precisely, only 2.4% of the calls requires rescuing, thus lifeboat assistance."

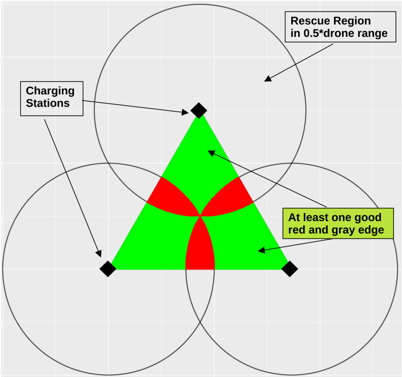
Basically for 2013:
$\frac{(6166\times\frac{2.4}{100})}{365} = 0.405$ → **about a call every other day!**
Considers only the calls people managed to made.
"Survivorship bias" blues! → Remember the ones that couldn't!

*Formica, N., Mostarda, L., Navarra, A. (2021). UAVs Route Planning in Sea Emergencies. In: Barolli, et al. (eds) AINA 2021. LNNS, vol 225. Springer. `https://doi.org/10.1007/978-3-030-75100-5_51`

Ref:1

Tri CS Grid: Good red-gray points

RQ: How often can you have such red-gray edges in Tri Grid?



Prob.: 0.78

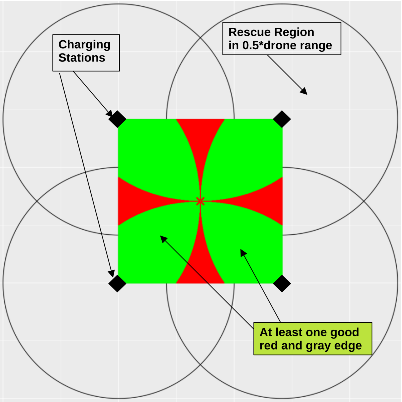
Table: **Triangular grid**, 69483 points are sampled.
b, g: bad, good
G, R: Gray, Red
Ex: 1gR+1G+1bR → 1 good Red + 1 Gray + 1 bad Red

Point Type	N	Prob.
3bR	0	0
1G+2bR	0	0
2G+1bR	15171	0.218
3G	5	0.00007
1gR+2G	68	0.00098
1gR+1G+1bR	14428	0.208
2gR+1G	39811	0.573

Ref:13

Sq CS Grid: Good red-gray points

RQ: How often can you have such red-gray edges in Sq Grid?



Prob.: 0.82

Table: **Square grid**, 160801 points are sampled.
b, g: bad, good
G, R: Gray, Red
Ex: 1G+2gR+1bR → 1 Gray + 2 good Red + 1 bad Red

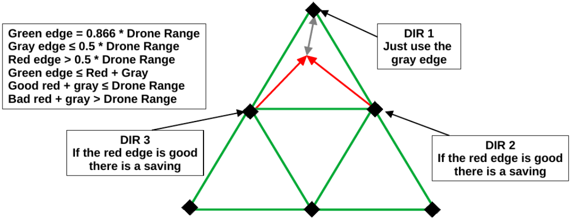
Point Type	N	Prob.
4bR	0	0
1G+3bR	0	0
2G+2bR	29452	0.183
3G+1bR	64	0.0004
4G	5	0.00003
1G+3gR	736	0.0046
2G+2gR	0	0
1G+2gR+1bR	66752	0.415
3G+1gR	64	0.0004
2G+1gR+1bR	63728	0.396
1G+1gR+2bR	0	0

Sq Grid wins!

Ref:13

Prob. of using a Good Red-Gray Path

RQ: How often can you benefit from these red-gray edges?



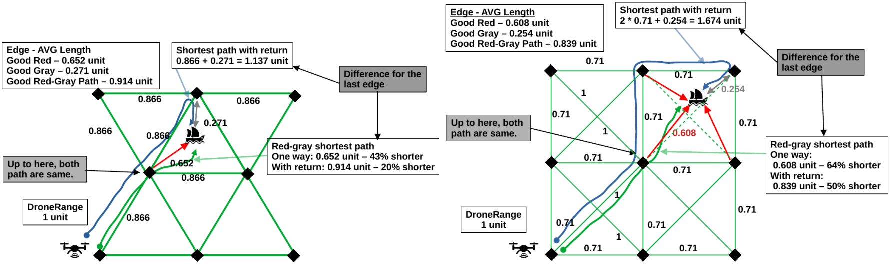
$$
\begin{aligned}
\boldsymbol{P(Benefit_{Tri})} &= P(1gR) \times P(gD) + P(2gR) \times P(gD) \\
&= 0.21 \times 0.333 + 0.57 \times 0.666 = \mathbf{0.45} \\
\boldsymbol{P(Benefit_{Sq})} &= P(1gR) \times P(gD) + P(2gR) \times P(gD) + P(3gR) \times P(gD) \\
&= 0.3964 \times 0.25 + 0.415 \times 0.5 + 0.0046 \times 0.75 = \mathbf{0.31}
\end{aligned}
$$

Tri Grid wins!

Ref:13

Tri vs Sq CS Grid: Theoretical Savings†

RQ: How much savings can you get from these red-gray edges?



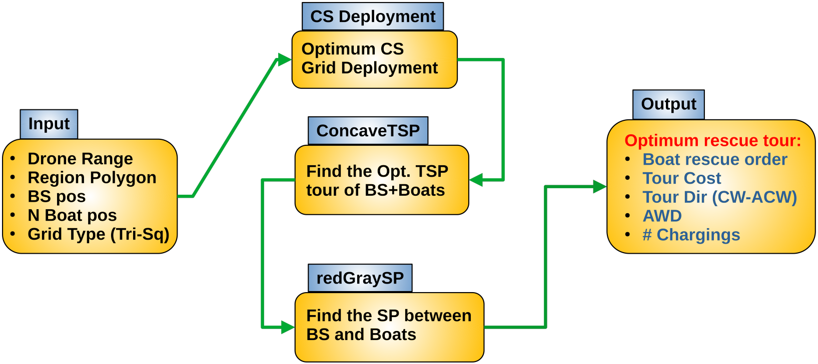
Savings from red-gray path heuristic compared to base case (go to the CS nearest to the Boat).

(Know how much profit you can get from your "heuristic"!) Sq Grid wins!

† Edge statistics table in Appendix 21

Ref:13

Boat Rescue: Process Flow



Ref:2

List of Publications (only aff. UNICAM) - I

- Kilic K.I., Gemikonakli O., Mostarda L. (2020) **Multi-objective Priority Based Heuristic Optimization for Region Coverage with UAVs**. AINA, Advances in Intelligent Systems and Computing, 1151:768–779. Springer.
DOI: https://doi.org/10.1007/978-3-030-44041-1_68

- Kilic, K. I., Gemikonakli, O. and Mostarda, L. (2021), **Voronoi Tesselation-based load-balanced multi-objective priority-based heuristic optimisation for multi-cell region coverage with UAVs**, International Journal of Web and Grid Services 17(2), 152-178.
DOI: https://doi.org/10.1504/IJWGS.2021.114574

List of Publications (only aff. UNICAM) - II

- Kilic K.I., Mostarda L. (2021) **Optimum Path Finding Framework for Drone Assisted Boat Rescue Missions**. AINA, Lecture Notes in Networks and Systems, 227:219–231. Springer.
DOI: `https://doi.org/10.1007/978-3-030-75078-7_23`
- Kilic K.I., Mostarda L. (2021) **Heuristic Drone Pathfinding over Optimised Charging Station Grid**. IEEE Accesss, vol. 9, pp. 164070-164089,
DOI: `https://doi.org/10.1109/ACCESS.2021.3134459`
- Kilic K.I., Mostarda L. (2022) **Novel Concave Hull-Based Heuristic Algorithm For TSP**, Operations Research Forum, Springer Nature, 3(2):25, DOI: `https://doi.org/10.1007/s43069-022-00137-9`

References I

[1] AL-HOURANI, A., KANDEEPAN, S., AND LARDNER, S.
Optimal LAP Altitude for Maximum Coverage.
IEEE Wireless Communications Letters 3, 6 (2014), 569–572.
http://dx.doi.org/10.1109/LWC.2014.2342736.

[2] AL-TURJMAN, F., LEMAYIAN, J. P., ALTURJMAN, S., AND MOSTARDA, L.
Enhanced Deployment Strategy for the 5G Drone-BS Using Artificial Intelligence.
IEEE Access 7 (2019), 75999–76008.
http://dx.doi.org/10.1109/ACCESS.2019.2921729.

[3] ARDIA, D., MULLEN, K., PETERSON, B., ULRICH, J., AND BOUDT, K.
DEoptim: Global Optimization by Differential Evolution, 2016.
https://cran.r-project.org/web/packages/DEoptim/index.html, online; accessed 12-September-2019.

[4] CHETLUR RAVI, V. V., AND DHILLON, H. S.
Downlink coverage probability in a finite network of UAV base stations.
In *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (July 2016), pp. 1–5.
http://dx.doi.org/10.1109/SPAWC.2016.7536859.

References II

[5] Choi, C. H., Jang, H. J., Lim, S. G., Lim, H. C., Cho, S. H., and Gaponov, I.
Automatic wireless drone charging station creating essential environment for continuous drone operation.
In *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)* (2016), pp. 132–136.
https://doi.org/110.1109/ICCAIS.2016.7822448.

[6] Collins, C. R., and Stephenson, K.
A circle packing algorithm.
Computational Geometry 25, 3 (2003), 233–256.
https://doi.org/10.1016/S0925-7721(02)00099-8.

[7] Deineko, V. G., van Dal, R., and Rote, G.
The convex-hull-and-line traveling salesman problem: a solvable case.
Information Processing Letters 51, 3 (1994), 141–148.
https://doi.org/10.1016/0020-0190(94)00071-9.

[8] Deĭneko, V. G., and Woeginger, G. J.
The Convex-hull-and-k-line Travelling Salesman Problem.
Information Processing Letters 59, 6 (1996), 295–301.
https://doi.org/10.1016/0020-0190(96)00125-1.

References III

[9] Galkin, B., Kibilda, J., and DaSilva, L. A.
UAVs as Mobile Infrastructure: Addressing Battery Lifetime.
IEEE Communications Magazine 57, 6 (2019), 132–137.
https://doi.org/10.1109/MCOM.2019.1800545.

[10] Helsgaun, K.
An effective implementation of the Lin–Kernighan traveling salesman heuristic.
European Journal of Operational Research 126, 1 (2000), 106–130.
https://doi.org/10.1016/S0377-2217(99)00284-2.

[11] Huang, H., and Savkin, A. V.
A Method of Optimized Deployment of Charging Stations for Drone Delivery.
IEEE Transactions on Transportation Electrification 6, 2 (2020), 510–518.
https://doi.org/10.1109/TTE.2020.2988149.

[12] Huang, H., and Savkin, A. V.
Optimal Deployment of Charging Stations for Aerial Surveillance by UAVs with the Assistance of Public Transportation Vehicles.
Sensors 21, 16 (2021), 5320.
https://doi.org/10.3390/s21165320.

[13] Jones, J., and Adamatzky, A.
Computation of the travelling salesman problem by a shrinking blob.
Natural Computing 13, 1 (2014), 1–16.
https://doi.org/10.1007/s11047-013-9401-x.

References IV

[14] LI, X., GUO, D., YIN, H., AND WEI, G.
Drone-assisted public safety wireless broadband network.
In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (March 2015), pp. 323–328.
http://dx.doi.org/10.1109/WCNCW.2015.7122575.

[15] LIANSHUAN, S., AND ZENGYAN, L.
An Improved Pareto Genetic Algorithm for Multi-objective TSP.
In *2009 Fifth International Conference on Natural Computation* (2009), vol. 4, pp. 585–588.
https://doi.org/10.1109/ICNC.2009.510.

[16] LUST, T., AND TEGHEM, J.
The Multiobjective Traveling Salesman Problem: A Survey and a New Approach.
In *Advances in Multi-Objective Nature Inspired Computing*, Coello Coello, Carlos A. and Dhaenens, Clarisse and Jourdan, Laetitia, Ed. Springer Berlin Heidelberg, 2010, pp. 119–141.
https://doi.org/10.1007/978-3-642-11218-8_6.

[17] MAINI, P., AND SUJIT, P. B.
On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications.
In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (2015), pp. 1370–1377.
https://doi.org/10.1109/ICUAS.2015.7152432.

References V

[18] MANTHEY, B., AND SHANKAR RAM, L.
Approximation Algorithms for Multi-Criteria Traveling Salesman Problems.
Algorithmica 53, 1 (2009), 69–88.
https://doi.org/10.1007/s00453-007-9011-z.

[19] MICHINI, B., TOKSOZ, T., REDDING, J., MICHINI, M., HOW, J., VAVRINA, M., AND VIAN, J.
Automated Battery Swap and Recharge to Enable Persistent UAV Missions.
In *Infotech@Aerospace 2011* (2011).
http://dx.doi.org/10.2514/6.2011-1405.

[20] MULLEN, K.
Continuous Global Optimization in R.
Journal of Statistical Software, Articles 60, 6 (2014), 1–45.
http://dx.doi.org/10.18637/jss.v060.i06.

[21] OUYANG, J., CHE, Y., XU, J., AND WU, K.
Throughput Maximization for Laser-Powered UAV Wireless Communication Systems.
In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)* (2018), pp. 1–6.
https://doi.org/10.1109/ICCW.2018.8403572.

[22] REINELT, G.
TSPLIB—A Traveling Salesman Problem Library.
ORSA Journal on Computing 3, 4 (1991), 376–384.
https://doi.org/10.1287/ijoc.3.4.376.

References VI

[23] SCRUCCA, L.
GA: Genetic Algorithms, 2019.
https://cran.r-project.org/web/packages/GA/index.html, online; accessed 12-September-2019.

[24] SUNDAR, K., AND RATHINAM, S.
Algorithms for Routing an Unmanned Aerial Vehicle in the Presence of Refueling Depots.
IEEE Transactions on Automation Science and Engineering 11, 1 (2014), 287–294.
https://doi.org/10.1109/TASE.2013.2279544.

[25] XIANG, Y., GUBIAN, S., SUOMELA, B., AND HOENG, J.
GenSA: Generalized Simulated Annealing, 2019.
https://cran.r-project.org/web/packages/GenSA/index.html, online; accessed 12-September-2019.

[26] YIN, S., ZHAO, Y., AND LI, L.
Resource Allocation and Basestation Placement in Cellular Networks With Wireless Powered UAVs.
IEEE Transactions on Vehicular Technology 68, 1 (2019), 1050–1055.
https://doi.org/10.1109/TVT.2018.2883093.

[27] ZORBAS, D., PUGLIESE, L. D. P., RAZAFINDRALAMBO, T., AND GUERRIERO, F.
Optimal drone placement and cost-efficient target coverage.
Journal of Network and Computer Applications 75 (2016), 16–31.
https://doi.org/10.1016/j.jnca.2016.08.009.