Evaluating Hardware Platforms and Path Re-Planning Strategies for the UAV Emergency Landing Problem

Jesimar S. Arantes, Márcio S. Arantes, André B. Missaglia,
Eduardo V. Simões and Claudio F. M. Toledo

November – 2017

Outline

1. Introduction
2. Problem Description
3. Methods
4. Computational Results
5. Conclusions

Introduction

This work addresses

- Path replanning to land the UAV under critical situation:
  - Greedy Heuristic (GH) (Arantes et al. 2015).
  - Genetic Algorithm (GA) (Arantes et al. 2015).
  - Ensemble GA-GH.
  - Ensemble GA-GA.
- Critical situations considered:
  - Motor failure.
  - Battery failure.
- Supervision of safety systems
  - IFA: In-Flight Awareness (Mattei et al. 2013).
- Performance evaluation:
  - Personal computer: Intel i5
  - Embedded computer: Intel Edison

Problem Description

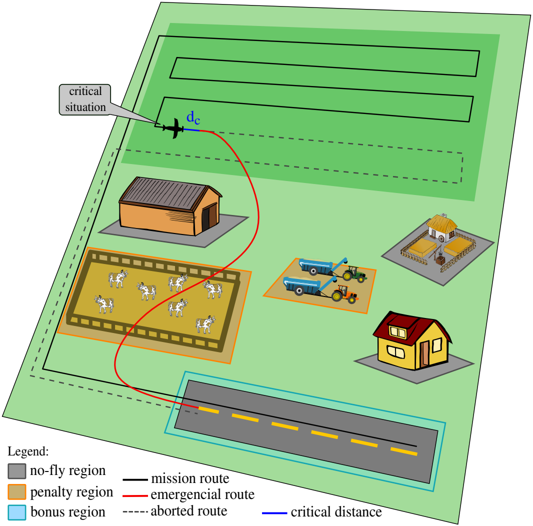


Figure 1: Illustrative scenario of emergency path re-planning.

Problem Description

Problem: Path-replanning problem

$$Minimize\ Pr\Big(\bigvee_{j} \mathbf{x}_T \in \mathbb{O}_j^p\Big) - Pr\Big(\bigvee_{i} \mathbf{x}_T \in \mathbb{O}_j^b\Big) \tag{1}$$

subject to:

$$\mathbf{x}_{t+1} = F_{\Psi}(\mathbf{x}_t, \mathbf{u}_t) + \omega_t \qquad \forall(t) \tag{2}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_{x_0}), \quad \omega_t \sim \mathcal{N}(0, \Sigma_{\omega_t}) \qquad \forall(t) \tag{3}$$

$$Pr\left[\bigwedge_{j}\bigwedge_{t} \mathbf{x}_t \notin \mathbb{O}_j^n\right] \geq 1 - \Delta \tag{4}$$

Problem Description

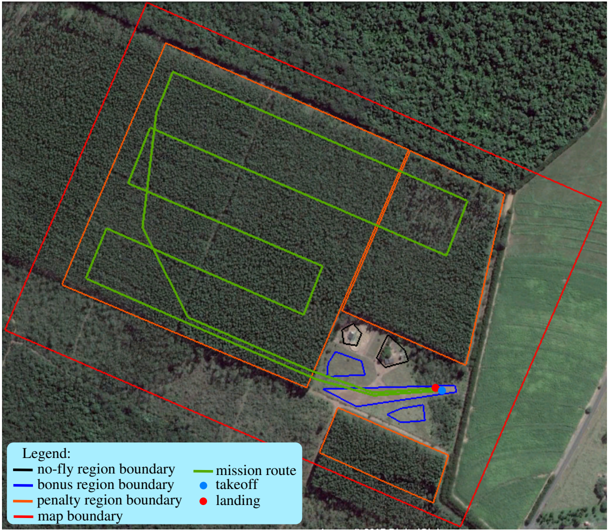


Figure 2: Real world case study for aerial imaging application.

Methods

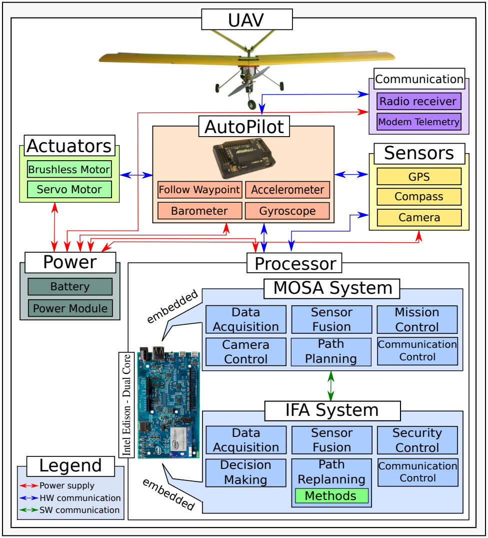


Figure 3: Embedded system architecture in the aircraft.

Methods

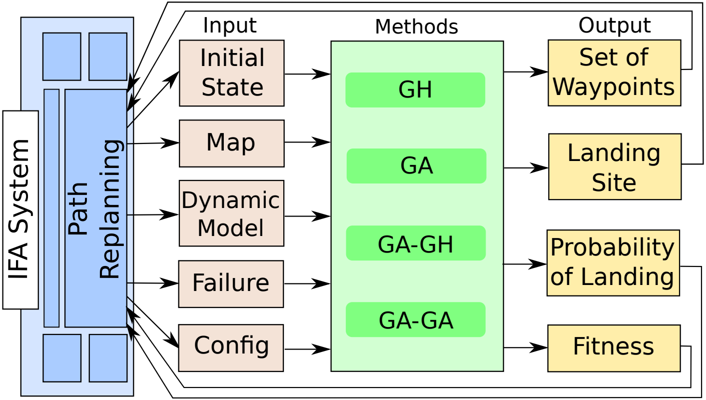


Figure 4: Embedded system architecture in the path re-planning module.

Methods

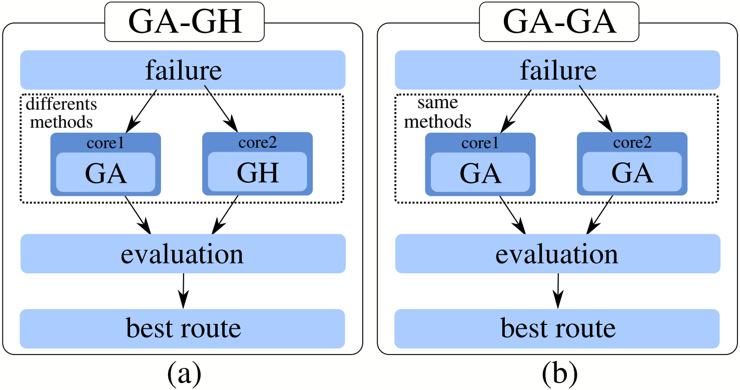


Figure 5: Implemented strategies running methods in parallel. (a) Combining of GA e GH. (b) Two executions of GA.

Methods

Algorithm 1: Greedy Heuristic.

```
begin
    EmergencyRoute listRoutes ← ∅;
    foreach region in map.setBonusRegions do
        EmergencyRoute route;
        initialize(route, region);
        evaluate(route, map);
        listRoutes.add(route);
    EmergencyRoute bestRoute ← getBestRoute(listRoutes);
    return bestRoute;
```

Methods

Algorithm 2: Genetic Algorithm.

```
begin
    EmergencyRoute vectorRoutes[numIndividuals];
    for i = 1 to numIndividuals do
        EmergencyRoute route;
        initialize(route, map);
        evaluate(route, map);
        vectorRoutes.add(route);
    repeat
        repeat
            for i = 1 to crossRate × numIndividuals do
                EmergencyRoute ind1, ind2;
                select(ind1, ind2);
                EmergencyRoute child ← crossover(ind1, ind2);
                mutation(child);
                evaluate(child, map);
                vectorRoutes.add(child);
        until converge(vectorRoutes);
        restart(vectorRoutes);
    until reach(stoppingCriterion);
    EmergencyRoute bestRoute ← getBestRoute(vectorRoutes);
    return bestRoute;
```

Computational Results

- Settings used in the GA method.

Parameter	Value	Parameter	Value
population size	39	*crossover rate*	0.5
tournament size	3	*mutation rate*	0.7
stopping criterion	time	*elitism*	yes

- Computer settings used.

	PC i5	Intel Edison
Frequency	1.8 GHz	500 MHz
Memory RAM	4 GB	1 GB
Operating System	Linux - Ubuntu	Linux - Yocto

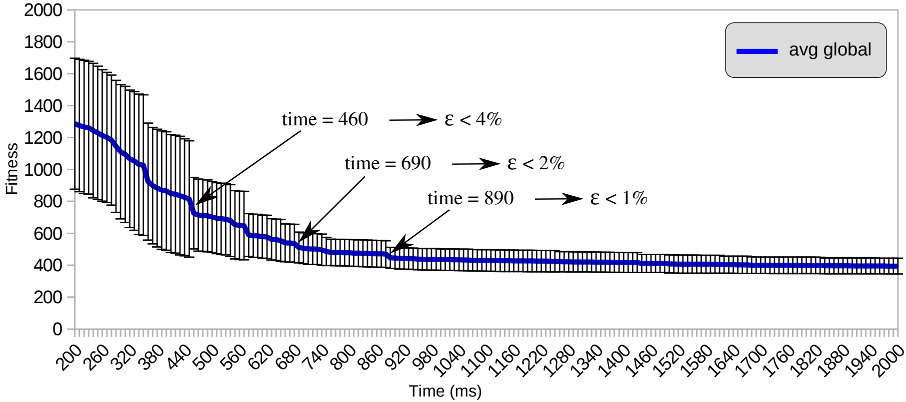
Computational Results

- Technical specifications of the Ararinha.

Component	Value	Component	Value
Wingspan	1.90m	Weight	2.83kg
Length	1.15m	Payload	0.60kg
Electric Power	740W	Endurance	15 minutes

Computational Results

- Time analysis for convergence of the GA.



Computational Results

- GH
  - We run 1 time over each map
  - We evaluated 30 artificial maps in total
  - We evaluated 2 critical situations (motor and battery)
- GA, GA-GH, GA-GA
  - We run 10 times over each map
  - We evaluated 30 artificial maps in total
  - We evaluated 2 critical situations (motor and battery)
  - We evaluated 3 stopping criterion based on time

Computational Results

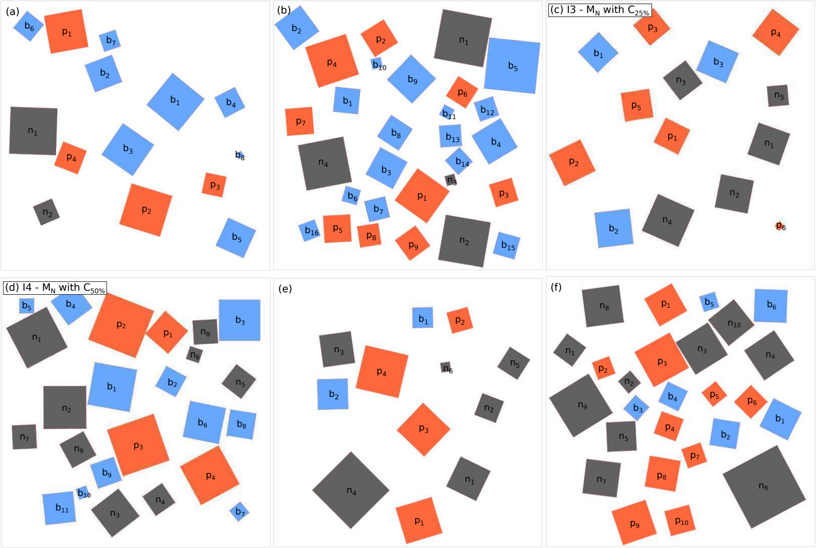


Figure 6: Artificial maps.

Computational Results

Table 1: Results obtained after evaluating different strategies in artificial maps.

		Intel i5				Intel Edison			
Methods	SC (ms)	Ψ_b	Ψ_m	Avg	Time (ms)	Ψ_b	Ψ_m	Avg	Time (ms)
GH	-	86.7%	60.0%	73.3%	41	86.7%	60.0%	73.3%	347
GA	250	96.3%	72.0%	84.2%	250	72.3%	62.3%	67.3%	250
GA	500	99.0%	73.0%	86.0%	500	84.7%	65.3%	75.0%	500
GA	1000	98.3%	75.7%	87.0%	1000	91.0%	68.0%	79.5%	1000
GA-GH	250	95.3%	71.0%	83.2%	250	89.3%	62.7%	76.0%	309
GA-GH	500	100.0%	72.3%	86.2%	500	90.3%	65.7%	78.0%	510
GA-GH	1000	99.3%	76.0%	87.7%	1000	92.3%	69.0%	80.7%	1000
GA-GA	250	99.3%	72.3%	85.8%	250	81.0%	65.0%	73.0%	250
GA-GA	500	99.7%	73.3%	86.5%	500	89.7%	68.3%	79.0%	500
GA-GA	1000	99.7%	78.0%	88.8%	1000	94.7%	70.0%	82.3%	1000
Final Avg	-	97.4%	72.4%	84.9%	-	87.2%	65.6%	76.4%	-

Computational Results

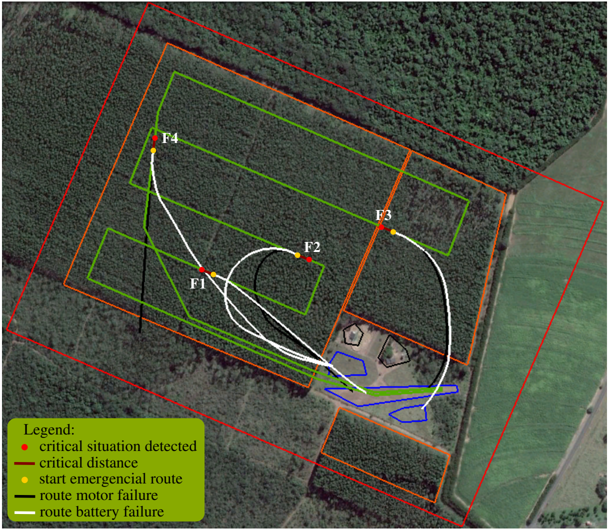


Figure 7: Results of case study in a real world scenario using a GA method.

Computational Results

Table 2: Results obtained after evaluating different strategies in the study case.

		Intel i5				Intel Edison			
Methods	SC (ms)	Ψ_b	Ψ_m	Avg	Time (ms)	Ψ_b	Ψ_m	Avg	Time (ms)
GH	-	25.0%	0.0%	12.5%	22	25.0%	0.0%	12.5%	192
GA	250	100.0%	50.0%	75.0%	250	35.0%	37.5%	36.3%	250
GA	500	100.0%	50.0%	75.0%	500	62.5%	50.0%	56.3%	500
GA	1000	100.0%	67.5%	83.8%	1000	95.0%	50.0%	72.5%	1000
GA-GH	250	100.0%	50.0%	75.0%	250	37.5%	37.5%	37.5%	234
GA-GH	500	100.0%	50.0%	75.0%	500	67.5%	50.0%	58.8%	481
GA-GH	1000	100.0%	75.0%	87.5%	1000	92.5%	50.0%	71.3%	992
GA-GA	250	100.0%	50.0%	75.0%	250	50.0%	45.0%	47.5%	250
GA-GA	500	100.0%	50.0%	75.0%	500	85.0%	50.0%	67.5%	500
GA-GA	1000	100.0%	75.0%	87.5%	1000	100.0%	50.0%	75.0%	1000
Final Avg	-	92.5%	51.8%	72.1%	-	65.0%	42.0%	53.5%	-

Computational Results

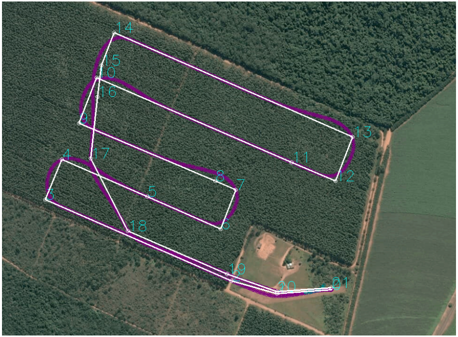


Figure 8: Complete route obtained in the SITL experiment.

https://youtu.be/k38GBjM5jXU

Conclusions

- As expected, the personal computer leverages the strategies performance.
- GA-GA presents a slightly better performance.
- This indicates the robustness of GAs to find good solutions despite hardware limitations.
- GA can be embedded and aid the decision making during fully autonomous flights.

Acknowledgements

Email to contact:

{jesimar.arantes, andrebm }@usp.br
{marcio, simoes, claudio}@icmc.usp.br

Thank You!