AI-guided optimal deployments of drone-intercepting systems in large critical areas

Marco Esposito

Computer Science Dept., Sapienza University of Rome, Italy

Abstract

The problem of designing effective systems to prevent risks and hazards caused by drones in critical areas has gained significant attention in recent years. In this short paper, we introduce the idea of computing optimal deployments of anti-drone sensors in a given region using simulation-based optimisation via heuristic-guided intelligent search and a geometric modelling of the problem, and show preliminary results in a real-world scenario.

Keywords

derivative-free optimisation, symbolic modelling, anti-drone systems, optimal sensor placement

1. Introduction

In safety-critical areas, such as airports, power plants and government buildings complexes, small unmanned aerial vehicles (UAVs, aka *drones*) nowadays represent a major hazard, due to the risks of collisions, radio jamming, or other terror crimes. To protect such areas, anti-drone systems are often deployed, which include sensors devoted to the detection and localisation of drones. Different technologies are available to implement anti-drone systems [1, 2]; here, we focus on radio-frequency (RF) sensors, which can detect the direction of arrival of signals emitted by drones and that can determine target location via triangulation. This kind of sensors can detect signals over large distances (1-5 km) with high accuracy, but require line-of-sight visibility of the target. A major problem in designing an anti-drone system is to determine where to deploy sensors in order to maximise their coverage of the region of interest (RoI), *i.e.*, the fraction of the region where drones can be localised with high accuracy. Since each sensor may cost up to several tens of thousands of Euros, it is also crucial to minimise the total cost of the deployment, which depends both on the number of sensors employed and on their selected positions (which implies different costs for mounting, wiring, etc.)

2. Modelling

Large critical areas often present certain characteristics that make it difficult to compute an optimal deployment of anti-drone sensors. In [3, 4], the RoI has been discretised in cells of

OVERLAY 2021: 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, September 22, 2021, Padova, Italy
esposito@di.uniroma1.it (M. Esposito)
0000-0003-4543-8818 (M. Esposito)

hundreds of square meters each, in order to compute, via Mixed Integer Linear Programming (MILP), Pseudo-Boolean and Satisfiability/Optimisation Modulo Theory solvers an optimal placement of relay nodes able to convey, in a fault-tolerant way, the network traffic from wisely placed RF sensors to the gateway. Such a discretised problem formulation led to the generation of *millions of constraints*.

Unfortunately, following similar methods to compute an optimal placement of the RF sensors themselves would be *unviable*, as, to achieve the desired accuracy in the computation of the quality of the deployment, a much finer discretisation of the RoI would be needed. Also, differently from relay nodes, optimal positions of anti-drone RF sensors could easily belong to the 3D space, as some sensors could be optimally placed on, *e.g.,* , the roofs or walls of buildings. All this would make a suitably-discretised version of the problem *intractably large*.

We introduce a *computational geometry*–based approach to represent the environment and the possible positions of the sensors, by defining the RoI and its obstacles (such as buildings and terrain asperities) as well as RoI priority regions and admissible sensor positions in terms of *bounded convex polytopes*. This allows us to achieve *any* desired level of accuracy, enables the efficient computation of the coverage of a given deployment via computational geometry techniques, and paves the way to the use of *simulation-based optimisation approaches*.

3. Computing optimised deployments of sensors

Our goal is to compute a deployment of a given set of sensors over a 3D region, optimising the linear combination of two (possibly conflicting) performance metrics: (1) the RoI % not covered by the sensors and (2) the economic cost of purchasing and physically deploying the sensors. The coefficient of the linear combination are chosen so that the objective value defines an amount of money, combining the expense for purchasing and placing the sensors and the implicit cost of not covering part of the RoI. We implemented a simulator that takes as an input the geometric description of the environment and a deployment of sensors and computes the associated value of the objective function (*objective value*). This simulator uses the Parma Polyhedra Library [5] to model the environment as sets of polytopes and to perform the necessary geometric operations (along the lines of [6]), and Monte Carlo estimation to compute an approximation of the objective value, which is then used by our optimisation algorithm to guide the search.

The latter combines *derivative-free optimisation* [7] with domain-based heuristics, and balances exploitation and exploration by alternating two different types of moves (as inspired from [8]). At each iteration, a step based on the *adaptive trust-region method* [9] computes and performs a greedy move that maximally improves the current deployment by moving a single sensor. Whenever a local optimum is reached (*i.e.,* a deployment such that moving any single sensor does not yield any improvement), a *heuristic-guided* random move is performed, where each sensor is randomly moved (in a new admissible position) by a distance randomly chosen depending on the sensor's *contribution* to the coverage of the RoI.

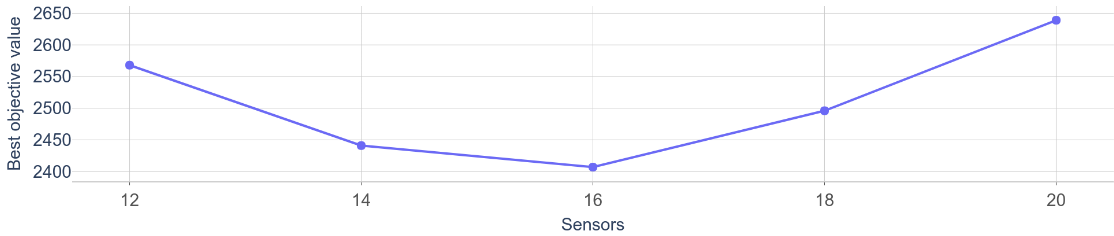

Figure 1: Objective values of the best solutions found with different numbers of sensors.

4. A case study

We evaluated the performance of our optimiser on a large, real-world scenario: the *Leonardo da Vinci* International airport of Rome, Italy. We used sensors with range of 1 km, which could be deployed both on the ground (but not on the runways, on the roads and on the taxiways) and on top of the buildings. We varied the number of deployed sensors between 12 and 20. In each experiment, we first sampled 100 random deployments and sorted them by their objective value (to be minimised) in ascending order. Then, we executed the optimisation algorithm multiple times using the sampled deployments as starting points, from the best to the worse. A time budget of 24 hours was set and the optimisation algorithm was run until either the whole time budget was consumed or all 100 runs were completed.

In most of the cases, the algorithm was able to quickly improve the objective value and find much better deployments than those found by random sampling. Among the best deployments found with different number of sensors (Figure 1), the one that employed 16 sensors showed the minimal objective value, *i.e.,* was the one that best combined the coverage (73.1% of the RoI volume covered) and the total expense for the deployment.

5. Related work

Many attempts have been proposed in the literature to solve *large* optimisation problems defined via logic-, automata- or constraint-based formalisms (*e.g.,* [10, 11, 12, 13]). However, such approaches cannot be applied when the problem model cannot be accurately defined within such formalisms and is available, *e.g.,* only as a black-box (*e.g.,* [14, 15]). In these cases, various kinds of intelligent black-box search or statistical model checking are often performed in the search space and a *simulator* is used to assess the quality of the current parameter assignment as well as to receive, when possible, some kind of gradient information to guide the search. Such approaches have proved to scale well in diverse application domains such as, smart grids (*e.g.,* [16, 17, 18]), system-level verification of cyber-physical systems (*e.g.,* [19, 20, 21, 22, 23, 25, 24]), *in silico* medicine (*e.g.,* [26, 27, 28, 29, 30]). As for the problem of optimally placing sensors in a large region, existing methods differ depending on the types of sensors considered and on the goal of the deployment.

Most works focus on monitoring 2D regions, with or without obstacles (*e.g.,* [31, 32, 33, 34, 35, 36]), hence are not easily adaptable to the localisation of flying objects. Approaches working in the 3D space [31, 37] do exist. However they typically discretise the RoI and the possible positions of sensors [38, 39], and thus do not scale over large scenarios when the targets must be localised with high accuracy.

6. Conclusions

In this short paper we have shown how to compute optimal deployments of anti-drone sensors in a given RoI through simulation-based optimisation via heuristic-guided intelligent search and geometric problem modelling. We also presented preliminary results in a real-world scenario (Leonardo da Vinci airport in Rome, Italy) which show our approach promising.

References

[1] X. Shi, *et al.*, Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges, IEEE Comm Mag 56 (2018).

[2] I. Guvenc, *et al.*, Detection, tracking, and interdiction for amateur drones, IEEE Comm Mag 56 (2018).

[3] T. Mancini, *et al.*, Optimal fault-tolerant placement of relay nodes in a mission critical wireless network, in: RCRA 2018, *CEUR* 2271, 2018.

[4] Q. Chen, *et al.*, MILP, pseudo-boolean, and OMT solvers for optimal fault-tolerant placements of relay nodes in mission critical wireless networks, Fund Inf 174(3-4) (2020).

[5] R. Bagnara, *et al.*, The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems, Sci Comp Progr 72 (2008).

[6] T. Mancini, Now or Never: Negotiating efficiently with unknown or untrusted counterparts, Fund Inf 149 (2016).

[7] C. Audet, *et al.*, Derivative-free and blackbox optimization (2017).

[8] T. Mancini, *et al.*, Combinatorial problem solving over relational databases: View synthesis through constraint-based local search, in: SAC 2012, ACM, 2012.

[9] Y.-x. Yuan, A review of trust region algorithms for optimization, in: Iciam, 99(1), 2000.

[10] M. Cadoli, *et al.*, Combining relational algebra, SQL, constraint modelling, and local search, Theor Pract Logic Progr 7 (2007).

[11] M. Cadoli, *et al.*, SAT as an effective solving technology for constraint problems, in: ISMIS 2006, *LNCS* 4203, Springer, 2006.

[12] T. Mancini, *et al.* An efficient algorithm for network vulnerability analysis under malicious attacks, in: ISMIS 2018, Springer, 2018.

[13] I. Melatti, *et al.* A two-layer near-optimal strategy for substation constraint management via home batteries, IEEE Trans Ind Elect (2021).

[14] F. Maggioli, *et al.*, SBML2Modelica: Integrating biochemical models within open-standard simulation ecosystems, Bioinf 36 (2020).

[15] S. Sinisi, *et al.*, Reconciling interoperability with efficient verification and validation within open source simulation environments, Simul Model Pract Theory 109 (2021).

[16] T. Mancini, *et al.* Demand-aware price policy synthesis and verification services for smart grids, in: SmartGridComm 2014, IEEE, 2014.

[17] T. Mancini, *et al.*, User flexibility aware price policy synthesis for smart grids, in: DSD 2015.
[18] T. Mancini, *et al.*, Parallel statistical model checking for safety verification in smart grids, in: SmartGridComm 2018, IEEE, 2018.
[19] T. Mancini, *et al.*, SyLVaaS: System level formal verification as a service, in: PDP 2015.
[20] T. Mancini, *et al.*, System level formal verification via distributed multi-core hardware in the loop simulation, in: PDP 2014, IEEE, 2014.
[21] T. Mancini, *et al.*, Anytime system level verification via random exhaustive hardware in the loop simulation, in: DSD 2014, IEEE, 2014.
[22] T. Mancini, *et al.*, Anytime system level verification via parallel random exhaustive hardware in the loop simulation, Micropr Microsys 41 (2016).
[23] T. Mancini, *et al.*, SyLVaaS: System level formal verification as a service, Fund Inf 149(1–2) (2016).
[24] T. Mancini, *et al.*, Any-horizon uniform random sampling and enumeration of constrained scenarios for simulation-based formal verification, IEEE TSE (2021). To appear.
[25] T. Mancini, On minimising the maximum expected verification time, Inf Proc Lett 122 (2017).
[26] E. Tronci, *et al.*, Patient-specific models from inter-patient biological models and clinical records, in: FMCAD 2014, IEEE, 2014.
[27] T. Mancini, Computing biological model parameters by parallel statistical model checking, in: IWBBIO 2015, *LNCS* 9044, Springer, 2015.
[28] T. Mancini, *et al.*, Computing personalised treatments through in silico clinical trials. A case study on downregulation in assisted reproduction, in: RCRA 2018, *CEUR* 2271, 2018.
[29] S. Sinisi, *et al.*, Complete populations of virtual patients for in silico clinical trials, Bioinf 36 (2020).
[30] S. Sinisi, *et al.*, Optimal personalised treatment computation through in silico clinical trials on patient digital twins, Fund Inf 174(3–4) (2020).
[31] A. A. Altahir, *et al.*, Visual sensor placement based on risk maps, IEEE Trans Instr Meas 69 (2019).
[32] K. An, *et al.*, Reliable sensor location for object positioning and surveillance via trilateration, Transp Res Part B: Methodol 117 (2018).
[33] L. Dai, *et al.*, Sensor placement based on an improved genetic algorithm for connected confident information coverage in an area with obstacles, in: LCN 2017, IEEE, 2017.
[34] Y.-G. Fu, J. Zhou, L. Deng, Surveillance of a 2D plane area with 3D deployed cameras, Sensors 14 (2014).
[35] Y. E. Osais, *et al.*, Directional sensor placement with optimal sensing range, field of view and orientation, Mob Netw Appl 15 (2010).
[36] X. Yang, *et al.*, Computer-aided optimization of surveillance cameras placement on construction sites, Comp-Aided Civil Infr Eng 33(12) (2018).
[37] J. Yang, *et al.*, A 3D sensing model and practical sensor placement based on coverage and cost evaluation, in: CYBER 2015, IEEE, 2015.
[38] A. Saad, *et al.*, Toward a realistic approach for the deployment of 3d wireless sensor networks, IEEE Trans Mob Comp (2020).
[39] S. Jun, *et al.*, Placing visual sensors using heuristic algorithms for bridge surveillance, Appl Sci 8 (2018).