

# Optimal Fault-Tolerant Relay Node Positioning in Critical Wireless Networks via Artificial Intelligence

Qian Matteo Chen <sup>1</sup>, Toni Mancini <sup>1</sup>, Igor Melatti <sup>1</sup>, Enrico Tronci <sup>1</sup>, and Alberto Finzi <sup>2</sup>

<sup>1</sup>Computer Science Dept., Sapienza University of Rome, Italy

<sup>2</sup>Dept. Electrical Engin. and Information Tech., University of Naples Federico II, Italy

## Abstract

Radio communication networks in *critical infrastructures* like airports are often *mission-critical*, and must be adequately protected from external electromagnetic interference and attacks. Given the long distances involved and the need to use low-powered WiFi signals, intermediate *relay nodes* must be deployed on the field to ensure multi-hop reliable communication, also guaranteeing a sufficient degree of fault-tolerance.

In this short paper, we introduce the problem and briefly review our recent approach to automatically synthesise a cost-optimal fault-tolerant relay network. Differently from ad-hoc or heuristic approaches, our tool-chain is *entirely* based on off-the-shelf Artificial Intelligence general-purpose reasoners, namely MILP, PB-SAT, and SMT/OMT solvers. We also show experimental results on large scenarios from the Leonardo da Vinci Airport in Rome, Italy.

## 1 Introduction

Radio communication networks in *critical infrastructures* (e.g., [4]) like airports (e.g., [22]) are crucial to ensure their safety. Being *mission-critical* (e.g., [29, 60]), such networks need adequate protection from attacks and external electromagnetic interferences. For example, in airports, communications between the air traffic control tower and vehicles, or service communication within the airport area use VHF/UHF radio (e.g., [61]). Thus, an electromagnetic attack preventing, e.g., communication between the air traffic control tower and aircraft would put the airport operations at risk (e.g., [16, 35]). As a consequence, such critical areas must be monitored to detect unauthorised electromagnetic emissions and localise their source. This is often done by deploying *radiogoniometers* on the field, along with a gateway gathering information from them. Since communication between the radiogoniometers and the gateway must *not* interfere with the radio communication used within the infrastructure under protection, unlicensed low-power WiFi is often used. In this case, the long distances involved (of the order of kilometres) imply the need of multi-hop communication through the use of relay nodes. In such scenarios, the reliability of the relay network becomes essential.

In this short paper, we introduce the problem and review our recently proposed approach [49, 14] for the computation of a *cost-optimal fault-tolerant deployment* of relay nodes in a large critical area, which exploits off-the-shelf Artificial Intelligence (AI) and Model Checking (MC) reasoners, namely: Mixed Integer Linear Programming (MILP) [28, 15], Pseudo-Boolean Satisfiability (PB-SAT) [9, 51], and Satisfiability Modulo Theory (SMT)/Optimisation Modulo Theory (OMT) [7] solvers.



## 2 Problem requirements

The installation of relay nodes in a large critical area like an airport (the *Monitored Area, MA*), entails the cost of the hardware (*e.g.*, antennas and amplifiers), and the cost their actual deployment. The latter may be quite relevant: for instance, deploying a relay node in an airport may require installing antennas within a ground movement area, and this might yield additional costs to neutralise any impact on the airport operations. In our setting, relay antennas are *directional* antennas, hence they are deployed in *arrays* mounted on poles. Each antenna array is positioned at a given elevation from the ground (the height of the poles). Antennas mounted on the same pole are pointed to different directions (among a user-specified finite set of possible directions).

Our problem amounts to decide *where* in the MA and *with which orientation* to deploy relay antennas in order to *satisfy* all given requirements *while* minimising the overall deployment and hardware cost.

The problem requirements are quite intricate, because several aspects must be taken into account. First of all, since the MA might have a size of several squared kilometres, different regions might have different *ground elevation*, and *obstacles* of various types (*e.g.*, terminals, hangars) might exist. Thus, radio visibility between points in the MA cannot be determined by simply considering their Euclidean distance, but obstacles and terrain orography should be considered as well. Starting from MA orography data from existing databases, we represent both the MA and obstacles as a set of *convex polyhedra* in a Cartesian 3D bounded space. The MA may also include *forbidden placement areas*, *i.e.*, areas (*e.g.*, runways, taxiways, holding areas) where no relay node can be placed. Also, *forbidden link areas* may exist, which represent regions of the MA that should not be traversed by radio transmission links (*e.g.*, to avoid interferences with other existing radio networks). Furthermore, the *cost of deploying an antenna array* is not constant throughout the MA, as it might depend on several parameters (*e.g.*, the accessibility of the chosen point for maintenance). We represent forbidden placement, forbidden link areas, and areas with different positioning costs using sets of *convex polyhedra* within our 3D Cartesian bounded space.

Additional requirements need to be suitably addressed to make our computed solution acceptable. In particular, *radio visibility requirements*, which define the conditions that must be met by the deployment of pairs of directional relay antennas for their network links to be of satisfactory quality, need some domain-specific reasoning: indeed, radio communication between two (oriented) antennas in points of the MA must be considered impossible/unsatisfactory not only if the 3D Euclidean distance between them is longer than a given threshold, but also if the presence of obstacles, buildings, or terrain elevation changes, makes their prospective radio link suffer from excessive interference. This latter step requires so called Fresnel-zone analysis [24].

The remaining requirements are as follows. *Fault tolerance* requirements ask that the network to be designed guarantees successful routing from each sensor to the gateway also in case of malfunctioning of at most  $F \geq 1$  relay nodes. This is addressed by computing a relay node placement defining at least  $F + 1$  *distinct routes* from each sensor to the gateway, with no distinct routes from the same sensor sharing a relay node. *Performance requirements* ask that each sensor-to-gateway route in the designed network consists of at most  $H \geq 1$  hops. Finally, *network capacity requirements* ask that the network is able to convey the needed traffic from all the sensors to the gateway. This means that there is an upper bound to the number of messages (from sensor nodes) that can be routed through a given radio link per second.

## 3 Computing optimal fault-tolerant placement of relay nodes

Given the intricate requirements outlined in Section 2, a possible approach to solve our problem is to develop a specialised ad-hoc algorithm. However, such an approach would be costly and error-prone, and proving optimality of the computed solution would be not easy. In [49, 14] we proposed a radically different approach in three steps, where all steps *extensively use standard AI reasoners to produce correct-by-construction outputs*. This keeps the code-base at a minimum, greatly reduces the cost for ad-hoc development and testing, and removes altogether the cost to obtain a proof of optimality of the final solution. The main steps of our algorithm are as follows.

**MA discretisation.** Starting from an input scenario (namely, the definition of the MA with associated polyhedra for terrain elevation, obstacles, and forbidden placement areas), our algorithm discretises the

MA into a number of cells (of user-specified size) where an array of relay antennas can indeed be deployed.

**Parallel computation of the radio visibility graph.** A parallel pre-processing step is then performed, in order to compute a data structure called *radio visibility graph*. Nodes of the graph define the set of possible placements of poles (*pole nodes*) within the cells of the MA computed in the previous step, as well as possible installations of directional antennas on such poles (*antenna nodes*). Undirected edges of the graph define the possibility to establish reliable communication links between two antenna nodes installed on different poles with any user-defined orientation (this includes Fresnel-zone analysis and avoidance of forbidden link areas via constraint-based over-approximation [23, 43]), and the wired communication links between each pole and locally-installed antennas. To compute the radio visibility graph, a set of helper computational geometry problems similar to those in [37] are automatically generated as MILPs, and solved in parallel on a High-Performance Computing (HPC) infrastructure.

**Generation and solving of the main optimisation problem.** Our main optimisation problem is automatically modelled as a 0/1 Linear Program (0/1-LP). Our developed tool performs suitable optimisations of the problem formulation along the lines of [11, 38]. The obtained 0/1-LP is encoded into the languages supported by several state-of-the-art general-purpose reasoners based on different paradigms, namely: MILP, PB-SAT, and SMT/OMT. Any such solver can then be used as a computational engine.

## 4 Experimental results and scalability of the approach

We experimented with several scenarios defining various areas of the Leonardo Da Vinci Airport in Rome, Italy, whose areas range from 10 000 m<sup>2</sup> to 50 km<sup>2</sup>. By varying the size of the cells, the maximum number of hops  $H$ , and the fault tolerance degree  $F$ , we generated 366 0/1-LPs, with 309 to about 6 million zero-one variables and 378 to about 8 million constraints.

Such 0/1-LPs were generated in up to 500 seconds each on an HPC infrastructure with 64 nodes, and have been solved using a portfolio of 8 state-of-the-art solvers of different kinds: MILP solvers IBM CPLEX [27], GLPK [2], and Gurobi [25]; PB-SAT solvers MiniSAT+ [19] (based on the well-known SAT solver MiniSAT [18], was one of the winners of the 2005 PB-SAT Evaluation [1]), NAPS [57], and Sat4j [32] (the last two won different tracks of the 2006 PB-SAT Competition [3]); SMT/OMT solvers Z3 [17] (which supports linear optimisation problems through  $\nu Z$  [10]) and OptiMathSat [58].

Our experiments show (see [14] for our full experimental campaign) that all solvers were able to solve real-world and hard instances of such an industry-relevant optimisation problem, consisting of a remarkably high number of constraints. In more details, MILP solvers were able to solve, within our time limit of 3 hours, the *greatest number* of and the *largest* instances (*i.e.*, instances with up to around 2 million constraints), with CPLEX clearly being the top-performer, followed by Gurobi and GLPK. PB-SAT solvers succeeded to solve instances with up to around half-a-million constraints (with Sat4j being able to solve some of the largest instances, MiniSat+ immediately following, and NAPS lagging behind). Noticeably, the performance of PB-SAT solvers are competitive to those of MILP solvers on unsatisfiable instances. OMT solvers were able to solve fewer of the largest instances, with Z3 slightly outperforming OptiMathSat. In general, OMT solvers seem to have harder time in finding an optimal solution on satisfiable instances, and generally show better performance on unsatisfiable instances.

## 5 Related work

Although node placement in wireless sensor networks is a well-studied problem (*e.g.*, the survey in [64]), none of the available methods focuses on finding a *cost-minimal* placement while at the same time *guaranteeing* network connectivity *notwithstanding* loss (faults) of up to a given number of relay nodes. For example, [55] presents methods for optimal and robust (with respect to throughput degradation) positioning of relay nodes. However, robustness in [55] is not the same as fault-tolerance in our setting. Optimal positioning of relay nodes in a WiFi network (our setting) has been studied also in [54], where, however, fault-tolerance (a key point in our setting), is not addressed. Other available approaches sacrifice optimality and *a priori* guarantees, and perform some sorts of random placements of relay nodes (see references in [64]) or employ heuristic methods (see, *e.g.*, [8, 26]).

Network traffic, congestion and interference have been analysed in [33] using MILP, while [34] focuses on the transmission capacity of ad-hoc networks. The relationship between energy consumption of network relay nodes and routing strategies is studied in [13] as a maximum flow problem, which is solved with a combination of MILP and heuristic approaches. In our setting, we do not have energy problems, since relays are connected to the electrical grid. Conversely, our main focus is to find a positioning of both the relay nodes *and* antennas with minimal cost. Modelling *obstacles* for radio communication in a real environment has been investigated in [30], and a 2D model for directional antennas is presented in [20]. Much of the current research has focused on routing through relay nodes with a *known* given position (see, *e.g.*, [5, 54, 8, 52]), whereas our setting is different, in that we have to compute at the same time the position of the relay nodes *as well as* the routing strategy for each relay node. Finally, differently from other works, our solution also defines the orientation of the antennas hosted by each relay node.

## 6 Conclusions

In this short paper we introduced the problem of computing a deployment for relay nodes in a wireless network that minimises the relay node network cost while at the same time guaranteeing proper working of the network even when some of the relay nodes (up to a given maximum number) become faulty (*fault-tolerance*). For such a problem, we surveyed our recent approach [49, 14] which, in a few minutes of computation on a small parallel infrastructure, produces a 0/1-LP formulation of an optimisation problem that can be straightforwardly given as input to different classes of standard AI solvers, in particular: MILP, PB-SAT and SMT/OMT. Our experimental results using 8 among the state-of-the-art MILP, PB-SAT and SMT/OMT solvers show that we can compute in a few hours (hence, within a fully acceptable time given our application domain) an optimal placement of relay nodes in areas of up to 50 squared kilometres (*i.e.*, the area of a medium-large airport) with a precision of a few hundred metres.

In future work we aim at experimenting with additional solving technologies like, *e.g.*, Constraint Programming (see, *e.g.*, [6]), Constraint Answer Set Programming (see, *e.g.*, [53]) as well as hybrid (*e.g.*, [31]) and decomposition-based approaches (*e.g.*, [56]), also by exploiting our experience in other mission-critical application domains, *e.g.*, smart grids (see, *e.g.*, [40, 50]) and network vulnerability analysis [48]. Also, in order to further increase the size of the areas we can manage, we plan to approach our problem with iterative improvement techniques like local search and meta-heuristic methods (see, *e.g.*, [21, 63]) specially suited for large instances (*e.g.*, [39]), of course sacrificing optimality guarantees. Finally, we aim at reducing the need of linear constraint over-approximations by exploiting our simulation-based approaches (*e.g.*, [62, 41, 46, 36]) driven by intelligent search (*e.g.*, [42, 45, 44, 47, 12, 59]).

**Acknowledgements** This work was partially supported by: Italian Ministry of University and Research under grant “Dipartimenti di eccellenza 2018–2022” of the Department of Computer Science of Sapienza University of Rome; INdAM “GNCS Project 2019”.

## References

- [1] Pseudo-boolean competition 2005, 2005. <http://www.cril.univ-artois.fr/PB05>.
- [2] GLPK (GNU linear programming kit), 2012. <http://www.gnu.org/software/glpk>.
- [3] Pseudo-boolean competition 2016, 2016. <http://www.cril.univ-artois.fr/PB16>.
- [4] C. Alcaraz and S. Zeadally. Critical infrastructure protection: Requirements and challenges for the 21st century. *Int J Crit Infr Prot*, 8, 2015.
- [5] A. Amis, *et al.* Max-Min D-cluster formation in wireless ad hoc networks. In *IEEE INFOCOM 2000*
- [6] K. Apt. *Principles of Constraint Programming*. Cambridge U.P., 2003.
- [7] C. Barrett and C. Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, 2018.
- [8] E. Biagioni and G. Sasaki. Wireless sensor placement for reliable and efficient data collection. In *HICSS-36*. IEEE, 2003.

- [9] A. Biere, *et al.*, eds. *Handbook of Satisfiability*, vol. 185. IOS, 2009.
- [10] N. Bjørner, *et al.*  $\nu Z$  – An optimizing SMT solver. In *TACAS 2015, LNCS*, vol. 9035. Springer, 2015.
- [11] L. Bordeaux, *et al.* A unifying framework for structural properties of CSPs: Definitions, complexity, tractability. *JAIR*, 32, 2008.
- [12] A. Calabrese, *et al.* Generating T1DM virtual patients for in silico clinical trials via AI-guided statistical model checking. In *RCRA 2019, CEUR W.P.*, vol. 2538. CEUR, 2019.
- [13] M. Cardei and D. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Netw*, 11(3), 2005.
- [14] Q. Chen, *et al.* MILP, pseudo-boolean, and OMT solvers for optimal fault-tolerant placements of relay nodes in mission critical wireless networks. *Fundam Inform*, 2020.
- [15] M. Conforti, *et al.* *Integer Programming*. Springer, 2014.
- [16] A. Davis and H. Chang. Airport protection using wireless sensor networks. In *IEEE HST 2012*. IEEE, 2012.
- [17] L. de Moura and N. Bjørner. Z3: An efficient SMT solver. In *TACAS 2008*.
- [18] N. Eén and N. Sörensson. An extensible SAT-solver. In *SAT 2003, LNCS*, vol. 2919. Springer, 2004.
- [19] N. Eén and N. Sörensson. Translating pseudo-boolean constraints into SAT. *JSAT*, 2, 2006.
- [20] D. Ganesan, *et al.* Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. *ACM TOSN*, 2(2), 2006.
- [21] F. Glover and G. Kochenberger, eds. *Handbook of Metaheuristics*, vol. 57. Springer, 2006.
- [22] K. Gopalakrishnan, *et al.* Cyber security for airports. *Int J Traffic Transp Engin*, 3(4), 2013.
- [23] G. Gottlob, *et al.* Conditional constraint satisfaction: Logical foundations and complexity. In *IJCAI 2007*. 2007.
- [24] Y. Guo and S. Barton. *Fresnel Zone Antennas*. Springer, 2002.
- [25] Gurobi optimizer, 2018. <http://www.gurobi.com>.
- [26] A. Hashim, *et al.* Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm. *J Netw Comp Appl*, 64, 2016.
- [27] IBM ILOG CPLEX Optimization Studio V12.8.0, 2018.
- [28] J. Karlof. *Integer programming: theory and practice*. CRC, 2005.
- [29] Y. Kim, *et al.* Resilient end-to-end message protection for large-scale cyber-physical system communications. In *SmartGridComm 2012*. IEEE, 2012.
- [30] A. Krause, *et al.* Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *IPSN 2006*. ACM, 2006.
- [31] P. Laborie. An update on the comparison of MIP, CP and hybrid approaches for mixed resource allocation and scheduling. In *CPAIOR 2018, LNCS*, vol. 10848. Springer, 2018.
- [32] D. Le Berre and A. Parrain. The Sat4j library, release 2.2. *JSAT*, 7, 2010.
- [33] Y. Lee, *et al.* Optimization of AP placement and channel assignment in wireless LANs. In *LCN 2002*.
- [34] J. Li, *et al.* Capacity of ad hoc wireless networks. In *MOBICOM 2001*. ACM, 2001.
- [35] G. Lykou, *et al.* Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies. *Sensors*, 20(12), 2020.
- [36] F. Maggioli, *et al.* SBML2Modelica: Integrating biochemical models within open-standard simulation ecosystems. *Bioinformatics*, 36(7), 2020.
- [37] T. Mancini. Now or Never: Negotiating efficiently with unknown or untrusted counterparts. *Fundam Inform*, 149(1-2), 2016.
- [38] T. Mancini, *et al.* Evaluating ASP and commercial solvers on the CSPLib. *Constraints*, 13(4), 2008.

- [39] T. Mancini, *et al.* Combinatorial problem solving over relational databases: View synthesis through constraint-based local search. In *SAC 2012*. ACM, 2012.
- [40] T. Mancini, *et al.* Demand-aware price policy synthesis and verification services for smart grids. In *SmartGridComm 2014*. IEEE, 2014.
- [41] T. Mancini, *et al.* Computing biological model parameters by parallel statistical model checking. In *IWBIO 2015, LNCS*, vol. 9044. Springer, 2015.
- [42] T. Mancini, *et al.* SyLVaaS: System level formal verification as a service. In *PDP 2015*. IEEE, 2015.
- [43] T. Mancini, *et al.* User flexibility aware price policy synthesis for smart grids. In *DSD 2015*. IEEE, 2015.
- [44] T. Mancini, *et al.* Anytime system level verification via parallel random exhaustive hardware in the loop simulation. *Microprocessors and Microsystems*, 41, 2016.
- [45] T. Mancini, *et al.* SyLVaaS: System level formal verification as a service. *Fundam Inform*, 1–2, 2016.
- [46] T. Mancini, *et al.* On minimising the maximum expected verification time. *Inf Proc Lett*, 122, 2017.
- [47] T. Mancini, *et al.* Computing personalised treatments through in silico clinical trials. A case study on downregulation in assisted reproduction. In *RCRA 2018, CEUR W.P.*, vol. 2271. CEUR, 2018.
- [48] T. Mancini, *et al.* An efficient algorithm for network vulnerability analysis under malicious attacks. In *ISMIS 2018*. Springer, 2018.
- [49] T. Mancini, *et al.* Optimal fault-tolerant placement of relay nodes in a mission critical wireless network. In *RCRA 2018, CEUR W.P.*, vol. 2271. CEUR, 2018.
- [50] T. Mancini, *et al.* Parallel statistical model checking for safety verification in smart grids. In *SmartGridComm 2018*. IEEE, 2018.
- [51] J. Marques-Silva and S. Malik. Propositional SAT solving. In *Handbook of Model Checking*, 2018.
- [52] S. Meguerdichian, *et al.* Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM 2001*.
- [53] V. Mellarkod, *et al.* Integrating answer set programming and constraint logic programming. *Ann Math Artif Intell*, 53(1–4), 2008.
- [54] R. Prasad and H. Wu. Gateway deployment optimization in cellular Wi-Fi mesh networks. *J Netw*, 1(3), 2006.
- [55] L. Qiu, *et al.* Optimizing the placement of integration points in multi-hop wireless networks. In *ICNP 2004*, vol. 4. IEEE, 2004.
- [56] R. Rahmaniani, *et al.* The Benders decomposition algorithm: A literature review. *Eur J Op Res*, 259(3), 2017.
- [57] M. Sakai and H. Nabeshima. Construction of an ROBDD for a PB-constraint in band form and related techniques for PB-solvers. *IEICE Trans Inf Syst*, E98-D, No. 6, 2015.
- [58] R. Sebastiani and P. Trentin. OptiMathSAT: A tool for Optimization Modulo Theories. *JSAT*, 2018.
- [59] S. Sinisi, *et al.* Optimal personalised treatment computation through in silico clinical trials on patient digital twins. *Fundam Inform*, 2020.
- [60] O. N. Skrypnik. *Radio Navigation Systems for Airports and Airways*. Springer, 2019.
- [61] D. Stacey. *Aeronautical Radio Communication Systems and Networks*. Wiley, 2008.
- [62] E. Tronci, *et al.* Patient-specific models from inter-patient biological models and clinical records. In *FMCAD 2014*. IEEE, 2014.
- [63] P. Van Hentenryck and L. Michel. *Constraint-Based Local Search*. MIT, 2009.
- [64] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Netw*, 6(4), 2008.