Solving Sensor Placement Problems In Real Water Distribution Networks Using Adiabatic Quantum Computation

Stefano Speziali[(1)*], Federico Bianchi[(1)*], Andrea Marini[(1)], Lorenzo Menculini[(1)], Massimiliano Proietti[(1)], Loris F. Termite[(2)], Alberto Garinei[(3,1)], Marcello Marconi[(3,1)], and Andrea Delogu[(4)]

(1) Idea-re S.r.l., Perugia, Italy; (2) K-Digitale S.r.l., Perugia, Italy; (3) Department of Engineering Sciences, Guglielmo Marconi University, Rome, Italy; (4) BlueGold S.r.l., Milan, Italy; * Corresponding Author

Based on arXiv:2108.04075

INTRODUCTION

Water leaks in Water Distribution Networks (WDNs) can be cause of significant economic loss, besides being waste of important resources. Asset management of WDNs is indeed a relevant issue for the scientific community and novel and more efficient solutions to detect and isolate leaks are always needed. In particular, the optimal placement of sensors is crucial if we want to monitor the behavior of a WDN and prevent fault events. It is often the case that the number of nodes in a WDN is much larger than the number of available sensors. While the former can be in the order of thousands for a realistic WDN, there is usually just a few dozens of the latter. Hence, sensors must be placed such that network-wide global relevant information can be provided.

In this work, we formulate the problem of sensor placement as a combinatorial optimization problem. Roughly speaking, combinatorial optimization refers to the computation of maxima or minima of a function over a discrete domain. Many of these problems can be addressed by means of a new computational technique, known as Ising machine. In order to employ Ising machines to solve an optimization problem, one should define the energy function (Hamiltonian) of the Ising model or QUBO (Quadratic Unconstrained Binary Optimization) problem which corresponds to the function we want to minimize (or maximize).

We propose an instance of Hamiltonian whose ground state encodes the optimal sensor placement for a generic WDN. The optimal solution is then found by means of Simulated Annealing (SA) or Adiabatic Quantum Computation (AQC), exploiting the Hybrid D-Wave solver. In order to program the optimization, we used PyQUBO, an open-source Python library from D-Wave useful to construct QUBOs from the objective functions and constraints of optimization problems.

(1) FORMULATION OF THE HAMILTONIAN

A WDN can be formally interpreted as a graph $G = (\mathcal{V}, E)$, where the nodes (or vertices), denoted collectively as $\mathcal{V}$ are either tanks or junctions (the former are source of water while the latter distribute the existing water flow to users through the pipes), whereas the edges, denoted collectively E, are the pipes connecting nodes.

Let x_i be a binary variable associated with the i-th vertex $\mathcal{V}$. We choose x_i to be 1 if the node of our WDN associated with the i-th vertex hosts a sensor, 0 otherwise.

The optimal sensor placement can be formulated as a QUBO problem where the defining hamiltonian is given by:

$$H_P = H_{(0)} + H_{(1)} + H_{(2)} \qquad (1)$$

$$H_{(0)} = A \sum_{(ij)\in E} w_{ij}(1 - x_i)(1 - x_j)$$

The hamiltonian $H_{(0)}$ corresponds to the hamiltonian of the well-known *minimum vertex cover* problem: it encodes the constraint that every edge $(ij) \in E$ of the graph has at least one vertex associated with a sensor. In fact, in realistic situations, not all edges of our network can be associated with a sensor. To make the problem more consistent, we weigh edges according to their intrinsic properties in the WDN: w_{ij} is a (positive) weight for the edge $(ij) \in E$, which we identify as the **Edge Betweenness Centrality**. The definition of edge betweenness for an edge e is:

$$C_e^B = \sum_{s\neq t\in\mathcal{V}} \frac{\sigma_{s,t}(e)}{\sigma_{s,t}}$$

where $\sigma_{s,t}$ is the number of shortest paths between vertices s and t, while $\sigma_{s,t}(e)$ is the number of shortest paths between s and t passing through e. The edge betweenness can be used as a parameter to determine the relevance of each edge into the network. To take into account the presence of water tanks, we created a network of n_f fictitious nodes around and connected only to source nodes, with n_f equal to (the integer part of) n/n_s, where n is the total number of nodes in the network and n_s is the number of source nodes. In this way, source nodes behave like hubs for water supply and the centrality metric is weighted accordingly.

$$H_{(1)} = \sum_{i\in\mathcal{V}} c_i x_i$$

The hamiltonian $H_{(1)}$ serves to minimize the number of vertices with an associated sensor. In the most general setup, each node $i \in \mathcal{V}$ comes with an associated cost c_i, which we always assume to be non-negative ($c_i \geq 0$). To represent the cost of a node, we identified two useful parameters: the degree of accessibility of the network in correspondence of the node and the water need associated to the given node. The former is a useful parameter to consider when physically installing sensors, while the latter guarantees that we pick in the optimization nodes with higher demands of water. Thus, we propose to define c_i as

$$c_i = Cf_i + Dg_i$$

where $f_i \equiv f(\hat{v}_i)$ is a function of the water need at each node i, while g_i weights different nodes differently, according to their degree of accessibility. C and D are positive weights that can be tuned arbitrarily, but that should not exceed B, in order not to violate the constraint in $H_{(2)}$.

$$H_{(2)} = B\left(\sum_{i\in\mathcal{V}} x_i - s\right)^2$$

The hamiltonian $H_{(2)}$ is needed in realistic situations, in which the number of available sensors is fixed and, often, much smaller than the number of nodes. B is a positive number, chosen so that the constraint on the fixed number of sensors is always satisfied.

(2) CASE STUDY: A REAL WDN

We solved the sensor placement problem for the case of a real WDN. The WDN we use for the simulations is depicted in Fig. 1 and corresponds to a real water network in the Lombardy region in Italy. It comprises 1368 nodes (grey points), two of which are associated with a tank (blue crosses) and 1391 edges (solid lines connecting nodes). The WDN has a total length of approximately 26 kilometers and serves about 4000 people.

The weighted edge betweenness - using pipe lengths as weights - has been normalized to lie in the range 0 to 1. Fig. 1 and 2 clearly show that the most central edges (dark green) are those in proximity of the two tanks and where water is more likely to pass through to reach demand nodes, as we expected. Here, we report the results of a run with hyperparameters $(A, B, C, D) = (1,30,5,1)$. Water consumption at each node corresponds to water need at each node over the period of one year. We have also chosen to set $g_i = 0$ when the i-th node is considered easily accessible and 1 otherwise.

(3) SIMULATIONS AND RESULTS

We have used the Hamiltonian (1) with a number of sensors fixed at 48 units. The minimization is carried out performing 100 runs of the **simulated annealing** algorithm, for the set of hyperparameters described in section (2). Each run comes with an associated energy, each supposedly close to the global minimum, and the final sensor placement corresponds to the best result (minimum energy) among the 100 runs. We have found that 35 out of 48 sensors are to be placed in correspondence of nodes classified as the most accessible, while the percentage of water flowing through the selected nodes is about 4.3% of the total flow. The best solution is found when the energy is (approximately) $H_P = 341.8$. The minimization of the Hamiltonian takes approximately 100 seconds on a standard laptop equipped with a CPU Intel i7. The results are reported in Fig.1.

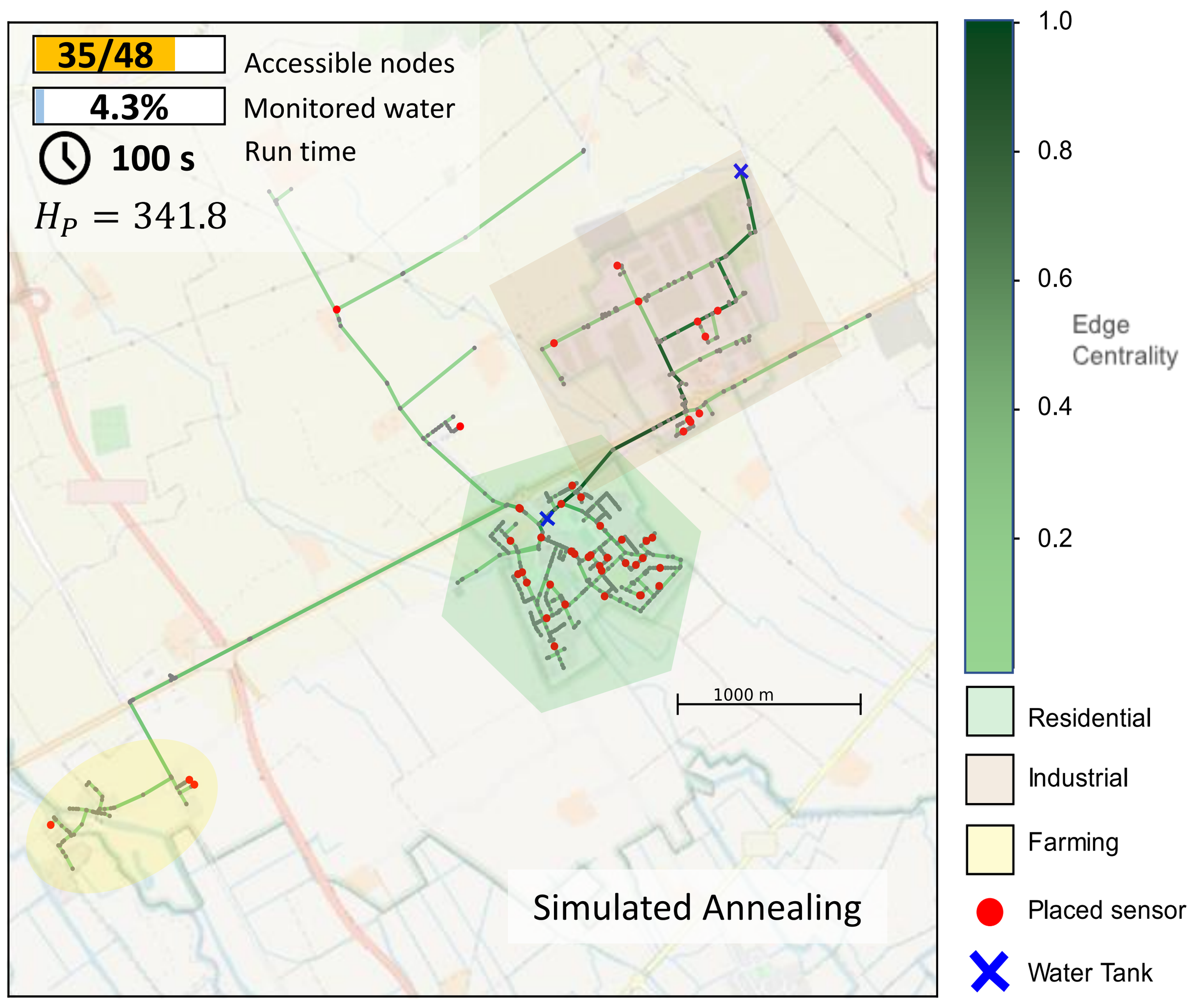

Fig. 1: best result after 100 runs of the **simulated annealing** algorithm. The edges of the WDN are plotted in shades of green, according to the edge betweenness centrality. The grey dots represent the nodes, while the red, bigger dots represent the nodes covered by a sensor. The two water tanks are shown in blue crosses. In this Network, three main zones can be found: a residential zone in green, an industrial zone in brown, and a mainly agricultural one, in yellow. On the upper-left corner the main results are summarized for this simulation.

As for the hybrid quantum-classical annealing, we have run our algorithm on the **D-Wave Leap hybrid solver service (HSS)**, where the only tunable parameter is the Computation Time. We have found that the optimal solution for the minimization problem (1) with the same set of hyperparameters as for simulated annealing case, is achieved when the run-time is set at 100 seconds (similar to Simulated Annealing). The best solution is attained when the energy is approximately $H_P = 304.8$, lower than in the classical case. We have also found that the number of selected nodes labeled as more accessible is of 39 out of 48 and percentage of water flowing through the selected notes is about 7.4% of the total flow. The results are reported in Fig.2

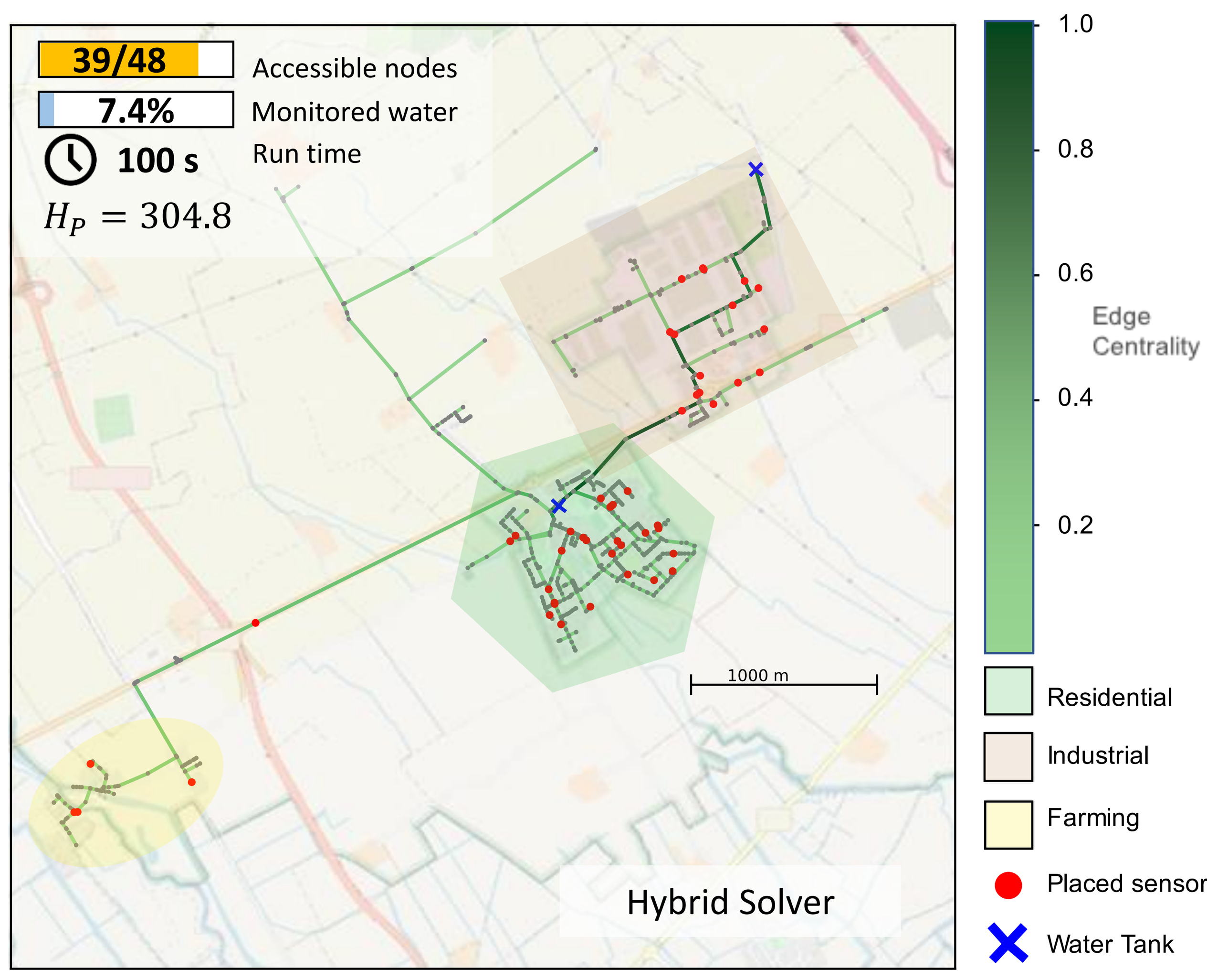

Fig. 2: best result after a 100 seconds run of the **Hybrid Solver** algorithm. The edges of the WDN are plotted in shades of green, according to the edge betweenness centrality. The grey dots represent the nodes, while the red, bigger dots represent the nodes covered by a sensor. The two water tanks are shown in blue crosses. In this Network, three main zones can be found: a residential zone in green, an industrial zone in brown, and a mainly agricultural one, in yellow. On the upper-left corner the main results are summarized for this simulation.

CONCLUSIONS

We considered a new heuristic method for the sensor placement problem on a real WDN which can be run on a classical computer by means of Simulated Annealing or on the publicly available D-Wave quantum solver. With the advent of Quantum Computing, we believe our work can be inspirational for other engineering applications.

The study presented here is part of the project "WATER A.I. – empower the efficiency of WATER networks through Artificial Intelligence for IoT" financed to BlueGold S.r.l by BANDO INNODRIVER S3 Azione I.1.b.1.1– sostegno all'acquisto di servizi per l'innovazione tecnologica, strategica, organizzativa e commerciale delle imprese, ID Progetto 1734422 — CUP E47B20000590007.