10/11/2024 02:54

(Article begins on next page)

Towards the fast and robust optimal design of Wireless Body Area Networks[☆]

Fabio D'Andreagiovanni[a,b,c,d,*], Antonella Nardin[e]

[a] *Department of Mathematical Optimization, Zuse Institute Berlin (ZIB), Takustr. 7, 14195 Berlin, Germany*
[b] *DFG Research Center MATHEON, Technical University Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany*
[c] *Einstein Center for Mathematics Berlin (ECMath), Straße des 17. Juni 135, 10623 Berlin, Germany*
[d] *Institute for Systems Analysis and Computer Science, Consiglio Nazionale delle Ricerche (IASI-CNR), Via dei Taurini 19, 00185 Roma, Italy*
[e] *Università degli Studi Roma Tre, Via Ostiense 169, 00154 Roma, Italy*

Abstract

Wireless body area networks are wireless sensor networks whose adoption has recently emerged and spread in important healthcare applications, such as the remote monitoring of health conditions of patients. A major issue associated with the deployment of such networks is represented by energy consumption: in general, the batteries of the sensors cannot be easily replaced and recharged, so containing the usage of energy by a rational design of the network and of the routing is crucial. Another issue is represented by traffic uncertainty: body sensors may produce data at a variable rate that is not exactly known in advance, for example because the generation of data is event-driven. Neglecting traffic uncertainty may lead to wrong design and

---

[*]Corresponding author
  Email addresses: d.andreagiovanni@zib.de (Fabio D'Andreagiovanni),
anto.nardin.mm@gmail.com (Antonella Nardin)
  URL: http://www.dis.uniroma1.it/~fdag/ (Fabio D'Andreagiovanni)

routing decisions, which may compromise the functionality of the network and have very bad effects on the health of the patients. In order to address these issues, in this work we propose the first robust optimization model for jointly optimizing the topology and the routing in body area networks under traffic uncertainty. Since the problem may result challenging even for a state-of-the-art optimization solver, we propose an original optimization algorithm that exploits suitable linear relaxations to guide a randomized fixing of the variables, supported by an exact large variable neighborhood search. Experiments on realistic instances indicate that our algorithm performs better than a state-of-the-art solver, fast producing solutions associated with improved optimality gaps.

Keywords: Body Area Networks, Wireless Sensor Networks, Network Design, Integer Linear Programming, Robust Optimization, Traffic Uncertainty, Metaheuristics, Exact Large Variable Neighborhood Search.

1. Introduction

In the last decade, wireless sensor networks have attracted a lot of attention and have represented a major research topic both from a theoretical and an applied point of view. Real-life applications of such networks have greatly enlarged and now span from environmental monitoring to preventive machinery maintenance and from security applications to intelligent building management, just to make a very few examples.

A *Wireless Sensor Network* (WSN) is a network of small portable wireless devices, the *sensors*, which are distributed over an area to (cooperatively) collect data about some phenomena and then forward the data through wireless links to one or more collectors, commonly called *sinks*. The sinks can store the data or transmit them to other networks, where the data are then processed. For an exhaustive introduction to theory and applications of WSNs, we refer the reader to the survey [36].

Thanks to recent technology advances in the field of portable medical sensors, WSNs have been also adopted in healthcare applications: besides the remote monitoring of health conditions of hospital inmates, WSNs can also support remote assistance of patients and can be used for evaluating sport training or conducting large in-field medical studies (see [28] for an overview).

In this work, we focus attention on a topic related to healthcare applications of WSNs: the design of body area networks. A *Body Area Network*
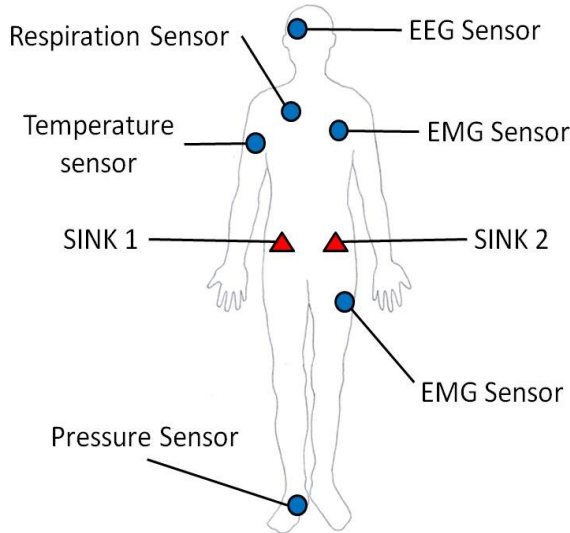
Figure 1: An example of BAN - the circles represent biosensors and the triangles represent sinks

(BAN) (also called body sensor network or wireless body area sensor network) is a WSN where a number of wireless sensors (*biosensors*) are placed over or inside the body of a person to collect biomedical data. The biosensors transmit the data to one or more sinks to be stored and/or processed and/or be transmitted to another network. An example of BAN is depicted in Figure 1. Designing a BAN essentially consists in deciding the topology of the network and how data are routed from the biosensors to the sinks.

BANs are not confined just to healthcare applications, but are used also in other contexts, such as sport and military training and interactive gaming. For a detailed introduction to BANs, we refer the reader to the survey [14]. Here we recall major features of a BAN that are relevant to our study. Though being a WSN, the design and the management of a BAN pose specific challenges coming from the use of sensors on the human body and require solutions and approaches that are sensibly different with respect to the traditional ones used for general WSNs [14]. A major question in a BAN is represented by propagation conditions of wireless signals along and through the human body, which are very peculiar and associated with high propagations losses (see [11] for a detailed discussion). High propagation

3

losses could in principle be tackled just by increasing the power emitted by the transmitting devices, like usually done in the design of wireless networks [16]. However, this cannot be done in a BAN for two main reasons:

- power emissions must be kept low to strictly respect *Specific Absorption Rates* (SARs) imposed by health regulatory bodies, to avoid damages to human tissues caused by radio signals and related overheating (see, e.g., [34]);

- higher power emissions imply higher energy consumption, which in turn leads to shorter lifetime of the batteries in BAN devices. Preserving the charge of batteries is a crucial objective in BANs, since the batteries are not easily replaceable and rechargeable as this negatively impacts on the comfort of patients (in the case of in-body sensors, for example, a surgical operation would be required) [11, 14].

These two points highlight a major challenge in the design of a BAN: minimizing the total energy consumed for wireless transmission and sensor functioning. We note that this challenge is also peculiar to the design of other WSNs, but in the case of BANs is really crucial.

In order to reduce energy consumption, a common solution in BANs and generally in WSNs is to introduce *relays*, which are wireless devices acting as intermediate nodes between the sinks and the sensors. The relays support transmissions over shorter distances and thus reduce energy consumption. As pointed out in many publications, such as [11, 25, 24, 23], relays allow to greatly overcome the energy inefficiency of single-hop routing, where all sensors directly transmit to the sink, and of multi-hop routing, where some sensors act as hotspots receiving and retransmitting data. We refer the reader to Figures 2 and 3 for a visualization of the differences between the three types of routing.

Though there is a conspicuous technical literature about BANs, concerning in particular the characterization of propagation conditions and the definition of routing protocols (see, for example, [14, 24, 11]), the problem of optimally designing a BAN through the solution of an optimization model has been practically neglected. To the best of our knowledge, [23] is the only work that has used mathematical optimization to model and solve a BAN design problem. This fact is stressed by the same author of [23], where the design problem is formulated as a mixed integer linear program where multiple-path routing and relay placement is established in order to minimize the total cost of deployment of a BAN. The resulting problem is solved by direct application of a commercial *Mixed Integer Linear Programming*
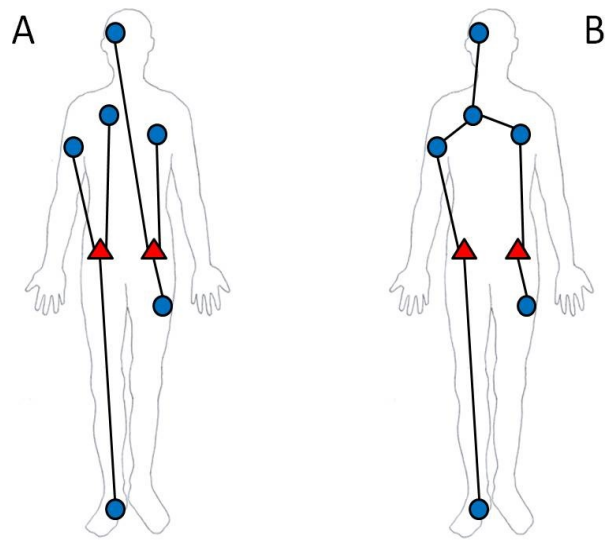
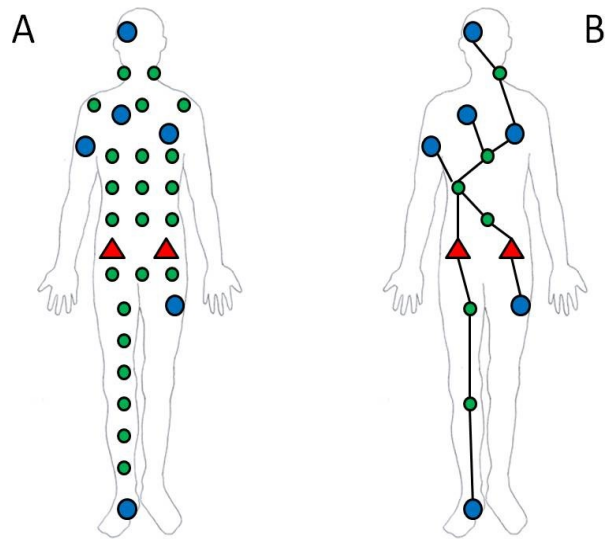Figure 2: Visualization of the same BAN using single-hop routing (A) and multi-hop routing.

Figure 3: The figure shows the potential topology of a BAN using relay nodes, represented as circle of smaller sizes, and the realized topology, where just a subset of relays has been activated.

(MILP) solver and no new algorithm is proposed to specifically improve the performance of the solver. In contrast to [23], a critical objective of our work has been to develop a new solution algorithm for BAN design having a better performance than a state-of-the-art solver.

Within the limited work available about BAN design, we also stress that *traffic uncertainty*, that is the fact that the volume of data that must be routed through the network is in general not known a priori, has been totally neglected. However, this is a relevant topic, as pointed out in [14] and as we discuss in more detail in Section 3. Data uncertainty in BANs can be due, for example, to event-driven biosensors, which generate data only under some specific conditions, so that their data rate is not constant and not exactly known in advance [27]. If we neglect data uncertainty, we risk to produce solutions that are actually infeasible and thus totally useless in practice [6]: in the case of event-driven biosensors, if the actual data rates result higher than expected, the capacity of the BAN could be insufficient to manage the produced data and we could lose a part of them. This would evidently have dramatic effects for patients, when sensors produce vital data, such as a cardiac sensor installed for early detecting ischemia. The design under data uncertainty is also relevant for WSNs in general, as pointed out for example in [25], where a min-max regret optimization framework is adopted to protect the WSN design from variations in the expected data rate of sensors.

In this work, our main original contributions are:

1. the first mathematical programming model for the joint optimization of *single-path* routing and relay deployment in BAN design. This model adopts binary variables to represent single-path routing decision and installation of relays and corresponds to a binary linear program. Our model better highlights the strong connections of the BAN design problem with classical capacitated network design problems dealing with flow optimization and capacity installation;

2. the first *robust optimization* approach for dealing with traffic uncertainty in BAN design. Specifically, we adopt a *scenario-based min-max* approach (see [1, 6]), where we aim to identify the solution having the best performance in the worst-case scenario. We think that this is a particularly appropriate form of robustness for BANs, since we are dealing with biomedical data and is thus crucial to have strict guarantees about the robustness of design solutions;

3. an original optimization algorithm that is based on the combination of 1) a heuristic exploiting ideas from randomized algorithms [31] and

from *ANTS - Approximate Nondeterministic Tree-Search* - [29], a refinement of classical *ant colony optimization* [22]; 2) an *exact* large variable neighborhood search that is formulated as a binary linear program, which is solved exactly by using a state-of-the-art optimization solver. A peculiarity of our algorithm is to exploit the precious information coming from suitable linear relaxations of the considered BAN design problem. Such information can be used to guide the fixing of variables during the heuristic construction of feasible solutions. Furthermore, by using linear relaxations, we are able to derive an *optimality gap*, i.e. a quality measure of solutions that indicates how far the produced solutions are from the optimum. This of course constitutes a major advantage of our algorithm with respect to "simple" heuristics, which do not provide any quality guarantee. We decided to investigate the combination of a variant of ANTS and of an exact search on the basis of our direct and successful experience in the context of other telecommunication network design problems, namely the design of cooperative wireless networks [17] and the multiperiod design of fixed telecommunications networks involving flow optimization problems [18, 19];

4. computational experiments considering a set of realistic BAN instances, showing that our optimization algorithm can fast compute solutions of better quality and associated with better optimality gaps w.r.t. those produced by a state-of-the-art optimization solver.

The remainder of this paper is organized as follows: in Section 2, we introduce an optimization model for the design of BANs; in Section 3, we introduce the new formulation for the scenario-based robust optimization of BANs; in Sections 4 and 5, we present our original optimization algorithms and computational results.

2. The Body Area Network Design Problem

In this section, we first provide a description of the elements of a BAN that are relevant for our new mathematical optimization model, then we proceed to describe the model, pointing out its strong connection with flow optimization problems in networks.

2.1. System elements

For modeling purposes, a BAN can be essentially described as a *set of biosensors B* that generate biomedical data intended to be collected by a *set of*

sinks S. The location of the biosensors and the sinks over/in the body is commonly pre-established. This is especially true for the biosensors, which must be usually positioned in specific body points for correctly measuring and generating the data. For each sink $s \in S$, each biosensor $b \in B$ generates a volume of data $d_{bs} \geq 0$, commonly expressed as a *bitrate* measured in bits per second (bit/s).

The biosensors transmit data to the sinks according to a multi-hop routing, which exploits a number of relays, i.e. devices whose task is to forward received data to another relay or to a sink. As pointed out in the previous section, the relays make the communications in the BANs more energy efficient. The network created by the relays can be actually viewed as a backbone network that transports the biomedical data from the biosensors to the sinks.

In contrast to the biosensors and the sinks, the location of the relay nodes can be chosen and a rational positioning of the relays can further improve the energy efficiency of the BAN. So a critical decision that we must take when designing a BAN is where to locate the relays: we are typically given a set of candidate locations and a maximum number of deployable relays and we must decide the number of deployed relays and their locations. We denote by R the set of *potentially deployable* relays and by $U > 0$ the maximum number of relays that can be deployed in the BAN. Each potential relay $r \in R$ is characterized by a *unique* position over/in the body and we must decide whether installing it or not. Additionally, each relay is characterized by a capacity $c_r \geq 0$ representing the maximum bitrate that it may handle.

The transmission of data from any BAN device (biosensor, relay node or sink) to another BAN device is based on a directional wireless link. Similarly to [23], we assume that the devices share the frequency band on which they operate according to a time division multiple access protocol that avoids interference. When any device transmits or receives data, it consumes energy and a crucial aim of our decision process is to design the BAN in order to minimize the total energy consumption so to increase the BAN lifetime. As in [11] and [24], we consider only transmitting and receiving energy consumption, which are dominant with respect to other forms of consumptions like sensing and processing. To express the total transmission energy E_{TX} and the total receiver energy E_{RX} (both expressed in joules) of any device of the BAN, we use the formulas proposed in [11]:

$$
\begin{aligned}
E_{\text{TX}}(v, \delta) &= E_{\text{TX}_{CIRC}} \cdot v + E_{\text{TX}_{AMP}}(\lambda) \cdot \delta^{\lambda} \cdot v \\
E_{\text{RX}}(v) &= E_{\text{RX}_{CIRC}} \cdot v
\end{aligned}
\tag{1}
$$

where we have highlighted the fact that E_{TX} and E_{RX} are functions of

the volume of transmitted/received data v (expressed in bits) and of the distance δ (expressed in meters) between the transmitter and the receiver. In the formulas: $E_{\text{TX}_{CIRC}}$ and $E_{\text{RX}_{CIRC}}$ are the energy consumed by the circuits to transmit and receive a single bit, respectively; λ is the path loss exponent of the path loss formula regulating wireless signal propagation (see [33] for a general introduction to signal propagation and losses in a real environment and [11] for a discussion about path losses for the specific case of BANs); $E_{\text{TX}_{AMP}}(\lambda)$ is the energy consumed by the transmitting amplifier and depends on the path loss coefficient. As in [24], we can assume that the devices operate power control and thus they consume minimal energy when transmitting to a receiver.

2.2. BAN Design as a Flow Optimization Problem

The optimal design of a BAN with relays can be naturally traced back to a flow optimization problem in a network, in particular a *Multicommodity Flow Problem* (MCFP): in an MCFP, we want to route a number of commodities through a network, while not exceeding the capacity of network elements and optimizing an objective function of the routing. We refer the reader to the book [3] for an exhaustive introduction to network optimization and MCFPs.

Since we are dealing with a network, it is natural to model the BAN by a *directed graph* $G(V, A)$ where:

- the set of *vertices* V contains one element for each wireless device (biosensor, relay or sink) of the network. The set V is thus the union of three disjoint sets of vertices:[1]

  1. the set B of vertices corresponding to biosensors;
  2. the set R of vertices corresponding to potentially deployable relays;
  3. the set S of vertices corresponding to sinks;

  Thus $V = B \cup R \cup S$. Each device can transmit to other devices that are positioned within its transmission range. The transmission range depends upon the propagation conditions and the power of the

---

[1] In order to be completely and rigorously formal, we should denote the three sets of vertices by V_B, V_R, V_S, distinguishing them from the corresponding sets of wireless devices B, R, S. However, this would complicate the notation and decrease readability. So we denote the (sets of) vertices by the symbols of the corresponding (sets of) devices of the wireless system.

transmitting device (see [11, 33]). We denote the subsets of devices that are within transmission range of any device as follows:

1. for each biosensor $b \in B$, we distinguish the subsets $R_b \subseteq R$, and $S_b \subseteq S$ representing the relays and sinks within the range of b, respectively;
2. for each potential relay $r \in R$, we distinguish the subsets $R_r \subseteq R$, and $S_r \subseteq S$ representing the relays and sinks within the range of r, respectively;
3. more generally, given a vertex $i \in V$, representing any type of BAN device, we denote by $V_i \subseteq V$ the subset of vertices representing devices within the transmission range of i.

Remark 1. We assume that each biosensor $b \in B$ *only generates and transmits* data and is never a receiver. So it is not necessary to characterize the subsets $B_r, B_s \subseteq B$ of biosensors within range of a relay r or a sink s. Analogously, we assume that each sink s *only receives* data and is never a transmitter. So it is not necessary to introduce the subsets $B_s \subseteq B$, $R_s \subseteq R$ of biosensors and relays within range of a sink s;

- the set of *arcs* A contains one element for each wireless link that can be established between a pair of wireless devices of the network. A link between two generic devices can be established if the receiving device is within the transmission range of the transmitting device.

A generic arc $a = (i, j) \in A$ is an ordered pair of vertices representing a directional wireless link from a device $i \in V$ to another device $j \in V_i$ within the range of i. We refer to vertices i, j of an arc $a = (i, j)$ as the tail and the head of arc a, respectively. The set A is the union of four disjoint sets of arcs:

1. the set $A_{B \to S}$ of arcs (i, j) such that the tail is a biosensor and the head is a sink within the range of the biosensor, i.e. $i \in B$, $j \in S_i$. They represent transmissions of biomedical data directly to sinks;
2. the set $A_{B \to R}$ of arcs (i, j) such that the tail is a biosensor and the head is a relay within the range of the biosensor, i.e. $i \in B$, $j \in R_i$. They represent transmissions from a biosensor to a relay;
3. the set $A_{R \leftrightarrow R}$ of arcs (i, j) such that both the tail and the head are relay nodes, i.e. $i, j \in R$ with $j \in R_i$. They represent wireless links between relay nodes;

4. the set $A_{R \to S}$ of arcs (i, j) such that the tail is a relay node and the head is a sink within the range of the relay, i.e. $i \in R$, $j \in S_i$. They represent transmissions from a relay to a sink;

Thus $A = A_{B \to S} \cup A_{B \to R} \cup A_{R \leftrightarrow R} \cup A_{R \to S}$.

Remark 2. Following the assumption that biosensors B only transmit data, there are no arcs in A that have biosensors as head (i.e., $\nexists (i, j) \in A : (i \in S \lor i \in R) \land j \in B_i$). In other words, *biosensors do not have ingoing arcs*. Analogously, following the assumption that sinks S only receive data, there are no arcs in A that have sinks as tail (i.e., $\nexists (i, j) \in A : i \in S \land (j \in B_i \lor j \in R_i)$). In other words, *sinks do not have outgoing arcs*. Additionally, we note that there are no arcs from biosensors to biosensors and from sinks to sinks.

The transmission of data over a link represented by an arc $a = (i, j) \in A$ implies consuming an amount of energy that is the sum of the energy consumed by i to transmit and the energy consumed by j to receive. We denote by $E_{ij} \geq 0$ the total amount of energy that is consumed to send one unit of data from i to j. This quantity can be computed by referring to the energy formulas (1) for the case of a single data unit transmitted (i.e., $v = 1$):

$$
\begin{aligned}
E_{ij} &= E_{\text{TX}}(1, \delta_{ij}) + E_{\text{RX}}(1) \\
&= \left[E_{\text{TX}_{CIRC}} + E_{\text{TX}_{AMP}}(\lambda_{ij}) \cdot \delta_{ij}^{\lambda_{ij}} \right] + E_{\text{RX}_{CIRC}}
\end{aligned}
\tag{2}
$$

where $\lambda_{ij}, \delta_{ij}$ are the path loss coefficient and the distance between i and j, respectively.

We can now provide a first informal statement of the *Body Area Network Design Problem* (BAND): given a BAN modeled by a graph $G(V, A)$ defined as above and the bitrate of each biosensor for each sink, we must decide which are the relays that are deployed in the BAN and how to route the data from the biosensors to the sinks through the deployed relays, so that the capacity of each relay is not exceeded and the total energy consumed for transmitting and receiving data is minimized. In a more formal way, we can state the BAND as follows.

Definition 1 (The Body Area Network Design Problem - BAND). Given:

- a BAN represented by a directed graph $G(V, A)$, where $V = B \cup R \cup S$ is the set of vertices and $A = A_{B \to S} \cup A_{B \to R} \cup A_{R \leftrightarrow R} \cup A_{R \to S}$ is the set of arcs;

- the bitrate $d_{bs} \geq 0$ of data generated by each biosensor $b \in B$ for each sink $s \in S$ - note that this actually corresponds to a commodity of a MCFP that must be routed from an origin node b to a destination node s in the network represented by the graph G(V,A);

- the capacity $c_r \geq 0$ of each relay $r \in R$;

- the energy coefficients $E_{ij} \geq 0$ expressing the total energy consumed to send 1 data unit from i to j;

the BAND consists in choosing which relays are activated and which single-paths are used to route the flow of data generated by each biosensor for each sink, in order to minimize the total energy consumption. $\qquad\square$

We note that the BAND constitutes a peculiar version of a *capacitated network design problem* involving an unsplittable MCFP problem (see [3, 19] for an introduction to capacitated network design) where: 1) the data of each couple (b, s) generated by a biosensor $b \in B$ for a sink $s \in S$ corresponds to a commodity; 2) commodities must be routed on a single path and cannot be split over a multiplicity of paths; 3) capacity installation decisions concern the vertices and not the arcs (deciding to deploy a relay is equivalent to get the corresponding vertex capacity - under the interference-free assumption made in the previous section, the capacity of the arcs representing wireless links is not relevant for the problem); 4) the objective function of the problem pursuing minimization of energy is a function of the routing decisions (routing flow on an arc corresponds to a transmission on a wireless link that make both the transmitter and the receiver consume energy).

We now proceed to derive an optimization model for the BAND.

2.3. A Binary Linear Program for the BAND

In the BAND, we must take two kind of decisions: 1) which are the single-paths used to route all data generated by each biosensor for each sink; 2) which are the deployed relays. These two decisions can be modelled by introducing two families of binary decision variables:

- binary *unsplittable flow variables* $x_{ij}^{bs} \in \{0, 1\}$ $\forall\, b \in B, s \in S, (i, j) \in A$ such that:

$$
x_{ij}^{bs} \;=\;
\begin{cases}
1 & \text{if all the data generated by biosensor } b \text{ for sink } s \\
 & \quad \text{are routed on arc } (i, j) \\
0 & \text{otherwise,}
\end{cases}
$$

- binary *relay deployment variables* $y_r \in \{0, 1\}$ $\forall\, r \in R$ such that:

$$y_r = \begin{cases} 1 & \text{if relay } r \text{ is deployed} \\ 0 & \text{otherwise} \end{cases}$$

Using these variables we then introduce several families of constraints that model the feasible solutions of the decision problem.

First of all, since we model transmissions through a flow problem on a graph, we must enforce flow conservation constraints. Assuming conventionally that outgoing flows of a vertex have negative value and ingoing flows have positive value, the flow conservation constraint of a generic node $i \in V$ can be written as:

$$\sum_{(j,i) \in A} f_{ji} - \sum_{(i,j) \in A} f_{ij} = \text{bal}_i$$

where f_{ji}, f_{ij} are the flows on a generic ingoing arc and outgoing arc of i, respectively, and bal_i is the flow balance of i. A vertex creating flow (source) has a negative balance $\text{bal}_i < 0$, whereas a vertex storing flow (sink) has a positive balance $\text{bal}_i > 0$. In a (transit) vertex that is neither a source nor a sink, the ingoing flow must equal the outgoing flow and the balance must be null (i.e., $\text{bal}_i = 0$).

In the case of the BAND, we distinguish three flow balance cases, one for each type of device/vertex: 1) biosensors $b \in B$, which only transmit data, have a negative flow balance; 2) relays $r \in R$, which are transit vertices and thus retransmit all the received data, have a null flow balance; 3) sinks $s \in S$, which only receive data, have a positive flow balance. In each of these vertex, the flow balance must be considered for the data flow generated by each biosensor $b \in B$ for each sink $s \in S$. The three corresponding families of flow conservations constraints are defined below.

For each biosensor $b \in B$, we have only outgoing arcs to relays and sinks and we must consider the flow balance for the data sent from b to each sink $s \in S$, which is equal to the bitrate d_{bs}, namely:

$$- \sum_{\substack{(b,j) \in \\ A_{B \to R} \cup A_{B \to S}}} d_{bs}\, x_{bj}^{bs} = -\, d_{bs} \qquad \forall\, b \in B, s \in S \tag{3}$$

Note that since we must route the entire data flow of b for s on a single path, the unsplittable flow variables x_{ij}^{bs} must be multiplied by the corresponding bitrate d_{bs}.

For each relay $r \in R$ and for each data flow from a biosensor $b \in B$ to each sink $s \in S$, the total ingoing and outgoing flow of r must be equal, so

we have:

$$\sum_{\substack{(j,r)\in \\ A_{B\to R}\cup A_{R\leftrightarrow R}}} d_{bs}\,x_{jr}^{bs} \;-\; \sum_{\substack{(r,j)\in \\ A_{R\leftrightarrow R}\cup A_{R\to S}}} d_{bs}\,x_{rj}^{bs} \;=\; 0 \qquad \forall\,b\in B, s\in S, r\in R \quad (4)$$

Finally, for each sink $s \in S$, the ingoing data flow from each biosensor $b \in B$ on each edge entering s must be equal to the data flow d_{bs}, so we have:

$$\sum_{\substack{(j,s)\in \\ A_{B\to S}\cup A_{R\to S}}} d_{bs}\,x_{js}^{bs} \;=\; d_{bs} \qquad \forall\,b\in B, s\in S \qquad (5)$$

After having introduced the flow conservation constraints, we need to introduce a family of constraints expressing the capacity of the relays:

$$\sum_{\substack{(r,j)\in \\ A_{R\leftrightarrow R}\cup A_{R\to S}}} d_{bs}\,x_{rj}^{bs} \;\leq\; c_r\,y_r \qquad (6)$$

Note that for each relay r, the constraint has a right-hand-side with variable value: if r is deployed, i.e. $y_r = 1$, then the constraints activates and the right-hand-side is equal to c_r. Otherwise, the right-hand-side is equal to 0 and forces to zero also the left-hand-side, thus preventing data flows to be received by or transmitted to r. Note that similar constraints must *not* be introduced for the biosensors and sinks, since we have assumed that the biosensors do not receive and the sinks do not transmit and that they have sufficient capacity to handle the traffic that they respectively transmit and receive.

It is practically reasonable to assume that the number of deployable relays in a BAN is limited by some value $U > 0$. This can be easily expressed by a constraint limiting above the sum of the activation variables:

$$\sum_{r\in R} y_r \leq U \qquad (7)$$

We complete the modeling of the BAND as an optimization problem, by introducing the following objective function that expresses the minimization of the total energy consumption:

$$\min \sum_{b\in B}\sum_{s\in S}\sum_{(i,j)\in A} E_{ij}\,d_{bs}\,x_{ij}^{bs} \qquad (8)$$

The function sums the energy consumed by each arc (i, j) when used for the transmission from i to j of the entire data flow d_{bs} sent from a biosensor b to

15

a sink s. This consumption is expressed as the product of the data flow and the energy E_{ij} consumed on (i, j) to transmit and receive 1 unit of data.

Summarizing what has been presented above, the BAND can be modelled by the following *Binary Linear Program* that we denote by BAND-BLP:

$$\min \sum_{b \in B} \sum_{s \in S} \sum_{(i,j) \in A} E_{ij} \, d_{bs} \, x_{ij}^{bs} \qquad \text{(BAND-BLP)}$$

$$- \sum_{\substack{(b,j) \in \\ A_{B \to R} \cup A_{B \to S}}} x_{bj}^{bs} = -1 \qquad\qquad b \in B, s \in S \qquad (9)$$

$$\sum_{\substack{(j,r) \in \\ A_{B \to R} \cup A_{R \leftrightarrow R}}} x_{jr}^{bs} - \sum_{\substack{(r,j) \in \\ A_{R \leftrightarrow R} \cup A_{R \to S}}} x_{rj}^{bs} = 0 \qquad b \in B, s \in S, r \in R \qquad (10)$$

$$\sum_{\substack{(j,s) \in \\ A_{B \to S} \cup A_{R \to S}}} x_{js}^{bs} = 1 \qquad\qquad b \in B, s \in S \qquad (11)$$

$$\sum_{\substack{(r,j) \in \\ A_{R \leftrightarrow R} \cup A_{R \to S}}} d_{bs} \, x_{rj}^{bs} \leq c_r \, y_r \qquad\qquad r \in R \qquad (12)$$

$$\sum_{r \in R} y_r \leq U \qquad\qquad (13)$$

$$x_{ij}^{bs} \in \{0, 1\} \qquad\qquad b \in B, s \in S, (i,j) \in A$$

$$y_r \in \{0, 1\} \qquad\qquad r \in R \, .$$

Remark 3. Note that we have simplified constraints (9-11) by dividing both sides of the inequalities by d_{bs}.

3. Robust Body Area Network Design

In the previous section, we have assumed that the data rate d_{bs} associated with each biosensor-sink couple (b, s) is precisely known when the optimization problem BAND-BLP is solved. However, this assumption may not be true in practice: data generation of sensors may be event-driven, so that the data rate is not constant and not known in advance [27]. To reduce energy consumption, some body sensors can be also configured to only generate data when an unusual situation arises, rather than to operate a continuous transmission of all readings. More in general, it is natural to assume that the data rate generated by wireless sensors is not exactly known, as also explained in [14] and [25].

If the actual data rate of a biosensor-sink couple is lower than expected, we are not really facing an issue: we have installed enough capacity to deal with a higher data rate, so the network will be able to deal with lower traffic volumes. On the other hand, we would have possibly designed a less costly and more energy efficient network, deploying a lower number of relays.

Troubles are instead going to arise if the actual data rates result higher than expected: deployed relays could indeed be insufficient to manage the higher rates and we could thus lose important biomedical data. This of course is a risk that we cannot run in BANs, especially in the case of biosensors that provide vital data, such as cardiac sensors installed for the early detection of ischemia.

The presence of data uncertainty in an optimization problem, namely the fact that a subset of the input data is not exactly known when the problem is solved, may result really tricky not just practically but also theoretically: as it is well-known from sensitivity analysis, even small deviations in the value of the input data may completely compromise the feasibility and optimality of produced solutions. Solutions supposed to be feasible may result infeasible and thus totally useless in practice, while solutions supposed to be optimal may result instead of mediocre quality. We refer the interested reader to [6, 7] for a thorough discussion about the issues associated with data uncertainty in optimization. Here we provide a simple example to clarify how infeasibility may arise in the BAND case.

Example 1 (Infeasible BAN design due to data rate variations).
Suppose to have a BAN including a relay node with a capacity of 250 kbit/s. Furthermore, suppose that the relay processes a total data flow that is produced by many biosensors and that is made up of two parts: a constant rate and exactly known part equal to 200 kbit/s and a variable rate part that may vary between 0 and 100 kbit/s (this could be for example produced by event-driven biosensors). When the variable rate exceeds 50 kbit/s, the capacity of the relay is exceeded and data are lost. We would thus have an infeasible design solutions that is not admissible in a BAN. It is thus necessary to deploy additional relays and/or modify the routing to guarantee protection from data uncertainty.

In order to tackle the issues coming from data uncertainty, over the years many methods have been proposed: after Dantzig's pioneering study about uncertain linear programs [20], many studies about optimization under uncertainty have been conducted and have in particular focused on Stochastic Programming (SO) methods, where it is in general necessary to characterize the probability distribution of the uncertain data and typically solve very

large-scale programs that produce *probabilistically* feasible and optimal solutions [35]. More recently, Robust Optimization (RO) has attracted a lot of attention as a very effective and efficient alternative to SO, especially due to its computational tractability and accessibility, pointed out in the seminal works about ellipsoidal uncertainty by Ben-Tal and Nemirovski (see e.g., [6]) and Γ-Robustness by Bertsimas and Sim [8]. RO produces so-called robust solutions that are *deterministically* feasible and optimal with respect to the worst case that uncertain data may assume in a domain specified by the decision maker. The reader is referred to [6, 7] for an exhaustive introduction to theory and applications of RO and for a comparison with SO pointing out determinant advantages of RO.

In this work, we adopt so-called *min-max robustness* (Min-Max) [1], also known as *absolute robustness* (e.g. [13]), a form of robust optimization that results particularly suitable in the case of critical applications where it is crucial to have high level of protection against data uncertainty. Recalling the example made before, about cardiac data that are collected by a body sensor for early ischemia detection, it is easy to understand that this approach looks really appropriate for an application like the BAND, where traffic volume uncertainty could even lead to the death of a non-correctly monitored patient. Another example of application is provided by the detection of contaminants in water networks (see [13]), where not having high level of protection against data uncertainty may have fatal consequences for large communities of people. Such kind of robustness is also appropriate when dealing with a very risk-averse decision maker, that requires full protection against simultaneous occurring of all the worst realization of each uncertain data of the decision problem (e.g., [4], which considers the unit commitment problem with uncertain market prices for a price-taker energy producer). In the context of the BAND, assuming the perspective of a highly risk-averse decision maker, who wants to guarantee a fully trustable monitoring of health conditions even under data uncertainty, looks appropriate.

To rigorously define the robust (Min-Max) version of an optimization problem, let us consider a generic uncertain binary linear program of the form:

$$V^* = \min \ c'x \qquad\qquad\qquad \text{(BLP)}$$
$$x \in \mathcal{F} = \left\{ \begin{array}{l} Ax \geq b \\ x \in \{0,1\}^n \end{array} \right\}$$

which consists in minimizing an objective function identified by a coefficient vector $c \in \mathbb{R}^n$ over a feasible set of solutions $\mathcal{F}$. The set $\mathcal{F}$ includes those

n-dimensional binary vectors x that satisfy a system of linear constraints identified by a coefficient matrix $A \in \mathbb{R}^{m \times n}$ and a right-hand side vector $b \in \mathbb{R}^m$. The problem BLP is uncertain in the sense that the values of the cost vector c and of the coefficient matrix A are just a reference and could vary in a way that is not known exactly when BLP is solved. However, we are able to identify a finite set Σ of possible data scenarios, such data each scenario $\sigma \in \Sigma$ specifies the entries of c_σ and A_σ representing a complete valorization of the uncertain cost vector c and of the coefficient matrix A.

Under these assumptions, if we denote by $c'_\sigma x$ the value of a feasible solution x in scenario σ, the Min-Max version of problem BLP for the set of scenarios Σ is the following:

$$V^{\mathrm{R}} = \min_x \max_\sigma c'_\sigma x : x \in \mathcal{F}_\sigma = \{A_\sigma x \geq b, \ x \in \{0,1\}^n\} \quad \forall \sigma \in \Sigma$$

which finds a *robust optimal solution*, namely a solution that is feasible in all the scenarios and have the best value under the worst scenario. The problem can be equivalently written as (see e.g., [6]):

$$
\begin{aligned}
V^{\mathrm{R}} = \min \ & \gamma & & \text{(Rob-BLP)} \\
& c'_\sigma x \leq \gamma & & \sigma \in \Sigma \\
& A_\sigma x \geq b & & \sigma \in \Sigma \\
& x \in \{0,1\}^n
\end{aligned}
$$

which is commonly called *robust counterpart* of the original problem BLP. We remark that in general the optimal value of the robust counterpart is worse than that of the original problem, i.e. $V^* \leq V^{\mathrm{R}}$. The deterioration in the optimal value commonly goes under the name of *price of robustness* [8], as it represents the "price" that we have to "pay" in order to guarantee protection against the data uncertainty specified by the scenarios in Σ.

We can adapt these results to the robust BAND case by introducing a set of scenarios $\sigma \in \Sigma$ such that each scenario σ is associated with a vector $d^\sigma = (d^\sigma_{11} \cdots d^\sigma_{bs} \cdots d^\sigma_{|B||S|})$ specifying the value of the bitrate between each biosensor-sink couple in σ. Such data uncertainty influences the coefficient matrix - specifically the relay-capacity constraints (12) - and the objective function and can be tackled as in the problem Rob-BLP, thus leading to the

following robust counterpart of the problem BAND-BLP:

$$\min \quad E \qquad\qquad\qquad\qquad \text{(Rob-BAND-BLP)} \quad (14)$$

$$\sum_{b \in B} \sum_{s \in S} \sum_{(i,j) \in A} E_{ij}\, d^{\sigma}_{bs}\, x^{bs}_{ij} \leq E \qquad\qquad \sigma \in \Sigma \qquad\qquad (15)$$

$$- \sum_{\substack{(b,j)\in \\ A_{B \to R} \cup A_{B \to S}}} x^{bs}_{bj} = -1 \qquad\qquad b \in B, s \in S \qquad (16)$$

$$\sum_{\substack{(r,j)\in \\ A_{R \leftrightarrow R} \cup A_{R \to S}}} x^{bs}_{rj} - \sum_{\substack{(j,s)\in \\ A_{B \to R} \cup A_{R \leftrightarrow R}}} x^{bs}_{jr} = 0 \quad b \in B, s \in S, r \in R \quad (17)$$

$$\sum_{\substack{(j,s)\in \\ A_{B \to S} \cup A_{R \to S}}} x^{bs}_{js} = 1 \qquad\qquad b \in B, s \in S \qquad (18)$$

$$\sum_{\substack{(r,j)\in \\ A_{R \leftrightarrow R} \cup A_{R \to S}}} d^{\sigma}_{bs}\, x^{bs}_{rj} \leq c_r\, y_r \qquad\qquad r \in R, \sigma \in \Sigma \qquad (19)$$

$$\sum_{r \in R} y_r \leq U \qquad\qquad\qquad\qquad\qquad (20)$$

$$x^{bs}_{ij} \in \{0,1\} \qquad\qquad\qquad b \in B, s \in S, (i,j) \in A$$

$$y_r \in \{0,1\} \qquad\qquad\qquad r \in R\,.$$

where we have introduced a single new decision variable E to represent an upper bound on the total energy consumed by routing decisions over all the scenarios Σ. Furthermore, we have introduced robust versions of the relay-capacity constraints (19), taking into account the bitrate of the specific scenario σ.

In the next paragraph, we proceed to describe the original optimization algorithm that we devised to solve problem Rob-BAND-BLP.

4. A fast optimization algorithm for the Rob-BAND-BLP

The optimization problem Rob-BAND-BLP is a binary linear program and in principle can be solved by any commercial *Mixed Integer Programming* (MIP) solver like IBM ILOG CPLEX [15]. However, the problem may actually result challenging even for a state-of-the-art solver like CPLEX, whose performance may result not satisfying and not sufficiently fast for practical application, as pointed out in the computational section. Our aim was thus to develop an algorithm able to deliver solutions of better quality and in less

time than CPLEX. Providing better solutions in less computational time was pointed out as a critical objective in discussions with medic professionals. The original algorithm that we propose is based on the combination of a heuristic variable fixing strategy based on randomized algorithms considerations and ant-colony-like procedures [31, 29] with an MIP heuristic used to repair and improve solutions. The ant-like part is partially inspired by ANTS (Approximate Nondeterministic Tree-Search) [29], a refined ant colony algorithm that attempts at exploiting information about bounds available for the problem. The MIP heuristic is based on executing an *exact large variable neighborhood search* [26], where the exploration is formulated as a mixed integer program solved exactly by using CPLEX (the formulation as a program and the solution by CPLEX is the reason for which the search is defined *exact* - in what follows we will also refer to it as an *exact MIP heuristic*).

We decided to investigate the combination of a variant of ANTS with an exact MIP heuristic on the basis of our direct and successful past experience in the context of other telecommunication network design problems, namely the design of cooperative wireless networks [17] and the multiperiod design of fixed telecommunications networks involving flow optimization problems [18, 19].

We stress that a distinctive feature of our original algorithm is that, though feasible solutions are built by heuristic approaches, we strongly found the construction on polyhedral information coming from suitable linear relaxations of the problem, namely formulations obtained by relaxing the integrality requirement of the variables (i.e., instead of considering $x \in \{0,1\}^n$ we allow $x \in [0,1]^n$). Using such information, we obtain lower bounds on the value of the optimal solutions that, when combined with the upper bound provided by a feasible solution, allow us to derive a so-called *optimality gap*, i.e. a measure of how far the solution is from the optimum. Our original algorithm can be viewed as a hybrid algorithm that combines a heuristic and an exact approach and has the desirable feature of providing quality guarantees about the solutions. For an introduction to hybrid heuristic, we refer the reader to the recent survey [10].

Before proceeding to describe our original optimization algorithm, we concisely review the main features of general ant colony algorithms and of the refined algorithm ANTS that we have taken as reference.

4.1. A concise introduction to ant colony optimization and ANTS

Ant Colony Optimization (ACO) is a metaheuristic inspired by the behaviour of ants searching for food that was originally proposed for com-

binatorial optimization problems by Dorigo, Maniezzo and Colorni [21] and later generalized to other classes of problems (see, e.g., [22, 29]). We refer the reader to [9] and [10] for an overview of theory and applications of ACO. The essential structure of an ACO Algorithm (ACOA) is commonly described as in Algorithm 1. The algorithm includes a main ant construction loop that is executed until an arrest condition is satisfied (e.g., reaching a computational time limit). During each execution of the loop, a number of ants, actually corresponding to a set of computational agents, attempts to build a set of of feasible solutions, one for each ant, by probabilistically fixing the value of integral variables using a function that resembles pheromone trails. At the end of the construction phase, the pheromone levels of the trails are updated, generally rewarding good fixing and penalizing bad fixing, and a new execution of the loop starts. When the arrest condition is satisfied, so-called daemon actions take place, trying to improve the quality of the produced solution, typically by a local search procedure that identifies local optima.

Algorithm 1 General ACO Algorithm (ACOA)

1: **while** an arrest condition is *not* satisfied **do**
2:     ant-based solution construction
3:     pheromone trail update
4: **end while**
5: daemon actions

In the rest of the subsection, we provide more details about the main features of the construction and of the pheromone update phases of the ACOA, referring anyway the reader to [21, 22, 29] for an exhaustive introduction to ACO algorithms. A detailed description of our algorithm, included the exact MIP heuristic adopted as daemon action, is provided in Paragraph 4.2.

4.1.1. Ant-based solution construction.

During each execution of the loop of the ACOA, $m \geq 0$ *ants* iteratively construct m feasible solutions. At a generic iteration of the construction phase, an ant is in a *state* corresponding to a *partial solution* for the optimization problem and can further complete the solution by executing a *move*, namely fixing the value of a variable that is not yet fixed. The choice of the fixing follows a probability function derived by combining an *a-priori* and an *a-posteriori* measure of the efficacy of the fixing. More specifically, the canonical formula for computing the probability of an ant k making a

move that fixes a variable j after having fixed a variable i is:

$$p_{ij}^k = \frac{\tau_{ij}^{\beta} + \eta_{ij}^{\delta}}{\sum_{f \in F} \tau_{if}^{\beta} + \eta_{if}^{\delta}} \quad, \tag{21}$$

where τ_{ij} is the so-called *pheromone trail value*, which represents the a priori measure of efficacy, and η_{ij} is the so-called *attractiveness*, which represents the a-posteriori measure of efficacy. In this formula, the measures are influenced by two parameters β, δ that should be chosen by the decision maker for the specific considered problem.

In our algorithm, we do not refer to (21), but we rely on the following improved probability formula that has been proposed in the algorithm ANTS - *Approximate Nondeterministic Tree-Search* [29]:

$$p_{ij}^k = \frac{\alpha \, \tau_{ij} + (1 - \alpha) \, \eta_{ij}}{\sum_{f \in F} \alpha \, \tau_{if} + (1 - \alpha) \, \eta_{if}} \quad, \tag{22}$$

We base part of our original algorithm for solving Rob-BAND-BLP on features of ANTS. We consider ANTS attractive since it proposes a number of smart refinements for the canonical ant algorithms that allow to better exploit polyhedral information about the problem. In particular, [29] sketches some ideas about how alternative formulations of the original problem could be exploited to define the pheromone trail and the attractiveness values. Specifically, the ideas that we use are to set the attractiveness values η_{ij} equal to fast computable lower bounds on the optimal value of the problem, whereas the initial pheromone trail values $\tau_{ij}(0)$ are set equal to lower bounds of better quality that however require more time to be computed.

The formula (22) of ANTS adopts more computationally efficient operations, replacing powers with products, and relies on a single parameter α, which is used to combine the a-priori and a-posteriori measures and replaces β and δ. The parameter $\alpha \in [0, 1]$ indicates the relative importance of the a-priori and the a-posteriori measures in the combination. As explained and discussed in [29], the values τ_{ij} and η_{ij} should be provided by suitable lower bounds of the considered optimization problem. In the case of our BAND: 1) τ_{ij} is derived from the values of the variables in the solution associated with the linear relaxation of the robust counterpart Rob-BAND-BLP; 2) η_{ij} is equal to the value of a (good) linear relaxation of BAND-BLP, where the value of a subset of the decision variables is fixed due to fixing decision that have been taken in previous steps of the algorithm.

4.1.2. Pheromone trail update

At the end of each ant-construction phase h, the ACOA provides for the update of the pheromone trail values, on the basis of how effective the ant moves have resulted. The canonical formula is:

$$\tau_{ij}(h) = \rho\,\tau_{ij}(h-1) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

where ρ is the persistence factor of the pheromone trail and $\Delta\tau_{ij}^{k}$ is the amount of pheromone that ant k deposits when using move (i,j). The ANTS algorithm proposes to refine the previous formula with the following one, which we also adopt:

$$\tau_{ij}(h) = \tau_{ij}(h-1) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad \text{with } \Delta\tau_{ij}^{k} = \tau_{ij}(0)\cdot\left(1 - \frac{z_{curr} - LB}{\bar{z} - LB}\right). \quad (23)$$

Here, z_{curr} is the value of the last solution produced, $\bar{z}$ is a moving average of the values of the last w solutions produced and LB is a lower bound on the value of the optimal solution of the problem. This formula does not contain the persistence factor ρ, whose setting is in general tricky, but depends on the width w of the moving average, whose setting is less delicate. Furthermore, the definition of $\Delta\tau_{ij}^{k}$ is connected to the quality of the last w solutions produced, expressed by the moving average $\bar{z}$ of their value, compared to the quality of the last solution produced, indicated by its value z_{curr} (see [29] for a detailed discussion of the formula).

We now proceed to describe in detail our original algorithm for solving Rob-BAND-BLP.

4.2. An optimization algorithm for the fast and robust design of BANs

A major challenge for a heuristic solving the problem BAND-BLP or its robust counterpart Rob-BAND-BLP is represented by the definition of the routing paths, identified by fixing the value of the unsplittable flow variables x_{ij}^{bs} for each relevant biosensor-sink couple (b,s). Once that the routing paths are defined, we can derive a possibly feasible activation of relay nodes that will support the routing.

To explain how we build the routing paths, we define the set of biosensor-sink couples $C \subseteq B \times S$ containing couples (b,s) for which there exists at least one data scenario with non-zero bitrate, i.e. $C = \{(b,s) \in B \times S : \exists \sigma \in \Sigma \text{ with } d_{bs}^{\sigma} > 0\}$, and we introduce the concept of routing state.

Definition 2 (Routing state - RS). Consider a subset of biosensor-sink couples $\bar{C} \subseteq C$. We define *routing state* a fixing of the unsplittable flow variables $x_{ij}^{bs} \; \forall (i,j) \in E$ for each $(b,s) \in \bar{C}$ such that the fixing is feasible for the flow conservation constraints (16-18).

A *routing state* can be interpreted as an assignment of a single routing path to each couple $(b,s) \in \bar{C}$. We say that a routing state is *partial* when $\bar{C} \subset C$ (i.e., only a subset of data flows is routed) and complete when $\bar{C} = C$ (i.e., all data flows are routed).

We remark that the definition of routing state made above is independent from relay capacity considerations. A complete routing state may thus result infeasible for the overall routing problem that includes the node capacity constraints: this may happen if we have built a routing state where many paths are using the same relay and the sum of the data that must be processed by the relay exceeds its capacity. Because of this possible infeasibility, the construction of a complete routing state must be followed by a *check-and-repair phase*, in which the feasibility of the routing state is checked and, if not verified, is repaired to become feasible. We attempt to repair an infeasible solution by using the same exact MIP heuristic based on variable neighborhood search that we adopt to improve a feasible solution (see Subsection 4.4 for details about the MIP heuristic for the reparation and the improvement).

Checking the feasibility of a complete routing state for Rob-BAND-BLP is straightforward: we activate all relays along routing paths used in the routing state (this corresponds to fix to 1 the values of the corresponding relay deployment variables y_r and to 0 all the other variables y_r) and we check if there exists any relay-capacity constraint (19) that is violated for some data scenario $\sigma \in \Sigma$. Moreover, we check whether the number of activated relays exceeds the limit imposed by constraint (20). If no constraint is violated, then we have characterized a feasible solution for Rob-BAND-BLP: we have indeed determined a fixing of the unsplittable flow variables x and derived a fixing of the deployment variables y that respect the flow conservation constraints in each scenario and the activation constraints. On the contrary, if any capacity or activation constraint is violated, then we have produced an infeasible routing and we must repair it. We note that the fact that produced routing states may be infeasible requires to develop an algorithm that is deeply different from those that we have proposed in [18, 19], where no infeasibility issue was faced.

The complete algorithm is presented in Algorithm 2 and we call it ROBUBAND. We remark that we denote the energy value of a solution

25

$(\bar{x}, \bar{y})$ by $E(\bar{x}, \bar{y})$. The algorithm is based on two nested loops: the outer loop is executed until a time limit is reached and provides for the execution of an inner loop where m feasible solutions are built according to an ANTS-like procedure. More in detail, the algorithm starts by solving the linear relaxation of Rob-BAND-BLP that represents, according to the principles of ANTS, the tighter linear relaxation used to initialize the a-priori measure of attractiveness $\tau_{ij}(0)$. A solution (x^*, y^*) is also introduced to denote and store the best solution found during the entire execution of RobuBAND. In each execution of the inner loop, the first step is to build a complete routing state by procedure BuildRouting described in Subsection 4.3. The complete routing state specifies the values of the variables $\bar{x}$ and is used to derive a relay installation $\bar{y}$, as explained before. This leads to the definition of a solution $(\bar{x}, \bar{y})$ that, however, is not necessarily feasible and may need to be repaired by running the heuristic MIP-VNS (described in detail in Subsection 4.4). If the solution $(\bar{x}, \bar{y})$ is feasible and better than the best solution found (x^B, y^B) until the current execution of the inner loop, we update (x^B, y^B). Then we iterate the inner loop.

Once that the inner loop is concluded, we update the values of τ_{ij} according to the ANTS formula (23) and we check whether the best solution of the inner loop (x^B, y^B) is better than the global best solution (x^*, y^*), operating an update if necessary. Finally, once that the time limit is reached, we execute the heuristic MIP-VNS for improving the best solution found and at the end of the execution we return (x^*, y^*).

In the next subsections, we describe in detail how we build a complete routing state and the features of the heuristic MIP-VNS.

4.3. Construction of a complete routing state

We construct a complete routing state by considering the assignment of paths to biosensor-sink couples in a pre-established order. Specifically, we sort couples $(b, s) \in C$ in descending order w.r.t. their highest bitrate value d_{bs}^{σ} over all the scenarios $\sigma \in \Sigma$. For each couple (b, s), we assign the entire data flow to a path p connecting b to s defined according to the procedure BuildRouting showed in Algorithm 3.

In Algorithm 3, the construction of a complete routing state is based on an outer loop that, following the ordering of couples, at each execution assign a path p^* to a couple $c = (b, s)$ connecting b to s. The path is chosen from a set of candidates P_c, which is built in the following way: 1) we solve the linear relaxation of Rob-BAND-BLP, where we have fixed the value of variables of couples $c \in \bar{C}$ for which a path has been assigned in previous executions of the loop; 2) using the solution (x^{RB}, y^{RB}) of the

Algorithm 2 ROBUBAND

1: compute the linear relaxation of Rob-BAND-BLP and initialize the values $\tau_{ij}(0)$ through it
2: let (x^*, y^*) denote the best solution found by ROBUBAND
3: **while** a global time limit is not reached **do**
4:     let (x^B, y^B) denote the best solution found in the inner loop
5:     **for** $k := 1$ to m **do**
6:         build a complete routing state $\bar{x}$ by procedure BUILDROUTING
7:         derive a relay installation $\bar{y}$ using $\bar{x}$
8:         **if** $(\bar{x}, \bar{y})$ is not feasible for Rob-BAND-BLP **then**
9:             run MIP-VNS$(\bar{x}, \bar{y})$ for repairing $(\bar{x}, \bar{y})$
10:         **end if**
11:         **if** $(\bar{x}, \bar{y})$ is feasible and $E(\bar{x}, \bar{y}) < E(x^B, y^B)$ **then**
12:             update the best solution found $(x^B, y^B) := (\bar{x}, \bar{y})$
13:         **end if**
14:     **end for**
15:     update $\tau_{ij}(t)$ according to (23)
16:     **if** $E(x^B, y^B) < E(x^*, y^*)$ **then**
17:         update the best solution found $(x^*, y^*) := (x^B, y^B)$
18:     **end if**
19: **end while**
20: run MIP-VNS(x^*, y^*) for improving (x^*, y^*)
21: return (x^*, y^*)

---

relaxation, we derive a graph $H^c(V, A^{mod})$ from $G(V, A)$ as follows: the set of vertices is the same, while in A^{mod} we maintain only arcs $(i,j) \in A$ such that $x_{ij}^{RB\,c=(b,s)} \neq 0$, i.e. arcs associated with flow variables having non-zero (fractional) values. Additionally, for each arc $(i,j) \in A^{mod}$ we introduce a weight $w_{ij} = x_{ij}^{RB\,c=(b,s)}$. We derive L candidate paths for couple $c = (b, s)$ on graph $H^c(V, A^{mod})$ by iteratively modifying $H^c(V, A^{mod})$: in an inner loop, at each iteration we find the shortest path p w.r.t. the weights w_{ij} in $H^c(V, A^{mod})$, then we add p to the set P_c and we delete the arc of p with lowest weight from $H^c(V, A^{mod})$. This is a simple yet fast and effective procedure to find candidate paths. We delete the arc with lowest weight since, if we interpret a fractional value of a binary variable as the probability of fixing to 1 the variable in a good solution, then lower fractional values should lead to fixing to 1 of lower quality (see [31] for a discussion about the interpretation of fractional binary solutions as probability in randomized rounding algorithms). After having established the set of candidate paths P_c for c, we compute the probability of choosing each path $p \in P_c$ to route the entire flow of couple c. This is done according to formula (22), where η_{ij} is set equal to the optimal value of the linear relaxation of BAND-BLP, whereas τ_{ij} is obtained as the sum of the current values of the a-priori measure τ_{ij} for the edges in path p. We remark that computing η_{ij} as the value of the linear relaxation of the *non-robust* problem BAND-BLP is in line with the principles of ANTS, suggesting to derive η_{ij} from a lower bound that can be fast computed at the price of a lower quality of the bound. Moreover, we note that the way we compute τ_{ij} privileges the selection of paths composed by arcs with higher a-priori measures, a criterion that should lead to better variable fixing decisions. Once that a path $p^* \in P_c$ has been probabilistically chosen according to formula (22) computed as we detailed above, we derive a fixing of variables x^{bs}, where $x_{ij}^{bs} = 1$ if (i,j) belongs to p^* and $x_{ij}^{bs} = 0$ otherwise. Finally, we add the couple c to the set of processed couples $\bar{C}$ for which the routing has been established.

After executing $|C|$ times the external loop, following the ordering of the couples, a complete routing state is then available. Assuming an ANTS point of view, we note that each iteration of the outer loop of BUILDROUTING can be interpreted as an ant moving from a partial routing state to a more complete partial routing state.

4.4. MIP-VNS - an exact large variable neighborhood search
In order to either repair the infeasibility of a solution $(\bar{x}, \bar{y})$ produced in the inner loop of ROBUBAND or to improve a feasible solution available at the

Algorithm 3 BUILDROUTING

 1: sort $(b, s) \in C$ in descending order of max $\{d_{bs}^\sigma \ \ \forall \sigma \in \Sigma\}$
 2: $\bar{C} = \emptyset$
 3: **for** (sorted $c \in C$) **do**
 4:     compute the linear relaxation of Rob-BAND-BLP for $\bar{C}$ and let (x^{RB}, y^{RB}) be the corresponding solution
 5:     let P_c be the set of candidate paths for couple c (initially empty)
 6:     build the graph $H^c(V, A^{mod})$ with arc weights given by x^{RB}
 7:     **for** $\ell := 1$ to L **do**
 8:         find the shortest path p for couple c in $H^c(V, E^{mod})$
 9:         $P_c = P_c \cup \{p\}$
10:         exclude the arc of p with minimum weight from $H^c(V, E^{mod})$
11:     **end for**
12:     compute the probability of using $p \in P_c$ as path for routing data of c according to formula (22)
13:     choose probabilistically a path $p^* \in P_c$ for routing data of c
14:     fix variables x^{bs} activating arcs in p^* and deactivating arcs not in p^*
15:     $\bar{C} = \bar{C} \cup \{c\}$
16: **end for**
---

end of the outer loop, we rely on an exact MIP heuristic based on a *large variable neighborhood search*: besides combining the principles of *large* [2] and *variable* neighborhood search [30], our heuristic has the feature of being *exact*, that is we formulate the search as a binary linear program exactly solved through a state-of-the-art MIP solver. Specifically, given a (feasible or infeasible) solution $(\bar{x}, \bar{y})$ to the problem, we consider the neighborhood $\mathcal{N}(\bar{x}, \bar{y})$ including all the feasible solutions of Rob-BAND-BLP that can be obtained from $(\bar{x}, \bar{y})$ modifying at most $\Gamma > 0$ components of $\bar{y}$ and leaving $\bar{x}$ free to vary. This can be expressed by a version of Rob-BAND-BLP where we impose the additional *hamming distance constraint*:

$$HD(\bar{y}, y) \ = \ \sum_{r \in R: \ \bar{y}_r = 0} y_r + \sum_{r \in R: \ \bar{y}_r = 1} (1 - y_r) \ \leq \ \Gamma$$

which counts the number of variables changing values in y w.r.t. y'. Furthermore, we impose an optimality constraint:

$$E(x, y) \leq E(x^*, y^*) - \epsilon$$

expressing that we must find a feasible solution in $\mathcal{N}(\bar{x}, \bar{y})$ that is better than the current best global feasible solution (in particular, we request a minimum improvement of $\epsilon > 0$). The resulting modified problem, denoted

by MOD-Rob-BAND-BLP, can then be solved by a solver like CPLEX. Note that this constitutes an *exact local search* (the search is formulated as an optimization problem solved exactly by CPLEX) in a *large variable neighborhood* (including all the feasible solutions that are within distance Γ w.r.t. variables $\bar{y}$). The presence of the optimality constraint is particularly useful for CPLEX: we observed that CPLEX can fast recognize a problem as infeasible, thus certifying that the neighborhood contains no better solution and thus allowing to not waste time in useless searches. It is also wise to impose a time limit to the solution of MOD-Rob-BAND-BLP: CPLEX may take a remarkable amount of time to close the optimality gap and, on the other hand, is generally able to fast find good solutions for problems of reduced size. We embed the solution of the problem MOD-Rob-BAND-BLP in the procedure that we denote by *MIP-VNS*, which is executed for reparation and improvement in Algorithm 2. The steps of MIP-VNS are detailed in Algorithm 4.

Algorithm 4 MIP-VNS

1: let $(\bar{x}, \bar{y})$ be a given incumbent solution
2: let (x^Γ, y^Γ) denote the best feasible solution found by MIP-VNS for $\Gamma > 0$
3: **while** a global time limit is not reached **do**
4:     solve MOD-Rob-BAND-BLP for $(\bar{x}, \bar{y})$ and $\Gamma > 0$ with time limit $\tau > 0$
5:     **if** $E(x^\Gamma, y^\Gamma) < E(x^*, y^*)$ **then**
6:         update the best solution found $(x^*, y^*) := (x^\Gamma, y^\Gamma)$
7:         update the value of the optimality constraint
8:         update the incumbent solution $(\bar{x}, \bar{y}) := (x^\Gamma, y^\Gamma)$
9:     **end if**
10:     $\Gamma := \Gamma + \Delta$
11: **end while**

Given an incumbent solution, we solve MOD-Rob-BAND-BLP a number of times until a global time limit is reached. Every time that a problem MOD-Rob-BAND-BLP is solved, we impose a local time limit. When the time limit of a single problem is reached or the problem is solved, we enlarge the neighborhood by increasing the hamming distance limit by a step $\Delta > 0$ (i.e., we impose $\Gamma = \Gamma + \Delta$ with $\Delta > 0$) and start a new exact search until the global time limit is reached.

After having detailed the entire algorithm, we finally proceed to present computational experiments in the next section.

5. Experimental results

We assessed the performance of our new optimization algorithm for the robust design of BANs by considering a set of 30 instances that we defined taking as reference past literature about the topic (in particular [11, 24, 23], which have analyzed the design of BANs with relays) and on the basis of discussions with medical professionals from a major Italian medical institution. All the 30 instances consider a BAN including a total of 16 biosensors (i.e., $|B| = 16$) and 2 sinks (i.e., $|S| = 2$) placed in pre-established positions over the reference body. Moreover, we consider 400 potential sites for the location of relays (i.e., $|R| = 400$), chosen randomly over the human body excluding head, hands and feet. As in [11, 24, 23] we assume that: a) the devices within a distance of 0.3 meters are able to communicate; b) the path loss coefficient λ_{ij} is equal to 3.38 for line-of-sight propagation and equal to 5.90 for non-line-of-sight wireless links $(i, j) \in A$; c) the BAN adopts Nordic *nRF2401* transceivers [32], which are often used in WSNs, and the corresponding energy consumption parameters are $E_{\text{TX}_{CIRC}} = 16.7$ nJ/bit, $E_{\text{TX}_{AMP}}(\lambda_{ij} = 3.38) = 1.97$ nJ/bit, $E_{\text{TX}_{AMP}}(\lambda_{ij} = 5.90) = 7990$ nJ/bit, $E_{\text{RX}_{CIRC}} = 36.1$ nJ/bit.

As in Section 2, we assume that the biosensors and sinks possess all the necessary capacity to process the data that they are supposed to transmit and receive, respectively. In the case of relays, we assume instead a capacity of $c_r = 250$ kbit/s. The set of traffic scenarios was generated randomly assuming that 50% of the biosensors generate and transmit data at a constant rate in {100,150,200} bit/s, whereas the rest of biosensors have a variable rate lying in the range [100,200] bit/s. For each instance, we generated 25 possible data rate scenarios that form the set Σ.

We performed all the experiments on a 2.70 GHz machine with 8 GB. The code was written in the C/C++ programming language and the optimization problems were solved by IBM ILOG CPLEX 12.5 interfaced with the code through Concert Technology and running with a time limit of 2400 second. All the instances led to the definition of Rob-BAND-BLP problems that proved challenging even for a state-of-the-art MIP solver like CPLEX. As it is clearly shown in Table 1, when CPLEX reaches the time limit, it gives feasible solutions that are still sensibly far from the optimum, as indicated by the measure *GapBLP%*. GapBLP% is the optimality gap, which indicates how far the best feasible solution found is far from the best lower bound produced for the optimal value (if V^* is the value of the best solution and LB the value of the best bound then we compute the gap as $Gap\% = |V^* - LB|/V^* \cdot 100$). In contrast, our original algorithm RobuBAND is able

to deliver solutions that are associated with much better optimality gap.

Concerning ROBUBAND, we first conducted preliminary tests that brought us to adopt the following setting of the parameters: the number L of candidate paths considered in BUILDROUTING is set equal to 5, whereas we set $\alpha = 0.5$ to balance the combination of the a-priori and a-posteriori measures. We adopted $m = 20$ computational agents/ants and the width w of the moving average was set equal to 4. In order to solve the linear relaxation of Rob-BAND-BLP and of BAND-BLP we used CPLEX. In the case of BAND-BLP, which must run many times, we improve the speed of solution by selecting a truncated primal simplex algorithm for computing good relaxations in small amount of time. Concerning the parameters influencing the MIP heuristic MIP-VNS, we used a tolerance $\epsilon = 10^{-1}$, a maximum hamming distance $\Gamma = 0.1|R|$ and a time limit of 10 minutes for MIP-VNS when used for improvements and of 1 minute for MIP-VNS used for solution reparation. The main cycle of ROBUBAND ran with a time limit of 30 minutes, which added up to the time reserved for MIP-VNS thus matched the time limit of CPLEX.

The complete set of results is presented in Table 1, where we show the performance of ROBUBAND, denoted by the acronym BR, and of CPLEX when applied directly to solve Rob-BAND-BLP, denoted by the acronym BLP. For both ROBUBAND and CPLEX, we report the measures related to the best solution found within the time limit for each instance identified by its ID. Specifically, E_{AVG} is the average value of the energy consumed (in $\mu J/bit$) while Gap% is the optimality gap computed as explained above. Finally, $\Delta Gap\%$ is the percentage increase of the optimality gap of CPLEX w.r.t. that of ROBUBAND. In the case of ROBUBAND, the optimality gap is derived comparing the value of the linear relaxation of Rob-BAND-BLP computed by CPLEX with the value of the best feasible solution found by ROBUBAND within the time limit.

It is clear that in most cases ROBUBAND performs much better than CPLEX reaching a final optimality gap and average energy consumption values that are sensibly smaller than that of CPLEX. In the vast majority of cases, CPLEX return solutions associated with average energy values and optimality gaps that are at least 10% higher and thus *worse* than that of ROBUBAND (remember that larger gaps are associated with higher distances of the best solution from the optimum, thus implying a worse certificate of quality of the solution). In particular, the average percentage increase in the optimality gap is about 25%. The higher performance of RobuBand is particularly evident in the case of instances I6, I26 and I29, where the $\Delta Gap\%$ indicates a performance that is dramatically better than

that of CPLEX and is accompanied by a similar behaviour of the average energy consumption. At the same time, we notice that there are also two cases (I17 and I24) in which RobuBAND performs worse than CPLEX, but the difference is anyway contained. These experimental results thus support the conclusion that RobuBAND represents a competitive alternative to CPLEX to get better solutions in less time.

6. Conclusion and future work

In this paper, we have presented the first robust optimization model for addressing traffic uncertainty in the design of body area networks with relays and adopting single-path routing. Our work has been aimed at extending the use of optimization models and algorithms in the design of BANs, which has received very little attention. Since the resulting robust optimization problem may result challenging even for a state-of-the-art optimization solver, we have proposed an original optimization algorithm exploiting linear relaxations to guide a heuristic fixing of variables, combined with an exact large variable neighborhood search. Experiments on realistic instances show that our optimization algorithm performs much better than a solver like CPLEX in the vast majority of cases, fast finding solutions of better quality and associated with lower optimality gaps. We plan to aim future research at further improving the performance of the algorithm and refining the representation of traffic uncertainty by more advanced models, such as Multiband Robust Optimization [12].

References

References

[1] H. Aissi, C. Bazgan, D. Vanderpooten : Min-max and min-max regret versions of combinatorial optimization problems: A survey. Europ. J. Oper. Res. 197 (2) 427-438 (2009) DOI: 10.1016/j.ejor.2008.09.012

Table 1: Experimental results

ID	E_{AVG}(RB)	GapRB%	E_{AVG}(BLP)	GapBLP%	ΔGap%
I1	4.228	41.63	5.460	55.22	32.62
I2	5.427	57.10	6.294	67.17	17.62
I3	3.425	35.83	3.824	40.26	12.35
I4	4.147	42.16	4.390	45.20	7.21
I5	5.077	54.54	6.469	68.60	25.79
I6	4.200	38.33	6.368	60.45	57.71
I7	3.225	34.52	4.338	45.65	32.24
I8	4.429	48.78	5.771	64.09	31.38
I9	4.658	47.77	5.842	60.66	26.98
I10	2.841	28.10	3.476	34.08	21.27
I11	4.940	48.50	6.402	61.42	26.61
I12	4.550	46.65	5.710	60.97	30.69
I13	4.060	37.66	6.202	59.96	59.20
I14	5.406	50.77	6.825	63.92	25.91
I15	3.173	28.95	3.830	34.61	19.55
I16	3.177	31.89	3.773	38.33	20.21
I17	4.124	41.10	3.704	36.45	-11.31
I18	3.917	41.61	5.028	53.04	27.74
I19	3.025	31.97	3.517	36.81	15.14
I20	2.940	30.87	3.418	36.08	16.89
I21	3.189	29.03	3.891	34.89	20.18
I22	3.903	42.63	4.953	56.89	33.44
I23	4.332	43.58	5.269	52.83	21.22
I24	4.628	50.34	4.290	47.13	-6.39
I25	3.781	35.06	4.505	41.50	18.34
I26	3.642	38.43	6.260	67.46	75.50
I27	3.032	29.52	3.493	34.18	15.78
I28	4.822	52.11	5.862	64.75	24.26
I29	4.261	41.25	7.039	70.31	70.43
I30	4.736	48.35	5.878	59.95	23.98

[2] R.K. Ahuja, O. Ergun, J. Orlin, A. Punnen: A survey of very large-scale neighborhood search techniques. Discrete Appl. Math. 123, 75-102 (2002)

[3] R.K. Ahuja, T. Magnanti, J. Orlin: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River (1993)

[4] L. Baringo and A. J. Conejo: Offering Strategy Via Robust Optimization. IEEE Trans. Power Systems 26 (3), 418-1425 (2011)

[5] T. Bauschert, C. Büsing, F. D'Andreagiovanni, A.M.C.A. Koster, M. Kutschka, U. Steglich: Network Planning under Demand Uncertainty with Robust Optimization. IEEE Communications Magazine 52 (2), 178-185 (2014) DOI: 10.1109/MCOM.2014.6736760

[6] A. Ben-Tal, L. El Ghaoui, A. Nemirovski: Robust Optimization. Springer, Heidelberg (2009)

[7] D. Bertsimas, D. Brown, C. Caramanis: Theory and Applications of Robust Optimization. SIAM Review 53 (3) 464-501 (2011)

[8] D. Bertsimas, M. Sim: The Price of Robustness. Oper. Res. 52 (1), 35-53 (2004)

[9] C. Blum: Ant colony optimization: Introduction and recent trends. Physics Life Rew. **2** (4), 353-373 (2005)

[10] C. Blum, J. Puchinger, G.R. Raidl, A. Roli: Hybrid metaheuristics in combinatorial optimization: A survey. Appl. Soft Comp. 11 (6), 4135-4151 (2011)

[11] B. Braem, B. Latre, I. Moerman, C. Blondia, E. Reusens, W. Joseph, L. Martens, P.Demeester: The Need for Cooperation and Relaying in Short-Range High Path Loss Sensor Networks. Proc. of SensorComm 2007, 566-571 (2007) DOI: 10.1109/SENSORCOMM.2007.4394980

[12] C. Büsing, F. D'Andreagiovanni: New Results about Multi-band Uncertainty in Robust Optimization. In: Klasing, R. (ed.) Experimental Algorithms, LNCS, vol. 7276, pp. 63-74. Springer, Heidelberg (2012) DOI: 10.1007/978-3-642-30850-5_7

[13] R.D. Carr, H.J. Greenberg, W.E. Hart, G. Konjevod, E. Lauer, H. Lin, T. Morrison, C.A. Phillips: Robust Optimization of Contaminant Sensor Placement for Community Water Systems. Math. Program. 107 (1-2) 337-356 (2006) DOI: 10.1007/s10107-005-0689-x

[14] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, V.C.M. Leung: Body Area Networks: A Survey. Mobile Netw. Appl. 16 (2) 171-193 (2010) DOI: 10.1007/s11036-010-0260-8

[15] IBM ILOG CPLEX, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer.

[16] F. D'Andreagiovanni, C. Mannino, A. Sassano: GUB Covers and Power-Indexed Formulations for Wireless Network Design. Management Science 59 (1), 142-156 (2013) DOI: 10.1287/mnsc.1120.1571

[17] F. D'Andreagiovanni: A Hybrid Exact-ACO Algorithm for the Joint Scheduling, Power and Cluster Assignment in Cooperative Wireless Networks. In: G. Di Caro, G. Theraulaz (eds.) Bio-Inspired Models of Network, Information, and Computing Systems, LNICST vol. 134, pp. 3-17. Springer, Heidelberg (2014) DOI: 10.1007/978-3-319-06944-9_1

[18] F. D'Andreagiovanni, J. Krolikowski, J. Pulaj: A hybrid primal heuristic for Robust Multiperiod Network Design. In: Esparcia-Alcazar, A. et al. (eds.) Applications of Evolutionary Computation, LNCS vol. 8602, pp. 15-26. Springer, Heidelberg (2014) DOI: 10.1007/978-3-662-45523-4_2

[19] F. D'Andreagiovanni, J. Krolikowski, J. Pulaj: A fast hybrid primal heuristic for Multiband Robust Capacitated Network Design with Multiple Time Periods. Applied Soft Computing 26, 497-507 (2015) DOI: 10.1016/j.asoc.2014.10.016

[20] G.B. Dantzig: Linear programming under uncertainty. Management Science 1 (3) 197206 (1955)

[21] M. Dorigo, V. Maniezzo, A. Colorni: Ant System: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B 26 (1), 29-41 (1996)

[22] M. Dorigo, G. Di Caro, L.M. Gambardella: Ant algorithms for discrete optimization. Artificial Life 5 (2) 137-172 (1999)

[23] J. Elias: Optimal design of energy-efficient and cost-effective wireless body area networks. Ad Hoc Networks 13, 560-574 (2014) DOI: 10.1016/j.adhoc.2013.10.010

[24] Ehyaie, A., Hashemi, M., Khadivi, P.: Using relay network to increase life time in wireless body area sensor networks . Proc. of IEEE WoW-MoM 2009, 1-6 (2009) DOI: 10.1109/WOWMOM.2009.5282405

[25] E.F. Flushing, G. Di Caro: Relay Node Placement for Performance Enhancement with Uncertain Demand: A Robust Optimization Approach. Proc. of WiOpt 2013, 556-563 (2013)

[26] P. Hansen, N. Mladenovic, J.A.M. Perez: Variable neighbourhood search: methods and applications: Ann. Oper. Res. 175, 367407 (2010) DOI: 10.1007/s10479-009-0657-6

[27] S.L. Keoh, K. Twidle, N. Pryce, A. E. Schaeffer-Filho, E. Lupu, N. Dulay, M. Sloman, S. Heeps, S. Strowes, J. Sventek, E. Katsiri: Policy-based Management for Body-Sensor Networks. In: S. Leonhardt et al. (eds.) 4th Intern. Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), IFMBE Proc. vol. 13, pp. 92-98. Springer, Heidelberg (2007)

[28] J. Ko, C. Lu, M.B. Srivastava, J.A. Stankovic, A. Terzis, M. Welsh: Wireless Sensor Networks for Healthcare. Proc. IEEE 98 (11) 1947-1960 (2007) DOI: 10.1109/JPROC.2010.2065210

[29] V. Maniezzo: Exact and Approximate Nondeterministic Tree-Search Procedures for the Quadratic Assignment Problem. INFORMS J. Comp. 11 (4) 358-369 (1999)

[30] N. Mladenovic, P. Hansen: Variable neighborhood search. Comput. Oper. Res. 24 (11) 1097-1100 (1997) DOI: 10.1016/S0305-0548(97)00031-2

[31] R. Motwani, P. Raghavan: Randomized algorithms. Cambridge University Press, New York (1995)

[32] Nordic Semiconductor, http://www.nordicsemi.com/eng/Products

[33] Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice Hall, Upper Saddle River (2001)

[34] H. Ren, M.Q. Meng: Bioeffects Control in Wireless Biomedical Sensor Networks. Proc. IEEE SECON 2006, 896-904 (2006) DOI: 10.1109/SAHCN.2006.288579

[35] A. Shapiro, D. Dentcheva, A. Ruszczynski: Lectures on stochastic programming: Modeling and theory. MPS/SIAM Series on Optimization 9. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2009)

[36] J. Yick, B. Mukherjee, D. Ghosal: Wireless sensor network survey. Computer Networks 52 (12) 2292-2330 (2008)