

Wireless multicasting under probabilistic node failures: a heuristic approach

János Barta · Roberto Montemanni

Received: 21 December 2010 / Accepted: 15 September 2011 / Published online: 6 October 2011
© Springer Science+Business Media, LLC 2011

Abstract The minimum power multicast (MPM) problem is a well-known optimization problem in wireless networks. The aim of the MPM problem is to assign transmission powers to the nodes of a wireless sensor network in such a way that multi-hop communication between a source node and a set of destination nodes is guaranteed, while the total transmission power expenditure over the network is minimized. Several extensions to the basic problem have been proposed, in order to obtain more realistic mathematical models. In this paper we deal with the probabilistic minimum power multicast (PMPM) problem, where node failure probabilities are considered and a global reliability level of the transmission is required. Since the so far available exact approach can handle only small-sized instances of the PMPM problem, in this paper we focus on the study of a heuristic approach. A heuristic algorithm for the PMPM problem is presented, together with a fast method for the reliability calculation based on previously unexplored combinatorial properties of the model. Computational experiments are finally discussed.

Keywords Minimum power multicasting · Probabilistic mathematical models · Multihop networks · Reliability calculation

1 Introduction

Since the very beginning of research in the area of wireless sensor networks, one of the major issues has been saving power. This optimization is typically faced during the design, and prior the deployment of the nodes of a network. Such a high attention for this factor is easy to identify: the nodes of the network (devices) are typically

J. Barta (✉) · R. Montemanni
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), USI-SUPSI, 6928 Manno-Lugano,
Switzerland
e-mail: janos@idsia.ch

equipped with low capacity, tiny batteries, and they have to stay alive in the longest possible time horizon. This, in an environment which is usually characterized by reduced accessibility. A tight management of the power budget is imposed by all these factors. Another peculiarity of sensor networks is that the largest share of power consumption is normally due to communication rather than to computation, sensing and state-changing activities (Negri et al. 2008; Wieselthier et al. 2000). We will base our study on this assumption.

Wireless sensor networks are typically used in commanding actuators, monitoring events or measuring values at locations where people cannot reach easily, or where a long term sensing task is required. Examples of applications are habitat monitoring (Mainwaring et al. 2002), civil structural monitoring (Kim et al. 2007) and environmental monitoring (Doolin and Sitar 2005). Nodes can usually be characterized as low cost devices, and are expected to be deployed in a potentially inaccessible area. Recharging the sensors after the deployment might therefore not be an option, both for logistic and economical reasons. In this context, energy-efficiency becomes perhaps the most important design criteria for sensor networks, since it directly impacts on the time the network itself is kept in operation. Many of the sensor networking applications are intrinsically about dissemination of information from a well-identified *source* node. It is therefore critical to identify energy efficient network topologies, optimized according to the type of communications that has to be supported. In this paper we will concentrate on multicasting topologies, where a piece of information has to be sent from a source node to a set of target nodes of the network. We will assume to work on a static network, i.e. nodes do not move after the deployment.

In this paper we focus on the problem of minimizing the power required to connect a source device to a set of hosts. This optimization problem, which is at present an intensive topic of study, is known as the Minimum Power Multicast (MPM) problem. Indeed, significant effort is being done for modelling and solving the MPM and its variants like the Minimum Power Broadcast (MPB) problem (see Leggieri et al. 2008; Leino 2002; Montemanni and Gambardella 2005 and Das et al. 2003). A detailed review on exact and heuristic methods to solve both problems can be found in the recent surveys (Guo and Yang 2007 and Min and Chinchuluun 2006).

All the above works assume a deterministic behavior of the transmitting devices. In reality it has to be expected that the terminals can be affected by temporary damage or permanent failure. Therefore it is reasonable to extend the deterministic case to a probabilistic formulation that takes into account the uncertain nature of node availability. It will be soon clear that this extension introduces an extra layer of complexity to the problem. However, the advances in heuristic and exact algorithms for the MPM, lead to the opportunity of studying more realistic, although more complex, models of the problem. Simplified models taking into account failure probabilities associated with nodes were discussed in Bhandari and Vaidya (2007) and Montemanni et al. (2008). A more realistic approach was studied in Barta et al. (2011). This work presented a probabilistic version of the MPM (referred to as PMPM in the remainder of the paper), that takes into account node failure probabilities. Each device has a known given probability of failure, due to environmental factors, and it is imposed that connectivity should be guaranteed with a given level of reliability. Notice that in wireless networks failure probabilities are also correlated to the transmission power

assigned to the devices. However, here we focus on situations in which the environmental sources of node failure (e.g. the deployment in a hostile area) dominate the channel/fading/mobility-related sources of failure. In this paper we do not consider issues regarding communication protocols, which are delegated to other layers of wireless architecture.

The resulting mathematical formulation lies into the class of probabilistic integer programming models. An exact solution approach based on the formulation is finally discussed. The method is able to solve problems with up to 30 nodes within one hour (on a modern computer). Some new valid inequalities for the integer programming formulation are discussed in Barta et al. (2010).

In order to handle (realistic) instances with more than 30 nodes, heuristic methods have to be developed. In the present paper we devise such an approach, based on the exploitation on some previously unexplored theoretical properties of the model.

It is trivial to see that PMPM is an NP-hard problem: the deterministic MPM problem, which is known to be NP-complete (see Cagalj et al. 2002 and Liang 2002), is a special case of problem PMPM. More precisely, a PMPM instance reduces to an MPM problem if the probability of failure related to each node is set to zero (see also Rosenthal 1977; Bodlaender and Wolle 2004 and Garey and Johnson 1979).

The remainder of the paper is organized as follows. Section 3 is devoted to the definition of reliability of a wireless sensor network, the discussion of the properties of reliability and the introduction of an efficient algorithm to calculate it. In Sect. 4 a heuristic algorithm for the PMPM, based on the reliability calculation previously discussed, is introduced. Computational results for the new approach are presented in Sect. 5. Section 6 is finally devoted to our conclusions.

2 Description of the probabilistic minimum power multicast problem

A network of wireless devices can be modelled mathematically as a directed graph $G = (V, A)$, where the elements of the set V are the devices and those of A are all the possible connections between pairs of devices. We denote by n the cardinality of the set V and we suppose that $n \geq 3$. We select a node s to be the source of the communication and a subset R of V that contains all nodes that are supposed to be reached by the signal generated in s . Each node $i \in V$ can receive data from other nodes of the network and send data to any node in its transmission range. With each arc $(i, j) \in A$ we associate the minimum amount p_{ij} of power that must be assigned to node i in order to establish a direct communication with node j . We say that the source node s is *connected* with a destination node $d \in R$, if the transmission powers assigned to the sensors in the network guarantee the existence of a path between nodes s and d .

The Minimum Power Multicast (MPM) problem consists in defining a range assignment function

$$\rho : V \rightarrow \mathbb{R}^+, \quad i \mapsto \rho(i),$$

which assigns to each node $i \in V$ a transmitting power $\rho(i)$, in such a way that the source s can be connected with each destination $d \in R$ and that the sum of the

transmission powers

$$\sum_{i \in V} \rho(i)$$

is minimized. It is important to underline that a wireless network exploits the so-called wireless multicast advantage (WMA). This fundamental property states that all the nodes within the transmission range of a transmitting node receive the signal. In other words several nodes can be covered at the same time by a single transmission.

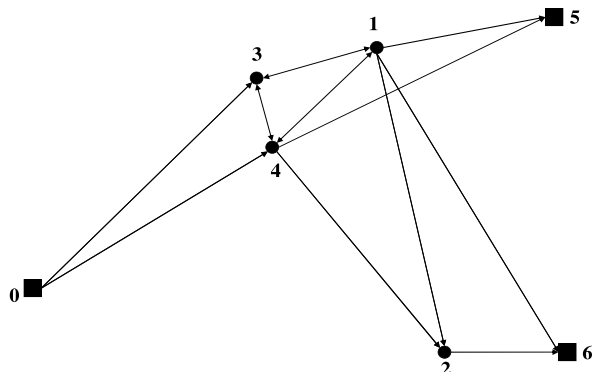
The probabilistic aspect of our problem lies in the fact that each node $i \in V$ is available with a given probability q_i . We assume that for each node $i \in R \cup \{s\}$ it holds $q_i = 1$. Typically the value of q_i will depend on the characteristics of both the node and the area where it is deployed. For example a node i positioned in a dangerous region or in an impervious territory will be assigned a small value of q_i . A consequence of probabilistic node failures is that the connectivity between the source s and all the destinations $d \in R$ can be achieved only with a given probability, which we refer to as *reliability* of the network.

The PMPM problem consists in minimizing the sum of the transmission ranges under the constraint that the reliability of the obtained network reaches a given threshold $\alpha \in [0, 1[$. It is easy to see that the PMPM problem reduces to the deterministic MPM problem if α is set to 0. A more formal definition of the PMPM problem will be given in the next section.

As an example, Fig. 1 depicts the optimal solution of a 7-nodes PMPM instance with one source (node 0) and two destinations (nodes 5 and 6). In this particular PMPM instance the availability q_i of the nodes 1, 2, 3 and 4 has been set to 0.9 and the required reliability to $\alpha = 0.95$. If, for instance, node 4 is affected by failure, nodes 1 and 3 still guarantee the connectivity between the source and the destinations.

It is interesting to remark that the PMPM problem can be formulated mathematically as an integer linear program (see Barta et al. 2011 and Montemanni et al. 2008). However, as shown extensively in Barta et al. (2011), an exact approach to the PMPM problem by means of an integer program formulation is reasonable only for small sized instances.

Fig. 1 Example of a network with 1 source and 2 destination nodes



3 Reliability of a wireless network

Once a transmission range $\rho(i)$ is assigned to each node $i \in V$, only the arcs within the range are considered as available. In other terms, we can define a subset of active arcs $A_\rho = \{(i, j) \in A \mid p_{ij} \leq \rho(i)\}$ and consequently a subgraph $G_\rho = (V, A_\rho)$, which we will refer to as the *topology* generated by the range assignment ρ .

The aim of the PMPM problem is to identify a minimum cost topology, which satisfies the reliability requirement. It is clear that any algorithm based on a search procedure will have to check the reliability value of many different topologies. Therefore the issue of the reliability calculation arises as a crucial aspect of the PMPM problem.

The reliability estimation in the exact algorithm for the PMPM problem IRCG (Iterated Row and Column Generation) presented in Barta et al. (2011) is based on a simple but quite time-consuming sequential search procedure. In this section we present a new technique for an efficient calculation of the reliability value of a topology G_ρ .

3.1 The concept of reliability

In a network model that takes into account probabilistic node failures it is useful to introduce the concept of node configurations. Since each node $i \notin R \cup \{s\}$ is available with a given probability q_i , there are $2^{n-|R|-1}$ different network scenarios that can occur. In the sequel we will refer to any of the possible scenarios as a *configuration*. Formally we define a configuration C as the corresponding set of working nodes and we denote by \mathcal{P} the set of all possible configurations. For instance, in the case of the 7-nodes instance in Fig. 1, configuration $C = \{0, 1, 3, 5, 6\}$ represents the scenario in which the nodes 0, 1, 3, 5 and 6 are available for transmission, while nodes 2 and 4 are affected by failure. The probability of realization of each configuration $C \in \mathcal{P}$, which we denote by $q(C)$, can be easily calculated as

$$q(C) = \prod_{i \in C} q_i \cdot \prod_{i \notin C} (1 - q_i). \quad (1)$$

Therefore the probability of configuration $C = \{0, 1, 3, 5, 6\}$ of the example is $q(C) = 0.9^2 \cdot 0.1^2 = 0.0081$. We remark that, since the source and the destination nodes are assumed to be always available, we will usually not mention them explicitly in the representation of a configuration. So for instance the representation of the previous configuration C will be shortened to $C = \{1, 3\}$.

It is worthwhile to underline that since in a given scenario not all nodes are able to transmit, each configuration C defines a subset of active arcs $A_\rho(C) = \{(i, j) \in A_\rho \mid i \in C\}$, which induces an active topology $G_\rho(C) = (V, A_\rho(C))$. The node configurations $C \in \mathcal{P}$ can be classified as connective or non-connective by means of the following definition.

Definition 1 We refer to a configuration C as *connective* in a topology G_ρ , if for each destination node $d \in R$ there is at least one path in the subgraph $G_\rho(C)$, that connects s with d .

For example, the configuration $C = \{1, 3\}$ considered above is connective, because its active topology $G_\rho(C)$ contains the connecting paths $\{0, 3, 1, 5\}$ and $\{0, 3, 1, 6\}$. In the remainder of the paper the set of the connective configurations of a topology G_ρ will be denoted by \mathcal{C}_ρ . The concept of reliability of a topology can now be defined formally.

Definition 2 Let be G_ρ a given network topology. The *reliability* of the topology G_ρ , denoted by $\text{rel}(G_\rho)$, is the sum of the probabilities of the connective configurations in G_ρ , that is

$$\text{rel}(G_\rho) = \sum_{C \in \mathcal{C}_\rho} q(C), \quad (2)$$

or similarly by referring to the complementary event

$$\text{rel}(G_\rho) = 1 - \sum_{C \in \mathcal{P} \setminus \mathcal{C}_\rho} q(C). \quad (3)$$

In the case of the topology G_ρ shown in Fig. 1 we have that $\mathcal{C}_\rho = \{\{1, 3\}, \{1, 4\}, \{2, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$. Therefore, summing up the probabilities $q(C)$ of the configurations belonging to \mathcal{C}_ρ we obtain the exact reliability $\text{rel}(G_\rho) = 0.972$.

Finally, using the notation that has been introduced, the PMPM problem can be stated as defining a transmission range function ρ on a network G in such a way that the reliability of the induced topology G_ρ is at least α and the sum of the transmission ranges is minimized. More formally we have

$$\begin{aligned} \text{PMPM} \quad & \min \sum_{i \in V} \rho(i) \\ \text{s.t.} \quad & \text{rel}(G_\rho) \geq \alpha. \end{aligned} \quad (4)$$

3.2 Reliability calculation

First of all it is worthwhile remarking that an exhaustive approach, based on the test of the connectivity of each configuration $C \in \mathcal{P}$, is reasonable only for small sized networks. In fact, although the connectivity test of a single configuration is a polynomial flow-problem (see Barta et al. 2011), the total number of configurations grows exponentially with the size of the network and makes an exhaustive enumeration impossible. Preliminary tests confirm that this naive approach is not practical for networks with more than 20–25 nodes.

One main contribution of this paper is an alternative method for the calculation of the reliability $\text{rel}(G_\rho)$, that exploits the structural properties of the set of the configurations \mathcal{P} . Essentially, this method will turn out to be much more efficient than the naive approach, because only a small subset of the configurations has to be tested on connectivity. The following two basic properties of network configurations are the starting point of the proposed new approach.

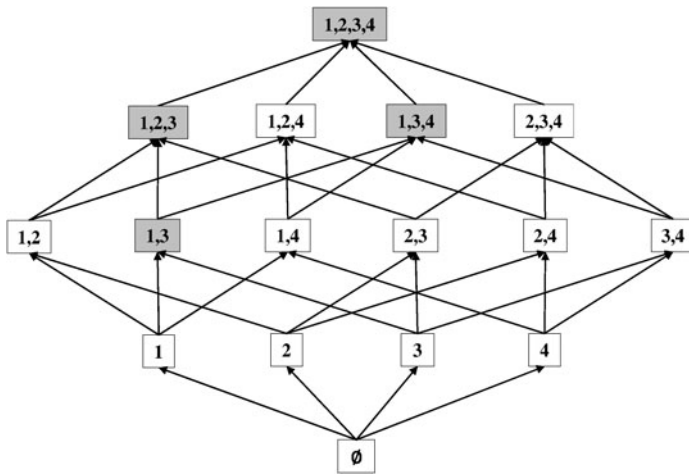


Fig. 2 Representation of an induced set in the Hasse diagram of the configurations

Property 1 (Induced connectivity) Let C be a connective configuration of a topology G_ρ . Any configuration \tilde{C} such that $C \subseteq \tilde{C}$ is also connective in the topology G_ρ .

In the sequel we denote by $I(C)$ the set $\{\tilde{C} \in \mathcal{P} \mid C \subseteq \tilde{C}\}$ of all configurations that include the connective configuration C and we refer to it as the set of the connective configurations induced by C , or shorter as the *induced set* of C . An analogous property can be stated for non-connective configurations.

Property 2 (Induced non-connectivity) Any subset \tilde{C} of a non-connective configuration C is also non-connective.

We denote by $\bar{I}(C)$ the *induced set* of a non-connective configuration C : $\bar{I}(C) = \{\tilde{C} \in \mathcal{P} \mid \tilde{C} \subseteq C\}$. A useful tool for the representation of the above defined induced sets is the so called Hasse diagram, which shows the lattice obtained by the configurations partially ordered by inclusion (for more details see for instance Stanley 1997). Figure 2 depicts the Hasse diagram related to the 7-nodes example of Fig. 1. The 16 possible configurations can be divided in connective and non-connective. It is easy to verify that in our example configuration $C = \{1, 3\}$ is connective. Its induced set $I(C) = \{\{1, 3\}, \{1, 2, 3\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$ contains the configurations that can be reached following the paths in the lattice outgoing from $\{1, 3\}$.

Due to the density of the relations in the Hasse diagram, it is to be expected that an induced set might cover a very large number of configurations. However the sum of the probabilities of the configurations contained in an induced set can be easily calculated without an explicit enumeration.

For any connective configuration C we denote by $Q(C)$ the sum of the probabilities of the configurations contained in the induced set $I(C)$. In an analogous way we refer to $\bar{Q}(C)$ as the sum of the probabilities of the configurations induced by a non-connective configuration C .

Property 3 For any connective configuration C of a topology G_ρ it holds

$$Q(C) = \prod_{i \in C} q_i. \tag{5}$$

Proof Since the induced set $I(C)$ of a connective configuration C contains all configurations having C as a subset, $Q(C)$ is equal to the probability that *at least* all nodes $i \in C$ are available, which is $\prod_{i \in C} q_i$. \square

Property 4 For any non-connective configuration C of a topology G_ρ it holds

$$\bar{Q}(C) = \prod_{i \notin C} (1 - q_i). \tag{6}$$

Proof The induced set $\bar{I}(C)$ of a non-connective configuration C contains all the subsets of C . Consequently, $\bar{Q}(C)$ is equal to the probability that *at most* the nodes $i \in C$ are working, or equivalently that *at least* the nodes $i \notin C$ are affected by failure, which is $\prod_{i \notin C} (1 - q_i)$. \square

Properties 3 and 4 suggest that the reliability of a network topology G_ρ might be obtained by expressing the set of the connective configurations \mathcal{C}_ρ as a union of induced sets $I(C)$, or similarly by expressing the complementary set $\mathcal{P} \setminus \mathcal{C}_\rho$ as a union of non-connective induced sets $\bar{I}(C)$. Following this idea a new problem turns out to be crucial: identifying a minimum number of configurations, such that the union of their induced sets cover the whole set \mathcal{C}_ρ , or respectively $\mathcal{P} \setminus \mathcal{C}_\rho$ in the non-connective case.

Definition 3 We refer to $B \in \mathcal{C}_\rho$ as a *minimal connective configuration*, if no proper subset of B is connective in the topology G_ρ . Similarly, $B \in \mathcal{P} \setminus \mathcal{C}_\rho$ is a *maximal non-connective configuration*, if any configuration strictly including B is connective. In the sequel we denote by \mathcal{B} the set of minimal connective configurations, respectively by $\bar{\mathcal{B}}$ the set of maximal non-connective configurations. In general in the remainder of the paper we will refer to minimal connective configurations or to maximal non-connective configurations also as *basis configurations*.

It is simple to verify that in the case of the topology depicted in Fig. 1 we have $\mathcal{B} = \{\{1, 3\}, \{1, 4\}, \{2, 4\}\}$ and $\bar{\mathcal{B}} = \{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$. According to Definition 3, set \mathcal{B} contains all connective configurations which can not be further reduced without loosing connectivity. This means that any configuration $C \in \mathcal{C}_\rho$ includes at least one basis configuration. It follows immediately that the whole set \mathcal{C}_ρ can be generated via induction from the minimal connective configurations, that is

$$\bigcup_{B \in \mathcal{B}} I(B) = \mathcal{C}_\rho \tag{7}$$

and in the same way we obtain for the non-connective counterpart

$$\bigcup_{B \in \bar{\mathcal{B}}} \bar{I}(B) = \mathcal{P} \setminus \mathcal{C}_\rho. \tag{8}$$

In some sense we can say that basis configurations implicitly contain all connectivity information of the topology G_ρ .

In order to calculate the reliability by means of the covering properties (7) or (8) it is necessary to take into account the overlapping of the induced sets. We overcome this hurdle by the well-known inclusion-exclusion principle of combinatorial theory (see for example Stanley 1997), which enables to calculate the cardinality of the union of several sets or similarly the probability of the union of several events. Given a collection $\{A_1, \dots, A_t\}$ of probabilistic events, the probability of the union $\bigcup_{i=1}^t A_i$ can be expressed as

$$\begin{aligned} \text{prob}\left(\bigcup_{i=1}^t A_i\right) &= \sum_{1 \leq i_1 \leq t} \text{prob}(A_{i_1}) - \sum_{1 \leq i_1 < i_2 \leq t} \text{prob}(A_{i_1} \cap A_{i_2}) + \dots \\ &+ (-1)^{t+1} \sum_{1 \leq i_1 < \dots < i_t \leq t} \text{prob}(A_{i_1} \cap \dots \cap A_{i_t}). \end{aligned} \tag{9}$$

Equation (9) can be adapted to the reliability calculation either through property (7) of the connective configurations or through property (8) of the non-connective configurations. For simplicity, we first present the connective case and then the non-connective variant.

Let be $\mathcal{B} = \{B_1, \dots, B_t\}$ the set of all minimal connective configurations of a given network G . By the covering property (7), the reliability $\text{rel}(G_\rho)$ can be formulated as

$$\text{rel}(G_\rho) = \sum_{C \in \bigcup_{i=1}^t I(B_i)} q(C), \tag{10}$$

which can be interpreted as the probability of the union of the induced sets. Applying the inclusion-exclusion formula (9) the reliability $\text{rel}(G_\rho)$ becomes

$$\begin{aligned} \text{rel}(G_\rho) &= \sum_{1 \leq i_1 \leq t} \left(\sum_{C \in I(B_{i_1})} q(C) \right) - \sum_{1 \leq i_1 < i_2 \leq t} \left(\sum_{C \in I(B_{i_1}) \cap I(B_{i_2})} q(C) \right) + \dots \\ &+ (-1)^{t+1} \sum_{1 \leq i_1 < \dots < i_t \leq t} \left(\sum_{C \in I(B_{i_1}) \cap \dots \cap I(B_{i_t})} q(C) \right). \end{aligned} \tag{11}$$

In order to simplify equation (11) we need to prove a useful property of the intersection of induced sets.

Lemma 1 *Let be $\{C_1, \dots, C_k\}$ a collection of connective configurations. The following property holds*

$$\bigcap_{i=1}^k I(C_i) = I\left(\bigcup_{i=1}^k C_i\right). \tag{12}$$

Proof $C \in \bigcap_{i=1}^k I(C_i) \Leftrightarrow \forall i = 1, \dots, k : C_i \subseteq C \Leftrightarrow \bigcup_{i=1}^k C_i \subseteq C \Leftrightarrow C \in I\left(\bigcup_{i=1}^k C_i\right).$ □

As a direct consequence we obtain for the probability of the intersection of induced sets that

$$\sum_{C \in \bigcap_{i=1}^k I(C_i)} q(C) = \sum_{C \in I(\bigcup_{i=1}^k C_i)} q(C) = Q\left(\bigcup_{i=1}^k C_i\right), \tag{13}$$

which can be easily evaluated by means of Property 3. The reliability formula (11) can now be rewritten in its final form

$$\begin{aligned} \text{rel}(G_\rho) = & \sum_{1 \leq i_1 \leq t} Q(B_{i_1}) - \sum_{1 \leq i_1 < i_2 \leq t} Q(B_{i_1} \cup B_{i_2}) + \dots \\ & + (-1)^{t+1} \sum_{1 \leq i_1 < \dots < i_t \leq t} Q(B_{i_1} \cup \dots \cup B_{i_t}). \end{aligned} \tag{14}$$

Applying (14) to the example shown in Fig. 1 we obtain as expected $\text{rel}(G_\rho) = 3 \cdot 0.9^2 - (2 \cdot 0.9^3 + 0.9^4) + 0.9^4 = 0.972$.

An analogous reliability formula can be derived outgoing from the maximal non-connective configurations. Let be $\tilde{B} = \{B_1, \dots, B_{\tilde{t}}\}$ the set of the maximal non-connective configurations according to Definition 3. The reliability $\text{rel}(G_\rho)$ can also be calculated as

$$\text{rel}(G_\rho) = 1 - \sum_{C \in \bigcup_{i=1}^{\tilde{t}} \tilde{I}(B_i)} q(C). \tag{15}$$

It is not surprising that also Lemma 1 has a symmetric counterpart in the non-connective case:

Lemma 2 *Let be $\{C_1, \dots, C_k\}$ a collection of non-connective configurations. The following equation holds*

$$\bigcap_{i=1}^k \tilde{I}(C_i) = \tilde{I}\left(\bigcap_{i=1}^k C_i\right). \tag{16}$$

Lemma 2 can be proved in a very similar way to Lemma 1. It follows that

$$\sum_{C \in \bigcap_{i=1}^k \tilde{I}(C_i)} q(C) = \sum_{C \in \tilde{I}(\bigcap_{i=1}^k C_i)} q(C) = \tilde{Q}\left(\bigcap_{i=1}^k C_i\right). \tag{17}$$

Therefore we obtain the non-connective variant of the reliability formula (14)

$$\begin{aligned} \text{rel}(G_\rho) = & 1 - \left(\sum_{1 \leq i_1 \leq \tilde{t}} \tilde{Q}(B_{i_1}) - \sum_{1 \leq i_1 < i_2 \leq \tilde{t}} \tilde{Q}(B_{i_1} \cap B_{i_2}) + \dots \right. \\ & \left. + (-1)^{\tilde{t}+1} \sum_{1 \leq i_1 < \dots < i_{\tilde{t}} \leq \tilde{t}} \tilde{Q}(B_{i_1} \cap \dots \cap B_{i_{\tilde{t}}}) \right). \end{aligned} \tag{18}$$

The reliability of the previous example can now be calculated according to equation (18) by means of the maximal non-connective configurations: $\text{rel}(G_\rho) = 1 - (3 \cdot 0.1^2 - (2 \cdot 0.1^3 + 0.1^4) + 0.1^4) = 0.972$.

3.3 BASIS: an algorithm for the reliability computation

The main interest of the reliability formulas (14) and (18) lies in the fact that they can be used to develop an efficient algorithm for the reliability computation of a network. Essentially, the reliability algorithm (BASIS) we propose performs a sequential search of the basis configurations in the Hasse diagram. The search direction will be upwards in the Hasse diagram for connective configurations and downwards for non-connective configurations. Let consider the connective case first.

We enumerate the elements of the set of the configurations \mathcal{P} by building a sequence $\{C_l\}_{l \in \{1, \dots, 2^{n-|R|-1}\}}$ in such a way that $|C_l| \leq |C_{l+1}| \forall l \in \{1, \dots, 2^{n-|R|-1} - 1\}$. In other terms the configurations are ordered by a non-decreasing number of available nodes, which means that the Hasse diagram is visited row by row from the bottom to the top.

Starting from the lowest scenarios in the diagram each configuration is tested on connectivity. As soon as a connective configuration C_l is detected, the algorithm checks whether C_l includes any of the already identified basis configurations B_1, \dots, B_k . If this is not the case, it means that C_l is a new basis configuration and therefore the current set of basis configurations $\mathcal{B} = \{B_1, \dots, B_k\}$ is updated with $B_{k+1} := C_l$.

It has to be remarked that BASIS performs the reliability calculation in a dynamical way. In other terms, if the current basis set \mathcal{B} contains k configurations, the reliability formula (14) is evaluated by setting $t := k$. Since the induced sets of the k available basis configurations cover only a subset of \mathcal{C}_ρ , it is clear that this evaluation provides a lower bound of the reliability $\text{rel}(G_\rho)$. Formally, for any $k \in \{1, \dots, t\}$ we can define an estimate of the reliability

$$LB_k = \sum_{1 \leq i_1 \leq k} Q(B_{i_1}) - \sum_{1 \leq i_1 < i_2 \leq k} Q(B_{i_1} \cup B_{i_2}) + \dots + (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq k} Q(B_{i_1} \cup \dots \cup B_{i_k}). \quad (19)$$

Proposition 1 For any $k \in \{1, \dots, t\}$ the estimate LB_k is a lower bound of the reliability $\text{rel}(G_\rho)$.

Proof

$$LB_k = \sum_{C \in \bigcup_{i=1}^k I(B_i)} q(C) \leq \sum_{C \in \bigcup_{i=1}^t I(B_i)} q(C) = \text{rel}(G_\rho). \quad (20) \quad \square$$

As soon as a new basis configuration B_{k+1} is detected, the current lower bound LB_k is updated to LB_{k+1} by adding to each sum of (19) the terms that involve the new

basis configuration B_{k+1} . It is easy to see that for any $k \in \{1, \dots, t - 1\}$ it holds

$$LB_k < LB_{k+1} \tag{21}$$

since

$$\sum_{C \in \bigcup_{i=1}^k I(B_i)} q(C) < \sum_{C \in \bigcup_{i=1}^{k+1} I(B_i)} q(C). \tag{22}$$

In other words, the lower bounds LB_k form an increasing sequence. Moreover usually good estimates of the exact reliability $\text{rel}(G_\rho)$ can be obtained already in an early phase of the sequential search. A simple explanation of this effect is that the induced sets of the first basis configurations cover a large portion of the set \mathcal{C}_ρ , while the contribution of the subsequent basis configurations becomes more and more marginal because of the overlapping with the previous sets.

As it can be expected from the symmetry of the model, an analogous decreasing sequence of upper bounds $UB_{\bar{k}}$ is obtained through a search of the maximal non-connective configurations in the Hasse diagram. Let be $\bar{\mathcal{B}} = \{B_1, \dots, B_{\bar{t}}\}$ the set of the maximal non-connective configurations. For any $\bar{k} \in \{1, \dots, \bar{t}\}$ we define

$$\begin{aligned}
 UB_{\bar{k}} = 1 - & \left(\sum_{1 \leq i_1 \leq \bar{k}} \bar{Q}(B_{i_1}) - \sum_{1 \leq i_1 < i_2 \leq \bar{k}} \bar{Q}(B_{i_1} \cap B_{i_2}) + \dots \right. \\
 & \left. + (-1)^{\bar{k}+1} \sum_{1 \leq i_1 < \dots < i_{\bar{k}} \leq \bar{k}} \bar{Q}(B_{i_1} \cap \dots \cap B_{i_{\bar{k}}}) \right). \tag{23}
 \end{aligned}$$

Proposition 2 $UB_{\bar{k}}$ is an upper bound of the reliability $\text{rel}(G_\rho)$ for any $\bar{k} \in \{1, \dots, \bar{t}\}$.

Proof

$$UB_{\bar{k}} = 1 - \sum_{C \in \bigcup_{i=1}^{\bar{k}} \bar{I}(B_i)} q(C) \geq 1 - \sum_{C \in \bigcup_{i=1}^{\bar{t}} \bar{I}(B_i)} q(C) = \text{rel}(G_\rho). \tag{24}$$

□

In addition, it is simple to prove that the sequence of upper bounds $UB_{\bar{k}}$ is decreasing. Summarizing, in algorithm BASIS the reliability calculation is carried out by improving simultaneously the lower bound LB_k and the upper bound $UB_{\bar{k}}$. In order to generate a bidirectional search in the Hasse diagram, to any configuration $C \in \{C_l\}_{l \in \{1, \dots, 2^n - |R| - 1\}}$ we associate its *complementary configuration*

$$\bar{C} = V \setminus \{C \setminus \{s \cup R\}\}. \tag{25}$$

The configurations C and \bar{C} are complementary in the sense that the not working nodes of configuration C are available in configuration \bar{C} and, conversely, the not working nodes of configuration \bar{C} are available in C . Therefore it is not difficult to see that in the complementary sequence $\{\bar{C}_l\}_{l \in \{1, \dots, 2^n - |R| - 1\}}$ the number of available nodes is non-increasing. In other words, as the sequence $\{C_l\}_{l \in \{1, \dots, 2^n - |R| - 1\}}$ defines

the upwards exploration of the Hasse diagram required for the lower bound, its complementary sequence $\{\bar{C}_l\}_{l \in \{1, \dots, 2^n - |R| - 1\}}$ defines the downwards exploration needed for the upper bound. Algorithm BASIS can now be described formally:

Step 0. Initialization

Set $l := 1, k := 0, \bar{k} := 0, \mathcal{B} = \emptyset, \bar{\mathcal{B}} = \emptyset, LB_k = 0$ and $UB_{\bar{k}} = 1$.

Step 1. Check configuration C_l

If C_l is connective and $\forall B \in \mathcal{B}$ it holds $B \not\subseteq C_l$, C_l is a new minimal connective configuration. In this case set $k := k + 1, \mathcal{B} := \mathcal{B} \cup \{C_l\}$ and go to Step 2. Otherwise go to Step 3.

Step 2. Lower bound

Calculate LB_k by adding to LB_{k-1} the terms involving the new basis C_l (according to formula (19)). Go to Step 3.

Step 3. Check the complementary configuration \bar{C}_l

If \bar{C}_l is non-connective and $\forall B \in \bar{\mathcal{B}}$ it holds $\bar{C}_l \not\subseteq B$, then \bar{C}_l is a new maximal non-connective configuration. In this case set $\bar{k} := \bar{k} + 1, \bar{\mathcal{B}} := \bar{\mathcal{B}} \cup \{\bar{C}_l\}$ and go to Step 4. Otherwise go to Step 5.

Step 4. Upper bound

Calculate $UB_{\bar{k}}$ by adding to $UB_{\bar{k}-1}$ the terms involving the new basis \bar{C}_l (according to formula (23)). Go to Step 5.

Step 5. Exit criterion

If $UB_{\bar{k}} - LB_k = 0$, the exact reliability has been found. Stop. Otherwise set $l := l + 1$ and go to Step 1.

It is important to remark that the proposed algorithm BASIS always terminates, providing the exact reliability $\text{rel}(G_\rho)$. In fact, as soon as all connective and non-connective basis configurations are identified, by (14) and (19), respectively (18) and (23) it follows

$$LB_l = UB_{\bar{l}} = \text{rel}(G_\rho). \quad (26)$$

On the other hand, if the search procedure is terminated before its natural end, algorithm BASIS provides useful lower and upper approximations of the exact reliability.

3.4 Performance of the algorithm BASIS

In order to test the efficiency of the reliability calculation via algorithm BASIS, we generated a set of random networks G_ρ with up to 100 nodes and we computed their exact reliability values $\text{rel}(G_\rho)$ using both, the basis approach and the naive sequential approach discussed in Sect. 3.2. The computational experiments have been carried out on a computer equipped with a Pentium M (1.5 GHz) processor with 512 MB of memory.

The results obtained are reported in Table 1, which is organized as follows. The first 3 columns contain the parameters of the multicast networks: the number of nodes n , the number of destination nodes $|R|$ and the exact reliability value $\text{rel}(G_\rho)$. Column 4 reports the percentage of the configurations explicitly processed during the execution of algorithm BASIS. Finally, in the last two columns the computational times obtained with the two methods considered are compared.

Table 1 Performance of BASIS compared with the sequential search

n	$ R $	$\text{rel}(G_\rho)$	BASIS		SEQ
			Conf (%)	Seconds	Seconds
30	10	0.8533	0.40	0.01	3.19
30	10	0.9987	0.07	0.01	3.60
30	5	0.8984	0.21	0.17	94.63
30	5	0.9957	4e-03	0.01	104.55
40	20	0.7940	0.38	0.03	3.26
40	20	0.9861	0.07	0.01	5.01
50	25	0.6943	0.03	0.07	112.34
50	25	0.7631	0.04	0.05	120.00
70	40	0.9251	6e-03	0.58	–
70	40	0.9941	4e-03	0.51	–
100	50	0.9652	4e-08	7.95	–
100	50	0.9984	1e-08	3.27	–

The results clearly confirm that the approach with basis configurations is much faster than the sequential approach and that it is able to handle networks with up to 100 nodes. In particular, the percentage of required configurations is extremely low. This is due to the fact that the quality of the reliability estimates improves rapidly as soon as the first basis configurations are identified. Consequently, the bidirectional search in the Hasse diagram is usually able to derive the exact reliability by involving only a very low number of configurations.

As a consequence of the considerable speed up obtained in the reliability calculation, the question arises, whether the performance of the exact algorithm for the PMPM problem presented in Barta et al. (2011) could be significantly improved by plugging in the procedure BASIS. Preliminary tests clearly showed that the speed up obtained on the exact approach is marginal. The main reason for this fact is that the exact algorithm proposed in Barta et al. (2011) is based on an iterative row and column generation procedure, which requires the solution of a long sequence of integer programs. It could be observed that most of the computational time is needed for the solution of the large number of integer programs, which vanishes the benefit of a fast method for the reliability calculation. On the other hand, as it will appear in Sect. 5, the procedure BASIS turns out to be very useful when plugged in a heuristic algorithm not based on the solution of integer programs.

4 A heuristic algorithm for the PMPM problem

4.1 Idea of the algorithm

By means of the fast reliability procedure BASIS described in the previous section it is possible to develop a heuristic algorithm that explores the search space of the network topologies G_ρ and detects feasible solutions for the PMPM problem with a possibly low cost in terms of transmission power.

In this section we present a heuristic algorithm based on the simple idea that the reliability value of a given topology $G_\rho = (V, A_\rho)$ can be increased by adding some new arcs to the set A_ρ , i.e. by increasing the transmission range of given nodes. In an analogous way the reliability decreases, when certain arcs are dropped from the solution.

Essentially, the algorithm we propose consists of two main phases. In the first phase, starting from an infeasible solution, a feasible topology is built by adding new paths that connect the source with one of the destination nodes. We remark that this first phase could alternatively be implemented using an arc insertion strategy instead of path insertion. However, since the connectivity of a configuration depends on the existence of *connecting paths*, it is reasonable to expect that new paths must be added to a topology G_ρ in order to modify its basis configurations and consequently increase its reliability. In our preliminary tests we could observe that the insertion of a single arc often does not affect the connectivity properties of a topology. Therefore, many attempts of arc insertion are needed to achieve a real reliability increase, which turns out to be very time consuming. The adopted choice strategy of the paths to be inserted is essentially random and will be discussed in detail in Sect. 4.2. Once a feasible solution is obtained, the current optimum is updated, if the new topology has a lower objective value.

The second phase of the algorithm consists in the elimination of one or more random arcs from the current topology, in order to reobtain an infeasible solution. It is interesting to remark that in this elimination phase it suffices to delete few arcs, because the reliability value of the solution generated in the first phase is usually little above the threshold α . For this reason in the second phase we implemented the elimination of single random arcs.

Summarizing, the algorithm, that in the remainder of the paper will be called ADDPATH, alternates iteratively the two phases of path insertion and arc elimination, generating a sequence of feasible topologies and thus of upper bounds to the PMPM problem.

4.2 Detailed description

In this section we describe formally the structure of the heuristic ADDPATH and we provide a pseudocode of the algorithm.

4.2.1 Path generation and initial solution

Before starting the iterated local search, it is necessary to create a pool of paths with a low cost in terms of required transmission power, which connect the source with one of the destination nodes.

The procedure that we implemented is based on a downwards exploration of the Hasse diagram of the configurations. For each configuration C encountered during the exploration the algorithm calculates the shortest paths between the source and all destinations $d \in R$ in the active topology $G(C) = (V, A(C))$, where $A(C) = \{(i, j) \in A \mid i \in C\}$. It is easy to see that the top configuration of the Hasse diagram provides the shortest paths for all destinations $d \in R$ and that lower configurations generate

suboptimal paths, since certain nodes are assumed to be unavailable. The procedure is terminated when a requested number $MAXPATHS$ of paths is reached or when a full exploration of the Hasse diagram is completed.

A straight-forward way to generate a suitable initial solution is to overlap the shortest paths obtained on the first configuration $C_0 = V$ and to add further arcs that are covered according to the WMA property. The obtained topology $G_0 = (V, A_0)$ is adequate as initial solution, because it is usually low-cost in terms of power assignment and has, at the same time, a reliability value greater than zero.

4.2.2 Path insertion phase

The aim of the path insertion phase is to generate a feasible topology by addition of new paths selected from the pool. The choice of the path h to be inserted is carried out in the following way: the destination node d of the path is chosen randomly from the set R . Then each path h in the pool between the source s and the destination node d is assigned a weight

$$w_h = \frac{1}{\sum_{(i,j) \in h} P_{ij}} \quad (27)$$

where the denominator contains the transmission power required by path h . Then, a Monte Carlo mechanism based on these weights is used to select a path. In general, the shortest paths will be preferred, however longer paths will also be considered for selection (with a lower probability). This mechanism gives the method the opportunity to explore different paths in a guided random fashion.

Once a path h is identified, the algorithm first checks that h is not already contained in the current topology. It is useful to remark that not only the arcs strictly belonging to the path h are added to the set of arcs A_ρ , but all the arcs covered via WMA property, that is the set of arcs

$$A(h) = \{(i, k) \in A \mid i \in h \wedge \exists j \in h : p_{ik} \leq p_{ij}\}. \quad (28)$$

The path insertion procedure is repeated until the algorithm BASIS detects that $\text{rel}(G_\rho) \geq \alpha$.

4.2.3 Arc retracting

Sometimes it is possible to reduce the cost of the feasible solution G_ρ obtained via path insertion without losing the feasibility. For this purpose we added a so-called arc retracting procedure to the path insertion phase, which first identifies for each node $i \in V$ a longest arc $(i, j_{\max}) \in A_\rho$ outgoing from node i , such that

$$p_{ij_{\max}} \geq p_{ij}, \quad \forall j \in V \text{ and } (i, j) \in A_\rho. \quad (29)$$

Then it checks for each node i , whether the removal of the longest arc (i, j_{\max}) affects the feasibility of the solution G_ρ or not. If the topology remains feasible (in terms of reliability) the arc is deleted from A_ρ and the new longest arc outgoing from node i

is examined. The retracting procedure is terminated when no more longest arc can be removed from the set A_ρ without losing the feasibility of the topology G_ρ .

It is interesting to remark that the information gained from the algorithm BASIS about the basis configurations of the current topology, turns out to be very useful in order to speed up the retracting phase. Indeed, as explained in Sect. 3.2, the reliability $\text{rel}(G_\rho)$ depends only on the source-destination paths induced by the basis configurations. Therefore an arc $(i, j) \in A_\rho$ not belonging to the topology activated by any of the basis configurations, can be regarded as irrelevant in the reliability calculation. In other words a longest arc (i, j_{\max}) can be removed without affecting the reliability value $\text{rel}(G_\rho)$, if the nodes i and j_{\max} do not belong simultaneously to any of the basis configurations. This test can be easily carried out, once the basis configurations are known.

4.2.4 Arc elimination phase

In the case that the feasible topology provided by the retracting phase does not improve the current best solution, the algorithm starts an arc elimination procedure. Substantially, the topology G_ρ is reduced by removing randomly chosen longest arcs, until the reliability value $\text{rel}(G_\rho)$ sinks under the threshold α . This reduced topology will be used as an initial solution in the next iteration. In this way the heuristic AD-DPATH generates a “random walk” in the search space of the PMPM problem.

In order to avoid long unsuccessful random walks, the algorithm controls that the maximum length of an exploration does not exceed a given value (denoted by *MAXEXP* in the following pseudocode). A new exploration starts either when a new best solution has been found, or when the length of the current exploration reaches the allowed maximum. The heuristic ADDPATH terminates providing the best feasible topology, when the allowed computational time is reached.

4.2.5 Pseudocode of algorithm ADDPATH

The algorithm ADDPATH can now be formalized in the following pseudocode:

Step 0. Path generation and initial solution

Generate the pool of connecting paths and an initial solution $G_\rho = (V, A_\rho)$ (according to Sect. 4.2.1). Set $N_{\text{exp}} := 0$, $N_{\text{iter}} := 1$ and $c^* := +\infty$.

If $\text{rel}(G_\rho) < \alpha$ go to Step 1, otherwise set $i := 0$ and go to Step 2.

Step 1. Path insertion

Choose a random path h from the pool (see Sect. 4.2.2).

Add the arcs covered by the path h to the topology G_ρ , that is set $A_{\rho'} := A_\rho \cup A(h)$, $G_{\rho'} := (V, A_{\rho'})$ and $\rho := \rho'$.

If $\text{rel}(G_\rho) < \alpha$ repeat Step 1. Otherwise a feasible topology has been found: set $i := 0$ and go to Step 2.

Step 2. Arc retract

Let be (i, j_{\max}) the longest arc in the topology G_ρ outgoing from node i .

Set $A_{\rho'} := A_\rho \setminus \{(i, j_{\max})\}$, $G_{\rho'} := (V, A_{\rho'})$ (see Sect. 4.2.3).

If $\text{rel}(G_{\rho'}) \geq \alpha$, update $\rho := \rho'$. Otherwise set $i := i + 1$.

If $i < n$ repeat Step 2. Otherwise go to Step 3.

Step 3. *Update and exit*

If $c(G_\rho) < c^*$, update $G_{\rho^*} := G_\rho$, $c^* := c(G_\rho)$ and reset $N_{\text{exp}} := 0$. Otherwise set $N_{\text{exp}} := N_{\text{exp}} + 1$.

If $N_{\text{exp}} == \text{MAXEXP}$, set $G_\rho := G_{\rho^*}$ and $N_{\text{exp}} := 0$.

Stop, if $N_{\text{iter}} == \text{MAXITER}$. Otherwise go to Step 4.

Step 4. *Arc elimination*

Delete a random longest arc $(i, j_{\text{max}}) \in A_\rho$ (as described in Sect. 4.2.4).

Set $A_{\rho'} := A_\rho \setminus \{(i, j)\}$, $G_{\rho'} := (V, A_{\rho'})$ and $\rho := \rho'$.

If $\text{rel}(G_{\rho'}) \geq \alpha$, repeat Step 4. Otherwise G_ρ is infeasible: set $N_{\text{iter}} := N_{\text{iter}} + 1$ and go to Step 1.

5 Computational results

All the algorithms described in this paper have been encoded in ANSI C. Ilog Cplex 12.2 (<http://www.ilog.com/products/cplex>) has been used in the exact algorithm IRCG (see Barta et al. 2011), in order to obtain optimal solutions for a comparison with our heuristic upper bounds.

All the tests reported in this section have been carried out on a computer equipped with an Intel Core i7 2.67 GHz processor and 3 GB of memory.

5.1 Description of the test problems

The computational tests have been carried out on a new benchmark problem set of random Euclidean instances with a size between 10 and 100 nodes. A multicast problem can be characterized by the following parameters:

- n : number of nodes in the network;
- $|R|$: number of destination nodes, i.e. devices that have to receive the messages originated at the source node s ;
- α : reliability level required for the multicast structure;
- q_{\min} : minimum value of the probabilities associated with the devices: $q_i \geq q_{\min} \forall i \in V \setminus \{s \cup R\}$;
- q_{\max} : maximum value of the probabilities associated with devices: $q_i \leq q_{\max} \forall i \in V \setminus \{s \cup R\}$;

The coordinates of the nodes are chosen at random on a 100×100 square grid. Power p_{ij} required to send from node i to node j is obtained according to the relation $p_{ij} = (d_{ij})^\gamma$, where d_{ij} is the Euclidean distance between nodes i and j and the coefficient γ —which models path loss in the signal propagation model—is set to 2.

5.2 Detailed results

For each problem considered, ten random instances have been generated. Table reftable1 and Table 3 report the results obtained by running algorithm ADDPATH on them with a maximum number of 2000, respectively 3000 iterations. Table 4 shows the results obtained on the same instances with a modified version of the algorithm,

Table 2 Performance of the algorithm ADDPATH, 2000 Iterations

n	$ R $	α	Gap (%)		Iterations		#BASIS	Seconds	
			Avg	StDev	Avg	StDev		Avg	StDev
10	3	0.9	0.00	0.00	22	53	15876	0.22	0.06
15	5	0.9	0.00	0.00	83	93	25115	1.44	0.68
20	10	0.9	0.12	0.39	481	518	37632	3.03	2.01
25	20	0.9	5.72	4.75	823	474	64648	1.21	0.41
30	20	0.9	–	–	918	676	62120	4.14	3.88
35	20	0.9	–	–	715	659	68770	43.17	43.10
40	20	0.9	–	–	1073	605	72758	198.33	135.82
50	30	0.9	–	–	1481	437	98682	267.32	183.04
100	80	0.9	–	–	1173	618	216007	469.01	346.60

Table 3 Performance of the algorithm ADDPATH, 3000 Iterations

n	$ R $	α	Gap (%)		Iterations		#BASIS	Seconds	
			Avg	StDev	Avg	StDev		Avg	StDev
10	3	0.9	0.00	0.00	22	53	22786	0.24	0.08
15	5	0.9	0.00	0.00	77	94	37737	2.28	1.31
20	10	0.9	0.11	0.36	548	582	56365	5.11	3.43
25	20	0.9	5.37	4.62	1120	929	97685	1.76	0.59
30	20	0.9	–	–	1143	719	93129	6.64	6.89
35	20	0.9	–	–	920	645	103806	70.71	65.97
40	20	0.9	–	–	1190	669	108803	297.58	202.18
50	30	0.9	–	–	1748	721	147171	404.88	276.20
100	80	0.9	–	–	2354	630	323363	699.97	519.57

which performs 3 runs of 1000 iterations and restarts each time from a new random initial solution. Table 5 and Table 6 highlight the behaviour of the algorithm when the number of destination nodes $|R|$, respectively the reliability threshold α are varied. Finally, Table 7 compares the running times of the heuristic algorithm ADDPATH and of the exact algorithm IRCG.

The tables are organized as follows. The first three columns are devoted to the problem parameters n , $|R|$ and α (see Sect. 5.1). We remark that in our tests the parameters q_{\min} and q_{\max} are fixed to the values 0.85 and 0.95, respectively. In the following columns averages and standard deviations of four crucial indicators are reported. The first indicator is the percentage gap between the heuristic upper bound and the optimal solution, where the optimal solution has been computed with the exact algorithm IRCG described in Barta et al. (2011). As already mentioned, the exact algorithm is not able to solve in reasonable times instances with more than 25 nodes and the lower bounds generated are also very weak. For this reason we reported only the gaps for $n \leq 25$.

Table 4 Performance of the multistart variant, 3×1000 Iterations

n	$ R $	α	Gap (%)		Iterations		#BASIS	Seconds	
			Avg	StDev	Avg	StDev		Avg	StDev
10	3	0.9	0.00	0.00	22	53	22774	0.21	0.09
15	5	0.9	0.00	0.00	75	94	37753	2.37	1.40
20	10	0.9	0.55	1.35	365	380	56447	5.16	3.66
25	20	0.9	5.35	4.95	1027	729	97523	1.76	0.55
30	20	0.9	–	–	1339	960	93522	6.62	7.09
35	20	0.9	–	–	1134	761	103806	69.91	69.12
40	20	0.9	–	–	964	496	108792	299.78	194.10
50	30	0.9	–	–	1512	896	148047	408.88	276.80
100	80	0.9	–	–	1759	1089	324076	691.72	497.98

Table 5 Performance of ADDPATH, number of destinations $|R|$ varied (2000 iterations)

n	$ R $	α	Gap (%)		Iterations		#BASIS	Seconds	
			Avg	StDev	Avg	StDev		Avg	StDev
20	5	0.9	–	–	182	160	28974	22.88	16.28
20	10	0.9	0.12	0.39	481	518	37632	3.03	2.01
20	12	0.9	0.73	1.18	1025	808	38451	1.04	0.54
20	15	0.9	3.29	3.11	1059	567	50060	0.60	0.08
20	17	0.9	4.05	3.69	784	538	56946	0.65	0.10

The indicator *Iterations* corresponds to the number of iterations needed to obtain the best upper bound, the indicator *#BASIS* indicates the total number of calls of the procedure BASIS for the calculation of the reliability during the execution of the heuristic algorithm ADDPATH and, finally, the columns *Seconds* report the averages and the standard deviations of the CPU-times required by the algorithm.

Tables 2 and 3 confirm that the proposed heuristic generates good quality solutions. In fact the average gap observed never exceeds 6% and it tends to decrease when the maximum number of iterations is increased. However, comparing the gaps of Table 2 and Table 3, it can be remarked that the improvement of the gaps is moderate (or in some cases inexistent). This phenomenon can be explained by the fact that usually the substantial improvements of the upper bounds occur in an early phase of the heuristic search procedure. Furthermore each run of the algorithm ADDPATH generates a new sequence of solutions. Therefore in some cases a run over 2000 iterations might give a better result than a run over 3000 iterations. Also the high values of the indicator *Iterations* confirm that the upper bounds might improve if the running time is increased.

In the column *#BASIS* it is worth to observe that the growth of the number of runs of the procedure BASIS is roughly proportional to the size of the problem. This result is achieved by avoiding systematically any unnecessary reliability calculation and it allows a substantial speed up of the whole algorithm.

Table 6 Performance of ADDPATH, reliability threshold α varied (2000 iterations)

n	$ R $	α	Gap (%)		Iterations		#BASIS	Seconds	
			Avg	StDev	Avg	StDev		Avg	StDev
20	10	0.70	1.37	2.13	468	482	34841	3.12	1.49
20	10	0.80	0.99	2.22	859	689	35231	3.24	1.85
20	10	0.85	0.00	0.00	376	446	35718	3.43	1.87
20	10	0.90	0.12	0.39	481	518	37632	3.03	2.01
20	10	0.92	0.00	0.00	241	302	38866	3.41	2.21

Table 7 CPU-times, Exact algorithm IRCG, ADDPATH (2000 Iterations)

n	$ R $	α	IRCG		ADDPATH	
			Avg	StDev	Avg	StDev
10	3	0.9	0.34	0.31	0.22	0.06
15	5	0.9	49.53	66.77	1.44	0.68
20	10	0.8	107.48	122.29	3.24	1.85
20	10	0.9	136.40	155.45	3.03	2.01
20	10	0.92	137.70	202.81	3.41	2.21
25	20	0.9	174.44	149.32	1.21	0.41

The reported running times show that the algorithm is very fast for problems with up to 30 nodes and that it requires in average less than 15 minutes to solve problems with up to 100 nodes. Comparing the running times reported in Table 2 and in Table 3, it is clear that in average they are proportional to the allowed number of iterations. As shown in Table 5, it is interesting to remark that the running times mainly depend on the difference $n - |R|$, because the total number of node configurations is $2^{n-|R|-1}$. For this reason problems with few destination nodes tend to be more difficult for both, the heuristic algorithm ADDPATH and the exact algorithm IRCG. For instance, the missing entry of the gap in Table 5 means that the exact algorithm could not provide the optimal values of this problem.

The results of Table 6 suggest that the performance of the heuristic algorithm ADDPATH is not directly affected by the required reliability level α . This can be explained by the fact that usually the adopted path insertion strategy is able to quickly generate a feasible solution. On the other hand the running times of the exact algorithm IRCG clearly depend on the reliability threshold α : high values of α require the solution of many integer programs, which is extremely time consuming. This effect can be observed in Table 7 when comparing the running times of the 3 problems with the size $n = 20$. In addition Table 7 clearly shows that the heuristic algorithm ADDPATH is much faster than the exact algorithm IRCG based on integer programs. The results reported in Table 4 show that in average the multistart strategy does not produce a relevant improvement in the quality of the solutions. However, on critical instances a multistart approach might be convenient, because each new run of the algorithm can generate a different sequence of solutions.

6 Concluding remarks

A heuristic approach to the probabilistic minimum power multicast problem has been presented in this paper. The algorithm proposed is based on an iterative path insertion and arc elimination procedure, as well as on a fast method for the reliability calculation that exploits some theoretical properties of the model. Experimental results confirm that the heuristic algorithm presented can handle medium to large instances of the PMPM problem providing good quality solutions.

References

- Barta J, Leggieri V, Montemanni R, Triki C, Nobili P (2011) Minimum power multicasting in wireless networks under probabilistic node failures. *Comput Optim Appl* 49(1):193–212
- Barta J, Leggieri V, Montemanni R, Nobili P, Triki C (2010) Some valid inequalities for the probabilistic minimum power multicasting problem. *Electron Notes Discrete Math* 36:463–470
- Bhandari V, Vaidya NH (2007) Reliable broadcast in wireless networks with probabilistic failures. In: *IEEE Infocom proceedings*, pp 715–723
- Bodlaender HL, Wolle T (2004) A note on the complexity of network reliability problems. Technical Report UU-CS-2004-001, Utrecht University
- Cagalj M, Hubaux JP, Enz C (2002) Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In: *Proceedings of the Mobicom 2002 conference*, Atlanta, GA, September 23–28, pp 172–182
- Das AK, Marks RJ, El-Sharkawi M, Arabshahi P, Gray A (2003) Minimum power broadcast trees for wireless networks: integer programming formulations. In: *Proceedings of the IEEE Infocom 2003 conference*, San Francisco, CA, March 30–April 3, vol 2, pp 1001–1010
- Doolin DM, Sitar N (2005) Wireless sensors for wildfire monitoring. In: *SPIE proceedings*, pp 477–484
- Garey MR, Johnson DS (1979) *Computers and intractability—a guide to the theory of NP-completeness*. Freeman, New York
- Guo S, Yang S (2007) Energy-aware multicasting in wireless ad hoc networks: a survey and discussion. *Comput Commun* 30:2129–2148
- Kim S, Pakzad S, Culler DE, Demmel J, Fennes G, Glaser S, Turon M (2007) Health monitoring of civil infrastructures using wireless sensor networks. In: *IPSN proceedings*, pp 254–263
- Leggieri V, Nobili P, Triki C (2008) Minimum power multicasting problem in wireless networks. *Math Methods Oper Res* 68(2):295–311
- Leino J (2002) Optimal multicast routing in ad hoc networks. Technical report
- Liang W (2002) Constructing minimum-energy broadcast trees in wireless ad hoc networks. In: *Mobihoc proceedings*, pp 112–122
- Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J (2002) Wireless sensor networks for habitat monitoring. In: *ACM WSNA proceedings*, pp 88–97
- Min M, Chinchuluun A (2006) Optimization in wireless networks. In: *Resende MGC, Pardalos PM (eds) Handbook of optimization in telecommunication*. Springer, Berlin, pp 891–915
- Montemanni R, Gambardella LM (2005) Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Comput Oper Res* 32(11):2891–2904
- Montemanni R, Leggieri V, Triki C (2008) Mixed integer formulations for the probabilistic minimum energy broadcast problem in wireless networks. *Eur J Oper Res* 190(2):578–585
- Negri L, Zanetti D, Montemanni R, Giordano S (2008) Power-optimized topology formation and configuration in Bluetooth sensor networks: an experimental approach. *Ad-Hoc Sens Wirel Netw* 6(1–2):145–175
- Rosenthal A (1977) Computing the reliability of complex networks. *SIAM J Appl Math* 32:384–393
- Stanley RP (1997) *Enumerative combinatorics*. Cambridge University Press, Cambridge
- Wieselthier J, Nguyen G, Ephremides A (2000) On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: *IEEE Infocom proceedings*, pp 585–594