Relay Placement in Sensor Networks

Jukka Suomela

Helsinki 20 September 2005

M. Sc. Thesis

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author

Jukka Suomela

Työn nimi — Arbetets titel — Title

Relay Placement in Sensor Networks

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level M. Sc. Thesis	Aika — Datum — Month and year 20 September 2005	Sivumäärä — Sidoantal — Number of pages 72 + 2 pages

Tiivistelmä — Referat — Abstract

In this thesis, I study relay placement in energy-constrained wireless sensor networks. The goal is to optimise balanced data gathering, where the utility function is a weighted sum of the minimum and average amounts of data collected from each sensor node. I define a number of classes of simplified relay placement problems, including a planar problem with a simple cost model for radio communication. The computational complexity of these classes is studied, and all classes are proved NP-hard; in some cases even finding approximate solutions is NP-hard. I also present algorithms for finding k-optimal solutions to the relay placement problem. These algorithms have been implemented, and their performance has been studied empirically; the implementation is freely available.

ACM Computing Classification System (CCS):
C.2.1 [Computer-communication Networks]: Network Architecture and Design
F.2.2 [Analysis of Algorithms and Problem Complexity]:
Nonnumerical Algorithms and Problems
G.1.6 [Numerical Analysis]: Optimization
G.2.1 [Discrete Mathematics]: Combinatorics
G.2.2 [Discrete Mathematics]: Graph Theory

Avainsanat — Nyckelord — Keywords

wireless sensor networks, relay placement, balanced data gathering, energy constraints

Säilytyspaikka — Förvaringsställe — Where deposited

Kumpula Science Library, serial number C-

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

In this thesis, I study the problem of placing relay nodes in *wireless sensor networks.* Sensor networks consist of a large number of sensor nodes which collect data. The collected data is routed via the network to a sink node. There may also be relay nodes in the network. The relay nodes only forward data; they are not equipped with sensors. The relay placement problems ask where the relay nodes should be located in order to make data collection efficient. Figure 1 illustrates this problem.

The starting point of this work is the problem of *balanced data gathering* in sensor networks, as presented by Falck *et al.* [FFK$^+$04]. In this formulation, the nodes have a *limited energy source*, and transmitting and receiving data consumes energy. The network keeps forwarding generated data to the sink node until all energy sources are drained. The utility function is a weighted sum of the minimum and average amounts of data gathered from the nodes. The goal is to collect a large total amount of data, but not at the cost of completely ignoring some parts of the monitored area. Falck *et al.* show that the problem of finding an optimal routing can be presented as a linear program.

If the optimum is not satisfactory, one solution could be to add a small number of new relay nodes to the network. The obvious question is how to determine the

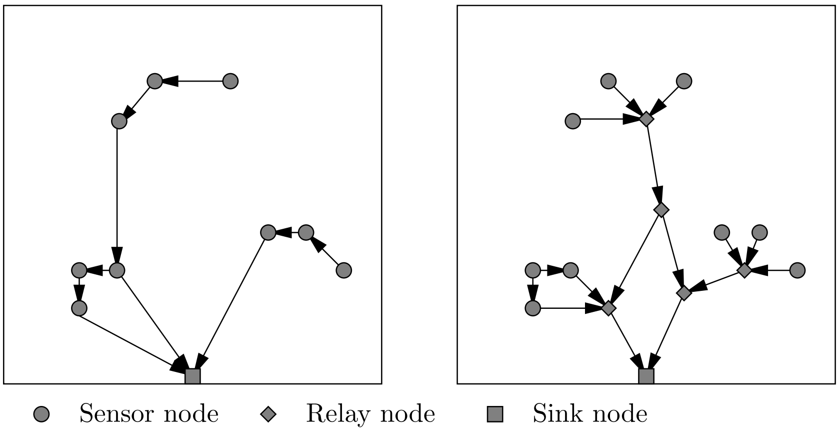


Figure 1: An illustration of the relay placement problem. The first image shows a sensor network and one possible way of routing data in this sensor network. Some long-distance links are unavoidable in this problem instance, and transmitting data over a long-distance link may require a considerable amount of energy, limiting the amount of data collected during the network lifetime. The second image shows how the situation may change after placing five relay nodes.

optimal locations of the relays. This is the *relay placement problem*. Falck *et al.* [FFK$^+$04] consider this problem briefly in the context of balanced data gathering. They present one relay placement method, incremental placement, and compare its performance to that of placing relays in an evenly spaced grid. The incremental method tries to find a local optimum by using greedy heuristics. While this method is computationally efficient, there is no guarantee of optimality.

In this thesis, I will formalise the relay placement problem. I will focus on *special cases* or *simplified versions* of the general relay placement problem. While simplified versions may be directly or approximately applicable to practical problems, they also serve a second purpose: a solution to a simplified problem may provide an upper or lower bound for the utility of a more general problem. For example, if there are additional bandwidth and timing constraints in the more general problem, these simpler problems may provide an upper bound; if the more general problem allows for a limited data aggregation, these simpler problems may provide a lower bound.

Relay placement can be optimised in the sense of *maximising the utility*, given a fixed number of relays, or *minimising the number of relays*, given a target value of the utility function. Both problems and their k*-approximate* versions are studied.

There are two major versions of the problem: the *finite* or discrete problem, and the *planar* or continuous problem. In the finite problem there is a finite set of possible relay locations. In the planar problem the relay locations are not restricted, and a model for radio propagation is needed. I will focus on the *free space model*, where the path loss is a simple function of the distance. In addition to the *Euclidean distance*, I will consider the *Manhattan distance*, which may be a better approximation of radio propagation in typical urban environments [GG93]. I will also present the *line-of-sight model*, which adds geometric constraints to the free space model: the model may contain obstacles, and routing data through obstacles is not possible.

While I will focus on balanced data gathering, much of this discussion can also be applied to more general settings of relay placement. Furthermore, the literature survey of this thesis presents a much wider range of research related to sensor networks; it not only illustrates the applicability of these results, but also presents possible directions for future research. We will see that various aspects of sensor network have been studied separately. However, we do not yet know how to maximise the overall *data quality*, if we consider aspects such as sensor coverage, redundancy of the data, relay placement, routing, radio communication, energy constraints, and monetary costs at the same time.

This work is organised as follows. Section 2 provides background information on sensor networks and the physical laws governing radio communication. The section explains the connection between real sensor networks and the abstract problems presented in this thesis. Section 3 is a literature survey on energy-efficient routing and node placement in wireless sensor networks. Section 4 presents the research questions of this thesis. The solutions are given in the following three sections: I will study the computational complexity of relay placement problems in Section 5, design algorithms for solving the problems in Section 6, and present empirical results in Section 7. Section 8 concludes the thesis. The mathematical symbols used in this thesis are summarised in Appendix 1 for easier reference.

2 Background Information

The relay placement problems presented in this thesis are pure mathematical formulations. They can be studied as such without any background knowledge. However, the models are not arbitrary; they are based on the physics of real sensor networks. This section provides background information which is needed in order to understand this connection.

2.1 Sensor Networks

Sensor networks consist of a large number of tiny nodes. Each node is a very small and cheap computer. Sensing, computation, and communication are combined into a single device. The typical size of a node is in the range of millimetres or centimetres. The energy consumption of a sensor node must be very low. Nodes may be scattered on the field and they may need to operate for years without anyone changing or recharging their batteries.

Sensor networks are typically used for monitoring some area. When interesting events are detected, information is routed from one node to another and eventually gathered in gateway nodes or base stations. Examples of possible uses include environmental and weather monitoring; home automation and air conditioning; monitoring soil in agriculture; tracking goods in commerce and industry; monitoring machines in manufacturing plants; health care and medical diagnostics; intrusion detection and other security systems; and military applications [ASSC02, AK04, CES04].

The creation of a sensor network may involve very little planning. The nodes may even be deployed by dropping them from an aeroplane. The number of sensor nodes may be on the order of thousands [AK04]. Nodes can communicate with each other by using, for instance, radio waves or light. In this thesis, we will focus on radio communication. Typical examples of radio frequency bands proposed for or used in sensor networks are 433 MHz, 915 MHz, and 2.4 GHz [ASSC02].

Sensor networks typically have no centralised control. Nodes need to locate their neighbours autonomously and find the best way of routing data through the network. Thus a sensor network can be seen as an example of an *ad-hoc network*. Unlike the nodes of traditional wireless networks, sensor nodes are not usually moved after deployment [AK04]. However, the topology of the sensor network may change due to failing nodes.

As sensor nodes have computing capabilities, possibilities for sending information from the sensor nodes to the sink are versatile. Some sensor nodes may act as *relays*, forwarding information from one node to another. Nodes may also buffer and summarise information.

Even in one sensor network, different sensor nodes may have very different computing and communication capabilities and energy resources. Hill *et al.* [HHKK04] have written a survey on sensor network platforms. They present various classes of sensor nodes from the simplest asset tags to high-end gateway nodes, and they show examples of real devices in each class. While capabilities may vary, energy consumption and power management are always a central theme when designing applications.

Special-purpose operating systems, such as TinyOS [HSW$^+$00], are used for controlling sensor nodes. In these operating systems, power efficiency and close interaction with the hardware are core issues.

For a general introduction to wireless sensor networks and their applications, see Culler *et al.* [CES04] or Akyildiz *et al.* [ASSC02]. The communication and routing aspects of sensor networks are also studied in the survey by Al-Karaki and Kamal [AK04].

2.2 Energy Consumption of Sensor Nodes

Raghunathan *et al.* [RSPS02] have written a survey on the energy consumption of typical modern sensor nodes. They illustrate energy consumption by two examples, Rockwell's WINS node and MEDUSA-II. For WINS, turning on the radio receiver increases the power consumption from 383 mW to 752 mW, and using the radio transmitter increases the power consumption to the range of 771 mW to 1081 mW, depending on transmitter power. For MEDUSA-II, turning on the receiver increases the power consumption from 10 mW to 22 mW, and using the transmitter increases the power consumption to the range of 19 to 27 mW, depending on transmitter power and other factors.

This means that the radio can consume more power than the other parts of the device combined. Thus, when considering battery lifetime, the radio is a key issue. It has been estimated that transmitting one bit of information may consume as much energy as executing more than a thousand processor instructions [RSPS02]. Performing significant amounts of data processing and computation in order to decrease the amount of radio communication is thus sensible. It is important to understand that energy resources are a hard constraint [Eph02]: when a node runs out of battery, there is *nothing* the node can do anymore.

The nodes used in the examples above can adjust their radio transmission power. We will now consider how much power is needed.

2.3 Radio Wave Propagation

Radio waves are a form of electromagnetic radiation. The physics of electromagnetic waves is well-known. See, for example, Grant and Phillips [GP90] for a basic textbook on electromagnetism. Radio waves are governed by Maxwell's equations and radio wave propagation can be analytically derived for free space and for simple object boundaries.

Rappaport [Rap99, Chapter 3] summarises how object boundaries affect radio propagation. There are three basic mechanisms: Relatively large objects *reflect* radio waves like a mirror reflects a light beam sharply in one direction. Relatively small objects *scatter* radio waves like a rough wall scatters a light beam in all directions. Finally, boundaries of objects *diffract* radio waves.

Both light and radio waves are forms of electromagnetic radiation and they obey the same laws of physics. However, our intuition on visible light fails in radio wave propagation for two reasons. Firstly, the wavelengths are much longer. In modern wireless communications, one typically uses frequencies in the range of hundreds of megahertz to a few gigahertz [ARY95]. The wavelengths are, correspondingly, in the range of centimetres to metres. Secondly, the radiation source typically emits coherent electromagnetic waves. Due to coherence, reflected waves can interfere with each other. Interference can be constructive or destructive.

Among the striking consequences of interference are the so-called small-scale fading effects [Rap99, Chapter 4]. For example, the amplitude of the signal can change rapidly if the receiver is moved by a fraction of a wavelength. As wavelengths are in the range of centimetres or metres, this phenomenon is clearly visible in practical applications. Andersen *et al.* [ARY95] report that the received signal power can vary by a factor of several thousand (30 or 40 dB), and variations by a factor of one hundred (20 dB) are typical. No matter how accurately we estimate the received power, a small error in positioning the transmitter or the receiver can change it by more than two orders of magnitude.

Even if we measured the received power after positioning the transmitter and the receiver, the results could not be applied to future situations. Small scale fading is also caused by the movement of any other object that reflects, scatters or diffracts radio waves. No practical environment is static, and if an artificial environment was static, there would be little point in monitoring it with a sensor network. For example, in an urban environment, people and cars move. On a larger time scale, nature in the winter and in the summer can look different not only from our point of view but also in radio frequencies.

We observe that *any model of radio propagation is inherently inaccurate.* The results are, at best, statistical models estimating the variation of signal strength over time.

2.4 Simple Radio Propagation Models

In this thesis, I will focus on simple, large-scale radio propagation models which are computationally efficient and whose accuracy is relatively good in comparison to the magnitude of typical small-scale fading effects.

Here we are primarily interested in *path loss.* By path loss we will refer to the ratio of the transmitted power to the received power for a given pair of transmitter and receiver locations. If the required reception power is known, the path loss can be used for estimating the required transmitting power. We denote the path loss between locations $\mathbf{x}$ and $\mathbf{y}$ by $PL(\mathbf{x}, \mathbf{y})$. In the telecommunication field, path loss and similar magnitudes are typically measured in decibels (i.e., on a logarithmic scale). However, in this thesis I will use a linear scale in order to make computation more explicit.

ideal waveguide	0
ideal free space	2
simplified earth reflection	4
examples of measured values	1.4 . . . 5.4

Table 1: A summary of both theoretical and empirical exponents for the power law (1).

The classical path loss model is the power law

$$PL(\mathbf{x}, \mathbf{y}) \propto d(\mathbf{x}, \mathbf{y})^{\alpha}, \tag{1}$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|. \tag{2}$$

In free space with no obstacles, the radio path loss accurately obeys the power law, with the value $\alpha = 2$ for the exponent [Rap99, Chapter 3]. In an ideal waveguide with superconducting walls there would be no attenuation at all, and the power law with $\alpha = 0$ would be an accurate model [GP90, Chapter 12]. A bit more realistic example is the case of simplified earth reflection. We assume that the surface of the earth is perfectly reflecting, and that the transmitter and the receiver are relatively close to the ground. By modelling the interference of directly propagating waves and reflected waves, one can estimate the path loss by applying the power law with $\alpha = 4$ [Rap99, Chapter 3].

We have seen that the power law (1) can be used for modelling various idealistic situations. Many researchers have studied how well real environments can be modelled with it by varying the exponent. I present some representative results below. Theoretical and empirical examples are summarised in Table 1.

Andersen *et al.* [ARY95] discuss the planning of cellular networks. Two important concepts in cellular networks are the so-called macrocells and microcells. The former concept refers to cells with a typical radius between 1 and 20 kilometres, while a typical microcell radius is between 0.1 and 1 kilometres. Andersen *et al.* note that applying the power law to macrocell coverage estimation is empirical, but that the results are good. Typical exponents for macrocells are between 2 and 4.

The power law can even be applied indoors. Various studies have fitted the power law to measurements. Seidel and Rappaport [SR92] report empirical exponents in the range of 1.81 to 5.22, and Andersen *et al.* [ARY95] report values in the range of 1.6 to 3.3. Sohrabi *et al.* [SMP99] studied path loss empirically from the point of view of typical sensor network nodes. They focused on near-ground antennas and performed measurements in varying locations both indoors and outdoors. They also fitted the power law to their measured data: the range of exponents was from 1.4 to 5.4.

As an example of the accuracy of the power law, we take a closer look at the results by Seidel and Rappaport [SR92]. For measurements on one floor of one building, standard deviations between 5.2 and 11.2 dB were reported. While a ten-fold error is large, one has to remember that typical small-scale fading is an order of magnitude larger. The power law model is a reasonably good approximation for predicting path loss, given the inherent inaccuracy due to small-scale fading.

Path loss is typically defined as a local average [ARY95]. However, one could equally well define it as an upper bound which holds at a given confidence level. For example, Seidel and Rappaport [SR92] model path loss as a sum of the prediction given by the power law and a log-normally distributed random variable. In a linear scale this corresponds to the product of the predicted value and a random variable. Thus it is easy to calculate the corrective factor needed for a given confidence level.

Finally, we note that the power law can be used for approximating non-polynomial loss functions, too, if the range of distances is relatively small. However, the path loss has to be a function of the Euclidean distance. There is at least one important and practical situation where this does not hold: a typical urban environment with rectilinear streets. Goldsmith and Greenstein [GG93] studied path loss on Manhattan. In their measurements, antennas were located below roof level. They noticed that constant path-loss contours became convex diamonds instead of circles. Erceg *et al.* [ERR94] explain the theoretical background for this phenomenon. No function of the Euclidean distance can describe such a path loss. However, Goldsmith and Greenstein propose using square diamonds as approximations of constant path-loss contours. Note that in this approximation, the path loss is in fact a function of the Manhattan distance.

2.5 Path Loss and Power Consumption

Once we have a model for estimating radio path loss, we can estimate the received power for a given link and a given transmission power. Naturally, a lower received power implies a lower signal quality and a higher bit error rate [ASSC02].

One possible approach is to use a transmitter with adjustable power. One can choose a minimal acceptable received power level, based on signal quality requirements. The product of the required received power and the path loss equals the required transmission power. If the power consumption of the transmitter depends on transmission power, the power consumption is an increasing function of path loss.

Another approach is to use a fixed transmission power. A higher path loss and a constant transmission power imply a larger bit error rate. Thus, re-transmissions are needed more often in order to send data successfully, and each re-transmission consumes power. Again, we see that the average power consumption is an increasing function of path loss.

Ephremides [Eph02] has written an overview on energy concerns in wireless networks. In the article, some general observations are made on the relationship between radio propagation and energy consumption. One important consequence of the physics of radio propagation is that *multihop routing* is often sensible in terms of energy conservation. Multihop routing means sending data from one node to another via relay nodes. There are actually two factors which need to be taken into account: the *total energy consumption* and the *energy consumption balance*.

Firstly, if transmission costs were a true metric in the sense that the triangle inequality was satisfied, multihop routing would have a larger total cost than single-hop routing. However, the squared Euclidean distance does not satisfy the triangle inequality; going from a to c via b may be cheaper than going directly from a to c. As we have already seen, energy consumption may be proportional to the radio path loss, which, on the other hand, may be approximately proportional to the squared Euclidean distance. Thus, multihop routing may be cheaper in terms of total energy consumption.

Secondly, each node has a limited battery capacity and if one battery is drained, the node will no longer work, no matter how little other batteries have been used. Thus, it may be better to consume a small amount of energy at a large number of nodes than a large amount of energy at one node. Multihop routing may help in this respect, too.

3 Review of the State of the Art

In this section, I review the state of the art in the areas which are directly related to the research questions of this thesis. There are two key themes: The first theme, how to route data in sensor networks, is reviewed in Section 3.1. The second theme, where to place network nodes, is reviewed in Section 3.2.

3.1 Routing in Sensor Networks

In this part, we will assume that a sensor network has already been deployed. We assume that the nodes have already determined which neighbouring nodes their radio transmitter can reach, and how costly each radio communication link is. This information, which may be available either centrally or in a distributed manner, describes the *network topology*.

The sensor nodes can generate data, some of which needs to be gathered to the sink nodes. A key issue is determining how to route data packets through the network, in order to maximise some data gathering objective before running out of batteries. Various methods for various needs have been studied in the literature; the most relevant ones will be reviewed in this section. Our emphasis is on principles used for making routing decisions, not on signalling protocols which are used for transferring topology information and for negotiating data flows.

3.1.1 Predetermined Routing

The first option is predetermined routing. A certain set of data flows may be formed when the network is set up. When data packets are created in the sensor nodes, these predetermined flows are used for sending the packets from the sensor nodes to the sink nodes. Routing protocols based on this idea are called *proactive* [AK04].

One may take either a *distributed* or a *centralised* point of view: routing may be planned separately in each node, based on some local subset of information on the network topology; or one may select a coordinating node, collect all topology information in this node, and plan routing based on all available information. In a sensor network, this coordinating node could be the sink.

While making centralised routing decisions may not be practical in many real sensor networks, studying centralised routing serves one important purpose: the globally optimal routing which is based on all available information provides an *upper bound* for any configuration which may be achieved by any means, including distributed algorithms [FFK$^+$04, OK04]. Similarly, the predetermined routing decisions based on a static, ideal network provide an upper bound for the amount of data gathered in real networks where some nodes may fail and some transmissions may be lost. These upper bounds achieved on centralised, static situations may be used for evaluating the performance of distributed and dynamic routing algorithms.

If all network topology information is centrally available, one may formulate the routing as an optimisation problem. One just has to choose the utility function which is maximised. Various approaches have been suggested in the literature.

Among the simplest utility functions is optimising the *total amount* of data gathered. This approach has been taken by, for example, Hong and Prasanna [HP04] and Sadagopan and Krishnamachari [SK04]. Hong and Prasanna formulate the routing task as a constrained flow maximisation problem, and develop a distributed algorithm for solving this problem. Sadagopan and Krishnamachari use a linear programming formulation and develop an efficient approximation algorithm for solving the problem.

Simply maximising the total (or average) amount of data gathered may lead into situations where the gathered data is dominated by a small number of sensor nodes which are close to the sink. Clearly this is not a desired situation if one wants to have representative data from all parts of the monitored area. One possible solution is to limit the amount of data available at each node [HP04, SK04]. A different approach is taken by Ordóñez and Krishnamachari [OK04], who formulate a non-linear optimisation problem where an explicit fairness constraint is included: each node may send at most a given fraction of the data.

Kalpakis *et al.* [KDN03] maximise the lifetime of the network. In their model, this is equal to maximising the *minimum amount* of data gathered from the nodes.

As mentioned in Section 1, Falck *et al.* [FFK$^+$04] formulate the task of *balanced data gathering*: the utility function is a weighted sum of the minimum and average amounts of data gathered from the nodes. This utility function is linear, making it possible to formulate the routing problem as a linear program.

The ability to formulate a problem as a linear program is of both practical and theoretical use: On the one hand, the problem can be solved exactly by using standard LP tools. On the other hand, having an LP formulation bounds computational complexity, as solving LP problems with integer coefficients is possible in polynomial time [Kha79].

Floréen *et al.* [FKKO05a] present an approximation algorithm for optimising balanced data gathering, allowing for a potential speed gain compared to exact LP solvers.

3.1.2 Per-Packet Routing

Instead of planning beforehand, routing may be determined when a data packet has been created or received, using the current information on available battery capacities. This approach makes it possible to adapt to changes in the network topology. Routing protocols where routes are computed on demand are called *reactive* [AK04].

Akyildiz *et al.* [ASSC02] summarise a number of possible approaches: In *maximum available power* routing, the path with the largest total battery capacity is chosen. In *minimum energy* routing, the path with the lowest total energy consumption is chosen. In *minimum hop* routing, the path with the lowest number of hops is chosen. In *maximum minimum available power* routing, one chooses a path on which all nodes have large battery capacities.

There are various heuristic methods which may be used for consuming energy resources more evenly [AK04]. Paths or next hops may be chosen randomly from a set of possible paths, using probabilities which may be derived from energy resources or from network topology. Hop counts and energy constraints may be combined into one distance metric. Routing may be based on data streams, and the number of streams transmitted through each node may be limited.

3.1.3 Query-Based Routing

Routing may be also *data driven* or *query-based*: the sinks may request certain pieces of data, and the sensors will only transmit when they know the piece of information which was requested [ASSC02, AK04]. The sensors may also broadcast messages which describe what kind of data they have.

In data-driven routing, the problem of routing data becomes bidirectional: one needs to send not only measurements from the sensors to the sink but also requests from the sink to the sensors. Requests may be sent using *broadcasting* (one-to-all transmission) or *multicasting* (one-to-many transmission), and responses may be routed back in the reverse direction.

Čagalj *et al.* [ČHE02] study energy-efficient broadcasting in a wireless sensor network. They formulate the problem of finding a minimum-energy broadcast tree, prove the problem NP-complete, and present heuristic algorithms for solving the problem.

However, minimising the total energy consumption does not necessarily maximise the lifetime of the network. This was observed by, for example, Singh *et al.* [SWR98] in their comparison of metrics for energy-aware routing in general ad-hoc networks. Floréen *et al.* [FKKO05b] discuss the problem of maximising network lifetime in wireless multicasting.

There are two major differences between routing queries and routing gathered data: Firstly, a query targeted to n sensor nodes only needs to contain 1 unit of data which will be replicated to all nodes, while gathered data from n sensor nodes contains n units of data, each of which needs to be transmitted to the sink node. Secondly, in wireless networks a single transmission may be simultaneously received by multiple nodes at no additional cost. This is the so-called *wireless multicast advantage*. Both issues are illustrated in Figure 2. In the next section, we will see how data aggregation may make it possible to forward not only queries but also some additional data for no additional cost.

3.1.4 Data Aggregation

Instead of forwarding all data packets unaltered from sensors to sinks, intermediate nodes may also *aggregate* and *summarise* data [ASSC02]. Sensor nodes may monitor neighbouring and even partially overlapping areas, and generated data may be highly redundant. Furthermore, instead of individual measurements, one might be interested in collecting statistical information: averages, medians, minimums, and maximums of measured values, or the total number of detected events. Even if all of the gathered data needs to be transmitted to the sink, buffering may be used for fitting more data in the smallest logical transmission unit if delays are not an issue.

In order to facilitate data aggregation, the logical network topology of sensor networks may be *hierarchical*. Nodes may be clustered and each cluster can do data reduction [PHC$^+$03, AK04]. Sensor networks may even contain higher-energy nodes which are responsible for aggregating and forwarding data generated in a local cluster. Information on physical node locations may be used to form hierarchies [AK04].

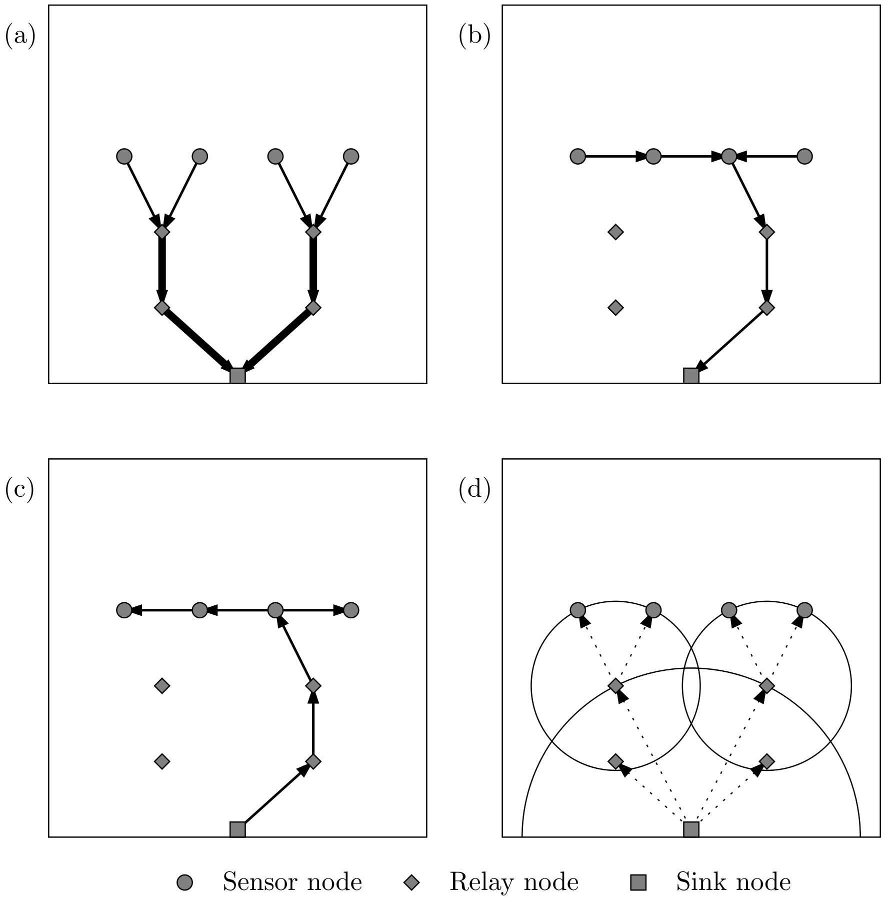


Figure 2: An illustration of different approaches to routing data and queries in a wireless sensor network. On the top row, data gathering is illustrated. Figure (a) illustrates the kind of data gathering setting which is the primary topic of this thesis. All data entering a relay node needs to be forwarded and no data aggregation is possible. Figure (b) illustrates how data aggregation may change the situation. If a node needs to transmit some data, forwarding some additional data may be possible for no additional cost. On the bottom row, query broadcasting is illustrated. Figure (c) show how a single query from the sink may be broadcast to all sensor nodes. Here no wireless multicast advantage is used; multicasting is performed by a number of point-to-point transmissions. The similarity with Figure (b) is evident. However, the lifetime-optimal solution is not necessarily the same. Transmissions in different directions use batteries on different devices. In Figure (d) wireless multicast advantage is taken into account: a single transmission may be simultaneously received by a number of nodes.

Krishnamachari *et al.* [KEW02] study the problem of routing with data aggregation in wireless sensor networks. They note that the problem of finding the minimum-energy routing with data aggregation is essentially the same problem as finding minimum-energy multicast trees; only the roles of transmitting and receiving nodes are reversed. Both problems may be seen as instances of the minimum Steiner tree problem. This observation is also illustrated in Figure 2. The similarity applies to multicasting with point-to-point connections only; there is no analogue of wireless multicast advantage in data aggregation.

As mentioned above, minimising the total energy consumption does not guarantee maximising the network lifetime. The lifetime maximisation problem by Kalpakis *et al.* [KDN03] (see Section 3.1.1 above) is designed specifically for sensor networks in which data aggregation is possible.

3.2 Node Placement in Sensor Networks

We have seen how one may try to improve the performance of a sensor network after the network has been deployed. Now I will review some studies which aim at improving the deployment process.

One of the main reasons for using a sensor network instead of one sensor is to be able to monitor a large area. Thus the *coverage* of a sensor network is a key factor when considering node placement in sensor networks. I will first show how coverage may be defined, after which I will review approaches to optimising node placement in order to maximise coverage. Finally, we will take radio communication and energy constraints into account.

3.2.1 Defining Coverage

There are various ways of measuring coverage. First we need a *sensor model* which describes which areas are covered by a given sensor at a given position. The sensor model may be *deterministic*, specifying an area which is completely covered by the sensor, or more realistically *probabilistic*, specifying the probability of detecting a target or phenomenon at a given location. In order to give a more realistic estimate of sensing capabilities, the sensor model may take obstacles into account.

Once a sensor model is given, we may define the coverage of the entire sensor network. If the sensor model is deterministic, an obvious measure for coverage is the area which is covered by at least one sensor. For probabilistic models we may use, for example, the global minimum of detection probabilities.

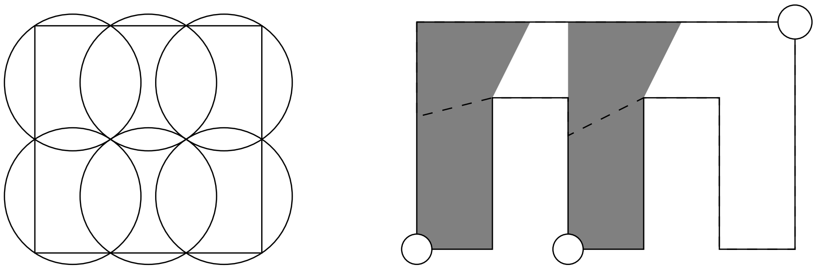
However, these approaches do not take into account the nature of the monitored phenomenon. For example, in a surveillance network we may be interested in maximising the probability of detecting an object which is passing through the monitored field. Thus, the coverage metric may be the strength of the *maximal breach path* [MKPS01], where the closest distance to any sensor at any point on the path is minimised, or the strength of the *minimum exposure path* [MKQP01], where the integral of the sensor intensity along the path is minimised. The former follows the line segments of a Voronoi diagram, but this does not hold for the latter definition.

3.2.2 Optimising Coverage

One of the simplest possible sensor models is a *unit disk*: the sensor has a fixed range, and the area within this range is completely covered. When coverage is measured as the area which is covered by at least one sensor, the best possible coverage is obtained when every point of the monitored area is covered by at least one sensor. Finding such a sensor deployment is equivalent to finding a way of covering an area with circles. For example, Melissen and Schuur [MS96] and Nurmela and Östergård [NÖ00] have studied the problem of covering a square with equal circles, and Nurmela [Nur00] has studied covering an equilateral triangle with equal circles.

Covering a square with equal circles is illustrated in Figure 3 (a). The solution given is simple but not optimal: it is actually possible to cover the square by using circles with a 0.6 % smaller radius [MS96].

Another simple approach assumes infinite sensor ranges but takes obstacles into account. This is directly related to so-called *illumination* or *art gallery* problems. In illumination problems, one studies questions such as how many light sources are needed in order to illuminate the interior of a given polygon. Art gallery problems rephrase the same question by asking how many guards are needed in order to guard the interior of an art gallery with a given floor plan. We may interpret light sources or guards as sensor nodes, and use the results from the field of illumination and art gallery problems in order to find sensor node placements. See, for example, Urrutia [Urr00] for a comprehensive survey on illumination and art gallery problems. In addition to traditional art gallery problems, Urrutia also surveys *floodlight* problems, where light sources have a limited angle of illumination. These results may be di-



(a) (b)

Figure 3: Examples of covering problems. Figure (a) illustrates a solution to the problem of covering a square with six equal circles. Figure (b) illustrates a solution to an art gallery problem. The small circles correspond to the positions of three vertex guards, and the grey colour illustrates the areas seen by two of these guards.

rectly applicable to, for example, various motion-detection sensors, as they typically have a limited angle of detection.

Figure 3 (b) illustrates an art gallery problem of finding *vertex guards*, i.e., guards positioned on polygon vertices, that can monitor the interior of an *orthogonal polygon*. In this example, there are 12 vertices in the polygon, and guarding it requires three vertex guards. Kahn *et al.* [KKK83] proved that guarding the interior of an orthogonal polygon is always possible with $\lfloor n/4 \rfloor$ vertex guards, where n is the number of vertices, and that this bound is tight.

Research related to optimising radio transmitter coverage is also relevant, as similar problems arise there. In that area, various node placement methods have been studied in cases where simple geometrical models are no longer applicable. In addition to traditional optimisation algorithms, approaches like neural networks [SE96] and genetic algorithms [ABN$^+$02] have also been proposed.

Coverage may also be optimised by using a distributed method. Howard *et al.* [HMS02] present a mechanism which is based on potential fields: virtual forces repel nodes from each other and from obstacles. This method is designed for mobile sensor networks, where each sensor is an autonomous robot. While this kind of distributed method cannot guarantee an optimal coverage, it may offer a way of finding a reasonable coverage by using only a limited amount of computation, communication, and coordination.

Finding optimal locations for sensor nodes may not be enough in practical applications. When nodes are deployed by, for example, dropping from aeroplanes, the exact locations of sensor nodes are at worst completely random, and even in the best case only approximately what was planned. Zou and Chakrabarty [ZC03] have taken this fact into account when designing their algorithms for sensor network deployment. They have developed heuristic methods which try to maximise the probability of detecting targets at any location, given the fact that both detection and sensor deployment are probabilistic. In their work they have studied a detection model in which the detection probability decreases exponentially with distance, and sensor locations are random variables with Gaussian probability distributions.

3.2.3 Optimising Coverage by Sensor Scheduling

In addition to finding node locations which cover the monitored area, one may also take a deployed sensor network as a starting point and try to find subsets of sensor nodes which would still cover the entire field. If it is possible to find multiple, mutually exclusive subsets of sensor nodes, where each subset is able to monitor the entire area, one may use this information to conserve energy. Only the sensors of one such subset need to be active at any point of time. This leads into a *schedule* which specifies when a certain node needs to be active.

Slijepcevic and Potkonjak [SP01] have developed algorithms for finding this kind of subsets in sensor networks. They first find fields which are parts of the monitored area covered by the same set of sensor nodes. Then they present a heuristic algorithm for finding mutually exclusive subsets which cover all of these fields.

3.2.4 Optimising Radio Communication and Energy

Covering an area with sensors is not enough; we also need to transmit data from the sensors to the sink. Thus, we are faced with the problem of optimising coverage while keeping in mind the limitations and costs of radio communication. As mentioned in the previous section, similar approaches may be used for optimising both sensor coverage and radio coverage. However, our primary goal here is not to cover the monitored area by wireless connectivity, but to form paths for forwarding data.

Pan *et al.* [PHC$^+$03] study the problem of placing the base station (sink node) in an optimal location. Pan *et al.* study so-called two-tiered networks, where sensor nodes are deployed in clusters, the sensor nodes of each cluster transmit data to a local application node, and the application node transmits data to the base station. They focus on optimising data transmission between application nodes and the base station, and we may use their results in one-tiered networks by interpreting application nodes as sensors. They assume that the locations of the sensor nodes and the application nodes are fixed, but the base station can be moved. When each application node transmits data directly to the base station, finding a lifetime-optimal location of the base station is a relatively simple problem of computational geometry. Pan *et al.* consider variations of the concept of lifetime and study how relaying data between application nodes changes the situation.

Falck *et al.* [FFK$^+$04] study the problem of adding relay nodes in the balanced data gathering setting described above. The locations of the sensor nodes are assumed to be fixed, and the goal is to find good locations for a small number of relay nodes in order to optimise balanced data gathering. They develop a greedy heuristic algorithm for relay placement and compare its performance to that of placing relays in a fixed grid.

Dasgupta *et al.* [DKK03] take more freedom: they assume that all nodes may be moved. They study how to place sensor nodes so that energy consumption is minimal but the sensors still cover the monitored area. They present a heuristic algorithm which minimises energy consumption by moving nodes and by changing the role assignments of nodes: some nodes are assigned the role of a relay node, while others are assigned the role of a sensor.

4 Problem Formulations

In this section, I will define the relay placement problems and present the research questions which will be studied in this thesis. First, I will define the problems formally. Section 4.1 contains the definitions and Section 4.2 presents the research questions which will be studied. Then, I will justify these choices. Section 4.3 compares these questions to the literature, and Section 4.4 compares the formal problems to real sensor networks.

4.1 Definitions of the Problems

I will begin by introducing the balanced data gathering problem, following the formulation by Falck *et al.* [FFK$^+$04]. Then I will use the balanced data gathering problem to define various relay placement problem classes. The symbols used in the definitions are summarised in Appendix 1, and the problem classes are summarised in Table 2. In this section, the definitions are presented as formal models with little

Type:	Decision		Definition 3
	Relay-constrained optimal		Definition 4
	Relay-constrained k-optimal		Definition 4
	Utility-constrained optimal		Definition 5
	Utility-constrained k-optimal		Definition 5
Utility:	Balanced data gathering		Definition 1
Possible relays:	Unrestricted		Definition 2
	— Planar	$\mathcal{P}_P$	Definition 6
	— Finite set	$\mathcal{P}_D$	Definition 7
	— Sensor upgrade	$\mathcal{P}_U$	Definition 8
Transmission costs:	Unrestricted		Definition 2
	— Location-dependent	$\mathcal{P}_L$	Definition 9
	— Line-of-sight	$\mathcal{P}_S$	Definition 10
	— Free space	$\mathcal{P}_F$	Definition 11
Batteries:	Unrestricted		Definition 2
	— Identical	$\mathcal{P}_I$	Definition 12

Table 2: A summary of relay placement problems. Indentation refers to models which are special cases of less restrictive models.

justification. Section 4.4 explains the practical use of these models.

In this thesis, the set of non-negative real numbers is denoted by $[0, \infty)$. The set of non-negative extended real numbers, $[0, \infty) \cup \{+\infty\}$, is denoted by $[0, \infty]$. The set of positive real numbers is $(0, \infty)$ and the set of positive extended real numbers is $(0, \infty]$. If g is a function $X \to Y$, and A is a subset of X, we will use $g_{|A}$ to denote the restricted function $h\colon A \to Y$ which maps $a \in A$ to $g(a)$.

4.1.1 Balanced Data Gathering Problem

An instance of the balanced data gathering problem is a tuple $B = (\lambda, S, R, \sigma, E, s, \tau, \rho)$. Here $\lambda \in [0, 1]$ is the balance parameter, S is a finite set of sensor nodes, R is a finite set of relay nodes, and σ is the sink node. The sets S, R and $\{\sigma\}$ are disjoint. Let S^+ denote the set of the sensor and sink nodes, i.e., $S^+ = S \cup \{\sigma\}$ and let V denote the set of all nodes, i.e., $V = S^+ \cup R$. The function $E\colon V \to [0, \infty]$ specifies the battery capacity of each node. The function $s\colon S \to [0, \infty]$ specifies how much data is available at each sensor node. The parameter $\rho \in [0, \infty)$ is the cost of receiving one unit of data for all nodes, and the function $\tau\colon V \times V \to [0, \infty]$

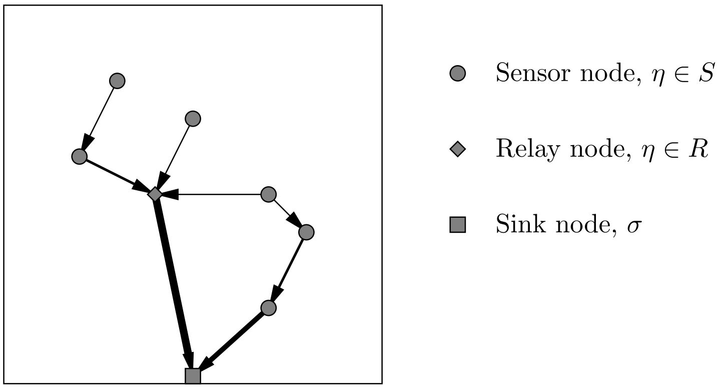


Figure 4: An instance of the balanced data gathering problem. Arrows show one possible solution, the widths of the arrows illustrating the amount of data flowing from one node to another. This figure also illustrates that nodes may divide the gathered data and send parts of it to different nodes.

maps a pair of nodes to the cost of sending one unit of data from the first node to the second one. We will use the notation $E_\eta = E(\eta)$, $s_\eta = s(\eta)$, and $\tau_{\eta\kappa} = \tau(\eta, \kappa)$ to emphasise that these functions can be represented as vectors and a matrix. An instance of the balanced data gathering problem is illustrated in Figure 4.

The solutions to the problem are flows. A *feasible* flow f is a matrix which satisfies the following equations:

$$\begin{aligned}
f_{\eta\kappa} &\geq 0, & \forall \eta, \kappa \in V, \quad && (3)\\
f_{\eta\eta} &= 0, & \forall \eta \in V, \quad && (4)\\
\sum_{\kappa \in V} f_{\sigma\kappa} &= 0, & && (5)\\
\sum_{\kappa \in V} (f_{\eta\kappa} - f_{\kappa\eta}) &\in [0, s_\eta], & \forall \eta \in S, \quad && (6)\\
\sum_{\kappa \in V} (f_{\eta\kappa} - f_{\kappa\eta}) &= 0, & \forall \eta \in R, \quad && (7)\\
\sum_{\kappa \in V} (\tau_{\eta\kappa} f_{\eta\kappa} + \rho f_{\kappa\eta}) &\leq E_\eta, & \forall \eta \in V. \quad && (8)
\end{aligned}$$

Here $f_{\eta\kappa}$ is the amount of data transmitted from the node $\eta \in V$ to the node $\kappa \in V$. Negative flows are physically meaningless and they are forbidden by equation (3). Equation (4) expresses that there is never need for a flow from a node to itself. Equation (5) requires that the sink node transmits nothing. Equations (6) and (7) require that the sensor and relay nodes forward all incoming data to other nodes. The sensor nodes may also generate a limited amount of new data, while the relay nodes never generate anything. Equation (8) formulates the energy constraint: the batteries must contain enough energy to transmit all outgoing data and receive all incoming data. We will use the symbol $\mathcal{F}(B)$ to denote the set of all feasible flows for the problem instance B.

The value $q_\eta(B, f)$ denotes the amount of data gathered from a node $\eta \in S$, given a flow f:

$$q_\eta(B, f) = \sum_{\kappa \in V} (f_{\eta\kappa} - f_{\kappa\eta}) \,. \tag{9}$$

If a flow is feasible, all of this data is transmitted to the sink node before the batteries of the nodes are drained. The utility of the flow, $F(B, f)$, is a weighted sum of the minimum and average amounts of data gathered:

$$F(B, f) = \lambda \min_{\eta \in S} q_\eta(B, f) + (1 - \lambda) \operatorname*{avg}_{\eta \in S} q_\eta(B, f). \tag{10}$$

Now we are ready to define the problem of optimising balanced data gathering.

Definition 1 (The balanced data gathering problem). An instance of the balanced data gathering problem is a tuple $B = (\lambda,\, S,\, R,\, \sigma,\, E,\, s,\, \tau,\, \rho)$ satisfying the conditions above. The set of all such tuples is $\mathcal{B}$. The solution is any feasible flow $f \in \mathcal{F}(B)$ and the utility of the solution is $F(B, f)$. An optimal solution is any flow $f^* \in \mathcal{F}(B)$ which maximises $F(B, f^*)$. A k-optimal solution is any flow $\tilde{f} \in \mathcal{F}(B)$ which satisfies $F(B, \tilde{f}) \geq \frac{1}{k} F(B, f^*)$.

4.1.2 Relay Placement Problem

An instance of the relay placement problem is a tuple $P = (\lambda,\, S,\, \mathcal{R},\, \sigma,\, E,\, s,\, \tau,\, \rho)$. Here λ, S, σ, E, s, and ρ correspond to the parameters of the balanced data gathering problem. Instead of the set of relays, the set of *possible* relays, $\mathcal{R}$, is given. Again, we will require that the sets S, $\mathcal{R}$, and $\{\sigma\}$ are disjoint, and we will define the set of possible nodes $\mathcal{V} = S^+ \cup \mathcal{R}$. The battery capacity function $E(\eta)$ must be defined for all possible nodes $\eta \in \mathcal{V}$, and the transmission cost function $\tau(\eta, \kappa)$ must be defined for all pairs of possible nodes $\eta, \kappa \in \mathcal{V}$. An instance of the relay placement problem is illustrated in Figure 5.

The solution is a finite subset R of possible relays $\mathcal{R}$. Given a relay placement instance P and its solution R, we can define the corresponding balanced data gathering

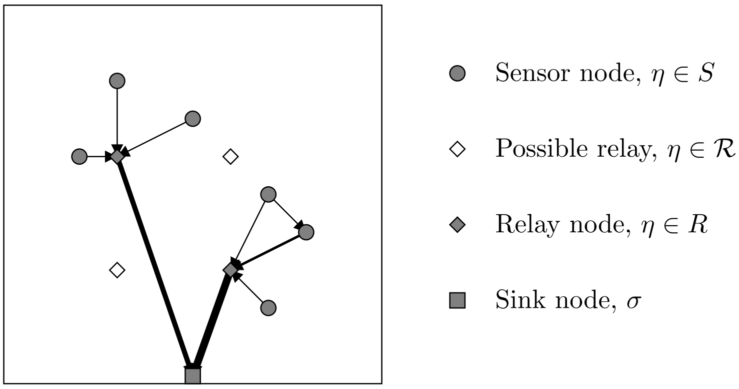

Sensor node, $\eta \in S$
Possible relay, $\eta \in \mathcal{R}$
Relay node, $\eta \in R$
Sink node, σ

Figure 5: An instance of the relay placement problem. In this problem instance the set of possible relays was finite; there were 4 possible relays. Grey relays (2 in this example) illustrate one possible solution to the relay placement problem. Arrows illustrate corresponding data flows in the underlying balanced data gathering problem.

instance $B = B(P, R)$ where

$$B(P, R) = (\lambda, S, R, \sigma, E_{|V}, s, \tau_{|V \times V}, \rho), \tag{11}$$

$$V = S^+ \cup R. \tag{12}$$

We define the utility of this solution, $U(P, R)$, as the highest possible balanced data gathering which can be achieved:

$$U(P, R) = \max_{f \in \mathcal{F}(B)} F(B, f), \qquad B = B(P, R). \tag{13}$$

Now we can summarise the key definitions.

Definition 2 (The relay placement problem). An instance of the relay placement problem is a tuple $P = (\lambda,\ S,\ \mathcal{R},\ \sigma,\ E,\ s,\ \tau,\ \rho)$ satisfying the conditions above. The set of all such tuples is $\mathcal{P}$. The solution is any finite set of relays, $R \subseteq \mathcal{R}$. The utility of the solution is $U(P, R)$.

Note that while the goal is to choose a subset of possible relays, the problem is referred to as the relay *placement* problem. This choice becomes more clear later when we introduce the concept of a location and discuss practical problems.

We can now formulate the decision version of the relay placement problem.

Definition 3 (The decision problem). An instance of the decision problem is a tuple (P, N, u) where $P \in \mathcal{P}$ is a relay placement problem instance, N is the number of relays, and u is the utility requirement. The answer to the decision problem is *yes* if and only if there is a solution R to the relay placement problem P such that $|R| = N$ and $U(P, R) \geq u$.

We will now define two optimisation problems. In the relay-constrained problem we will optimise the utility by using a fixed number of relays. On the other hand, in the utility-constrained problem we will try to find the minimum number of relays with which a utility constraint can be satisfied.

Definition 4 (The relay-constrained problem). An instance of the relay-constrained problem is a pair (P, N) where $P \in \mathcal{P}$ is a relay placement problem instance and N is the number of relays. The solution is any $R \in \mathcal{R}$ with $|R| = N$. A solution R^* is optimal if it maximises $U(P, R^*)$. A solution $\tilde{R}$ is k-optimal if it satisfies $U(P, \tilde{R}) \geq \frac{1}{k} U(P, R^*)$.

Definition 5 (The utility-constrained problem). An instance of the utility-constrained problem is a pair (P, u) where $P \in \mathcal{P}$ is a relay placement problem instance and u is the utility requirement. The solution is any $R \in \mathcal{R}$ with $U(P, R) \geq u$. A solution R^* is optimal if it minimises $|R^*|$. A solution $\tilde{R}$ is k-optimal if it satisfies $|\tilde{R}| \leq k|R^*|$.

Now the relay placement problem is formally defined. As mentioned in the introduction, the main focus of this thesis is on simplifications of the problem.

4.1.3 Simplified Problems

Without loss of generality, we can always assume that $\mathcal{V}$ is a subset of $\mathbb{R}^2 \times W$ for some set W. (If this was not the case for some set $\mathcal{V}'$, we could choose $W = \mathcal{V}'$, and injectively map each $\eta \in \mathcal{V}'$ to $((0, 0), \eta) \in \mathbb{R}^2 \times W$.)

We will now define two functions:

$$l \colon \mathbb{R}^2 \times W \to \mathbb{R}^2 \colon (l, w) \mapsto l, \tag{14}$$

$$w \colon \mathbb{R}^2 \times W \to W \colon (l, w) \mapsto w. \tag{15}$$

Note that our assumption guarantees that $l(\eta)$ and $w(\eta)$ are defined, when η is a sensor node, a relay node, a sink node, or just a possible relay node. Things get more interesting, when we interpret that $l(\eta)$ describes the *location* of the node on the real plane, and $w(\eta)$ describes the *identity* of the node. By using different identities, we can have more than one possible relay at each location; thus we can also have a solution with multiple relays at the same location. We can also use identities to model different kinds of relays in cases where it is not clear which kind of relay is ideal at each location.

Let us now see how we can simplify the model. First we will show three models for restricting the set of possible relays, $\mathcal{R}$. In the first model, we assume that the relays can be placed anywhere and that all possible relays have the same identity.

Definition 6 (Planar model). A problem instance $P \in \mathcal{P}$ is planar, denoted by $P \in \mathcal{P}_P$, if the set of possible relays $\mathcal{R}$ is $\mathbb{R}^2 \times \{y\}$ for some identity y.

The second model only allows for finite sets of relays. The definition is obvious.

Definition 7 (Finite relay set). A problem instance $P \in \mathcal{P}$ has a finite relay set, denoted by $P \in \mathcal{P}_D$, if $\mathcal{R}$ is finite.

The third model is the sensor upgrade model. Here we require that for each sensor location, there is exactly one possible relay in the same location. Note that here the set of possible relays is also finite, i.e., $\mathcal{P}_U \subset \mathcal{P}_D$.

Definition 8 (Sensor upgrade model). A problem instance $P \in \mathcal{P}$ uses the sensor upgrade model, denoted by $P \in \mathcal{P}_U$, if the set of possible relays $\mathcal{R}$ is $l(S) \times \{y\}$ for some identity y.

Next we will focus on restricting the transmission costs, τ. First, we will require that radio transmission costs depend only on locations.

Definition 9 (Location-dependent model). A problem instance $P \in \mathcal{P}$ has location-dependent transmission costs, denoted by $P \in \mathcal{P}_L$, if $\tau(\eta, \kappa) = \tau'(l(\eta), l(\kappa))$ for some function τ'.

Now we will present two concrete location-dependent models where the transmission costs are explicitly parametrised. First, we introduce the parameter O, which is a finite set of disjoint obstacles. Each obstacle $o \in O$ is a simple (not self-intersecting) polygon in the real plane. Each obstacle o is described by enumerating its vertices. The number of vertices in the obstacle is $\#o$, and the total number of vertices in all obstacles is $\#O = \sum_i \#o_i$.

In the line-of-sight model we require that the transmission cost $\tau'(l_1, l_2)$ is infinite if the line segment $\overline{l_1 l_2}$ intersects some obstacle $o \in O$. Furthermore, we require that there are two parameters, the path loss exponent $\alpha \in [0, \infty)$ and the norm parameter $p \in (0, \infty]$. If the line segment does not intersect any obstacle, we require that

$$\tau'(l_1, l_2) = d_p(l_1, l_2)^\alpha, \tag{16}$$

where $d_p(\cdot, \cdot)$ denotes the distance measured using the p-norm:

$$d_p(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p, \tag{17}$$

$$\|\mathbf{z}\|_p = \left(\sum\nolimits_i |z_i|^p\right)^{\frac{1}{p}} \quad \text{for finite } p, \tag{18}$$

$$\|\mathbf{z}\|_\infty = \max\nolimits_i |z_i|. \tag{19}$$

Definition 10 (Line-of-sight model). A problem instance $P \in \mathcal{P}_L$ uses the line-of-sight model, denoted by $P \in \mathcal{P}_S$, if the transmission cost model can be defined by some parameters α, p, and O as described above.

Furthermore, we can remove the obstacles from the model:

Definition 11 (Free space model). A problem instance $P \in \mathcal{P}_S$ uses the free space model, denoted by $P \in \mathcal{P}_F$, if $O = \emptyset$.

Now we have described how the set of possible relays $\mathcal{R}$ and the transmission cost τ can be restricted. Finally, we will see how battery capacities E can be restricted. In the identical battery model, all relays have the same battery capacity. It should be noted that this does not restrict the batteries of the sensor and sink nodes.

Definition 12 (Identical batteries). A problem instance $P \in \mathcal{P}$ has identical batteries, denoted by $P \in \mathcal{P}_I$, if there is an E such that $E(\eta) = E$ for all possible relays $\eta \in \mathcal{R}$.

Now the simplified relay placement problems have been defined. We can construct a problem description by choosing one definition from each class presented in Table 2. One could, for example, focus on the relay placement problems with the sensor upgrade model, free space transmission costs, and identical batteries. In other words, one could focus on problems in the subset $\mathcal{P}_U \cap \mathcal{P}_F \cap \mathcal{P}_I$. Furthermore, one could concentrate on finding k-optimal solutions to the utility-constrained problem for some k.

In order to further simplify our notation, we will denote $\mathcal{P}_x \cap \mathcal{P}_y$ by $\mathcal{P}_{xy}$, etc. For instance, $\mathcal{P}_U \cap \mathcal{P}_F \cap \mathcal{P}_I$ can be denoted by $\mathcal{P}_{UFI}$. One can construct a total of 32

Type:	Decision	N, u	$O(1)$
	Relay-constrained	N	$O(1)$
	Utility-constrained	u	$O(1)$
Utility:	Balanced data gathering	$\lambda, S, \sigma, E(S^+), s(S), \rho$	$O(\lvert S\rvert)$
Relays:	Unrestricted	$\mathcal{R}$	$O(\lvert\mathcal{R}\rvert)$
	Planar	w	$O(1)$
	Finite set	$\mathcal{R}$	$O(\lvert\mathcal{R}\rvert)$
	Sensor upgrade	w	$O(1)$
Costs:	Unrestricted	$\tau(\mathcal{V}, \mathcal{V})$	$O(\lvert\mathcal{V}\rvert^2)$
	Location-dependent	$\tau'(\mathcal{V}, \mathcal{V})$	$O(\lvert l(\mathcal{V})\rvert^2)$
	Line-of-sight	α, p, O	$O(\#O)$
	Free space	α, p	$O(1)$
Batteries:	Unrestricted	$E(\mathcal{R})$	$O(\lvert\mathcal{R}\rvert)$
	Identical	E	$O(1)$

Table 3: The parameters for the relay placement problem with their asymptotic worst-case sizes. The sizes are calculated in terms of how many real numbers are needed to define a problem instance.

relay placement problem classes from Table 2. We will denote the set of these classes by $\mathcal{P}^*$, i.e.,

$$\mathcal{P}^* = \{\mathcal{P}, \mathcal{P}_P, \mathcal{P}_D, \mathcal{P}_U, \ldots, \mathcal{P}_{DFI}, \mathcal{P}_{UFI}\}, \tag{20}$$

and we will use $\mathcal{P}_x$ to refer an arbitrary member of $\mathcal{P}^*$.

4.1.4 Parametrisation of Problems

As we are interested in solving the problems computationally, we will have a look at how to describe the instances of the problems. The required parameters are collected in Table 3. The asymptotic worst-case size of each parameter is described in terms of the number of values. Continuing with our example, the full description of a $\mathcal{P}_{UFI}$ problem instance would consist of a tuple $P = (\lambda,\ S,\ \sigma,\ E(S^+),\ s(S),\ \rho,\ \alpha,\ p,\ E)$. The total description length would be $O(|S|)$, dominated by the need to enumerate all sensor nodes with their battery capacities and amounts of available data. We will denote the full description length of a problem instance by $|P|$. Again, this quantity is the number of values required to represent the instance.

All combinations do not necessarily have finite or even countably infinite descriptions. However, there are two large and interesting subsets of problems which do have a finite description. The first case is a relay placement problem with a finite relay set, i.e., the set $\mathcal{P}_D$. No other restrictions are needed. Now $\mathcal{R}$ and $\mathcal{V}$ are finite and the worst case description length is $O(|\mathcal{V}|^2)$. The length is dominated by the need to enumerate transmission costs between all pairs of possible nodes.

The second case is the planar line-of-sight model with identical batteries, i.e., the set $\mathcal{P}_{PSI}$. With these three restrictions, the description length is $O(|S| + \#O)$. The length is dominated by the need to enumerate all sensor nodes with their parameters, and the need to describe all obstacles.

So far, we have assumed that we can represent arbitrary real numbers. In this thesis, we will make the following simplification: *we restrict our discussion to problem instances whose parameters are rational numbers.* For example, in planar problems node locations will have rational coordinates; in unrestricted problems the transmission cost matrix will consist of rational numbers. By definition, rational numbers can be expressed as a pair of integers, (a, b). By following these conventions, one needs $O(|P|)$ integers to represent a problem instance. The number of bits is thus $O(|P| \log x)$, where x is the largest integer value needed in the problem description.

In the line-of-sight and free space models, even if node locations are rational, the transmission cost matrix is not necessarily rational. However, in two common special cases, $\alpha = 2$, $p = 2$ and $\alpha = 2$, $p = 1$, it turns out that the transmission costs are sums of squares of two rational numbers and squares of sums of two rational numbers, respectively:

$$d_2(\mathbf{x}, \mathbf{y})^2 = \Big(\Big(\sum_i |x_i - y_i|^2\Big)^{\frac{1}{2}}\Big)^2 = \sum_i |x_i - y_i|^2, \tag{21}$$

$$d_1(\mathbf{x}, \mathbf{y})^2 = \Big(\Big(\sum_i |x_i - y_i|^1\Big)^{\frac{1}{1}}\Big)^2 = \Big(\sum_i |x_i - y_i|\Big)^2. \tag{22}$$

Thus, in those special cases the implicitly defined transmission cost matrix is rational. If the cost matrix is rational, the LP formulation of the underlying balanced data gathering problem is also rational.

4.2 Research Questions

Now we are ready to present the research problems which will be studied in this thesis:

1. Analyse the computational complexity of each relay placement problem class $\mathcal{P}_x \in \mathcal{P}^*$.

2. Design algorithms which solve a subset of the relay placement problems k-optimally.

4.3 Relation to the State of the Art

The balanced data gathering task is a recent formulation and, as seen in the review of the state of the art above, it has not yet been thoroughly studied. There are efficient algorithms for solving the data gathering problem accurately and approximately. However, the problem of relay placement in the balanced data gathering setting has only been studied by Falck *et al.* [FFK$^+$04]. Only one class of relay placement problems has been formulated, the computational complexity of the problem has not been analysed rigorously, and no approximation algorithms have been presented. Thus, to my knowledge, the problem classes defined in this section are new, and the research questions have not yet been answered.

We can see from the literature survey in Section 3 that approaches to optimising node placement are, in general, of a heuristic nature [SE96, ABN$^+$02, HMS02, DKK03, ZC03, FFK$^+$04]. There is seldom any guarantee of the quality of the solution. Developing k-approximation algorithms for relay placement in the balanced data gathering setting may also give us new ideas for solving more general node placement problems.

It should be noted that while our formulation of balanced relay gathering is compatible with Falck *et al.* [FFK$^+$04], it differs slightly from the formulation in the more recent article by Floréen *et al.* [FKKO05a]. In the latter presentation, the possibility to limit the amount of available data, i.e., the s parameter, is missing. Floréen *et al.* do not need to handle this restriction separately because their discussion is not tied to a particular radio geometry. Thus, the amount of available data can be restricted by replacing the sensor with a pair of nodes, one sensor node and one relay node, and adding a communication link of unit cost from the sensor to the relay node. This way, the battery capacity of the relay node acts as a limitation to the available data. However, as our problem formulations include, among others, the free-space model, this kind of conversion is no longer possible, and the amount of available data needs to be expressed explicitly.

4.4 Relation to Sensor Networks

Our problem formulation is applicable to practical situations where the performance of a battery-powered sensor network needs to be improved. We assume that the sensor locations have already been determined: a human expert may have chosen the best sensor locations, an optimal deployment plan may have been calculated by optimising sensor coverage, or a sensor network may have already been deployed and sensor locations have been determined after deployment. The relay-constrained optimisation problem answers the following practical question: now that we have N relays, where do we place them? On the other hand, the utility-constrained optimisation problem takes a different point of view: now that we know how much data is required, how many relays do we need and where do we place them? If we do not want to install new relay nodes but just want to install larger batteries in some sensor nodes, we may use the sensor upgrade model.

We assume that data reporting is time-driven: there is no need to transmit queries from the sink to the sensor. We also assume that topology information is available in a central location and that routing can be predetermined. Thus, the topology of the sensor network should be stable. While some of these assumptions may not hold in practise, the presented results can still be used for obtaining upper bounds for the network utility, which is useful in benchmarking dynamic, distributed algorithms.

Our method focuses on cases where data aggregation is infeasible. This method can be used in situations where each sensor may generate unique information. The presented model is suitable for heterogeneous networks where some nodes have larger battery capacities than others.

Maximising the total amount of data gathered is a special case of the balanced data gathering problem, and the results of this thesis are also applicable to this special case. However, the balanced data gathering setting helps with making sure that each network node is actually used in monitoring while still allowing more data to be gathered from areas where it is cheaper in terms of energy consumption.

In the balanced data gathering formulation, one node may send parts of its data to multiple different nodes. In practical sensor networks, this corresponds to scheduling transmissions so that the total amount of data transmitted over each link approximately matches the desired solution.

As was seen in Section 2, the free space model is compatible with approximate models for radio propagation in real environments, both indoors and outdoors. Being able to use an arbitrary norm makes it possible to also approximate diamond-shaped constant path-loss contours which are typical in urban environments with rectilinear streets. When contours are not symmetric, one may scale the entire plane in one direction by a constant factor. The line-of-sight model may be applied to, for example, optical communication methods where an unobstructed visibility is required.

We assume that not only the path loss but also the transmitter energy consumption can be modelled by a power of the distance between two nodes. While this approximation does not necessarily hold strictly, it should be noted that the methods presented in this thesis may also be adapted to more complicated transmission cost functions. Multipath delay spread and similar factors are assumed to be included in the transmission cost model. Interference from other links is ignored as we are focusing on the case of long lifetime and limited batteries. In those cases, transmissions are infrequent and collisions should be rare.

5 Computational Complexity

In the following sections, the research questions presented in Section 4.2 are answered. We will begin by analysing the computational complexity of the relay placement problem. We will present two different reductions from well-known NP-complete problems to a number of relay placement problem classes. The first reduction shows that all classes of relay placement problems are NP-hard. The second reduction shows that even approximate versions of some problem classes are NP-hard.

These two reductions are based on very different ideas, and studying them may give us additional insight into the computational complexity of the relay placement problems: In the first reduction we optimise the average amount of data collected, while in the second one we optimise the minimum amount of data collected from each node. The first one depends on large, carefully chosen distances, while the second one uses a specific geometry of obstacles. In the first one we use the Manhattan distance, and the second one demonstrates using the Euclidean distance. Furthermore, while both reductions are presented by using a radio path loss exponent of $\alpha = 2$, the proofs may be easily adapted to different radio transmission cost models. Thus, computational difficulties may arise from different aspects of relay placement problems.

5.1 Reducing Partition to Relay Placement

In this section, we will prove that the relay placement problem classes $\mathcal{P}_{UFI}$ (the sensor-upgrade problem with the free space model and identical batteries) and $\mathcal{P}_{PFI}$ (the planar problem with the free space model and identical batteries) are NP-hard. We will show this by developing a polynomial reduction from Partition, which is a well-known NP-complete problem [Kar72, GJ03].

In the Partition problem, one is asked to divide a set of integers into two sets with equal sums. We may assume that the integers are positive [GJ03]. Formally:

Definition 13 (Partition). An instance of the problem Partition consists of a list of positive integers, $(a_1, \dots, a_n)$. A set $X \subseteq \{1, 2, \dots, n\}$ is a feasible solution if $\sum_{i \in X} a_i = \sum_{i \notin X} a_i$.

We will now develop a polynomial reduction from Partition to $\mathcal{P}_{UFI}$ and $\mathcal{P}_{PFI}$. The same construction applies to both cases. Let a list of positive integers, $(a_1, \dots, a_n)$, be given. We will assume that the sum of the integers is even; otherwise the answer to the problem would be trivially *no*. Construct a relay placement problem instance P as follows. First, define:

$$a^* = \max a_i, \tag{23}$$

$$b = \frac{1}{2} \sum a_i. \tag{24}$$

Note that b is a positive integer. Choose $\lambda = 0$ so that we are interested in optimising the average amount of data. Then, choose the following radio propagation model:

$$p = 1, \tag{25}$$

$$\alpha = 2, \tag{26}$$

$$\rho = 0. \tag{27}$$

Choose any values satisfying the following inequalities:

$$z \geq (na^*)^{\frac{1}{\alpha}}, \tag{28}$$

$$y \geq z + 1, \tag{29}$$

$$x \geq ny. \tag{30}$$

Construct the problem geometry as shown in Figure 6. Firstly, there are 2 sensors of type 1, η and η', with the following characteristics:

$$\begin{aligned} E(\eta) &= E(\eta') = bx^\alpha, & (31)\\ s(\eta) &= s(\eta') = b, & (32)\\ l(\eta) &= +(z/2 + 1/2 + x/2, -z/2 - 1/2 - x/2), & (33)\\ l(\eta') &= -(z/2 + 1/2 + x/2, -z/2 - 1/2 - x/2). & (34) \end{aligned}$$

Then, there are n diagonal rows of nodes, each row corresponding to one integer in the Partition problem. We will first define the centre points of these rows:

$$l_i = ((2i - n - 1)y/2, (2i - n - 1)y/2). \quad (35)$$

On each row, there are two sensors of type 2, κ_i and κ'_i, with the following characteristics:

$$\begin{aligned} E(\kappa_i) &= E(\kappa'_i) = a_i, & (36)\\ s(\kappa_i) &= s(\kappa'_i) = 0, & (37)\\ l(\kappa_i) &= l_i + (z/2 + 1/2, -z/2 - 1/2), & (38)\\ l(\kappa'_i) &= l_i - (z/2 + 1/2, -z/2 - 1/2). & (39) \end{aligned}$$

Furthermore, on each row there are two sensors of type 3, μ_i and μ'_i, with the following characteristics. The only purpose of these nodes is to act as possible relay locations in the sensor upgrade model:

$$\begin{aligned} E(\mu_i) &= E(\mu'_i) = 0, & (40)\\ s(\mu_i) &= s(\mu'_i) = 0, & (41)\\ l(\mu_i) &= l_i + (z/2, -z/2), & (42)\\ l(\mu'_i) &= l_i - (z/2, -z/2). & (43) \end{aligned}$$

Finally, on each row there is one sensor of type 4, ν_i, with the following characteristics:

$$\begin{aligned} E(\nu_i) &= z^\alpha, & (44)\\ s(\nu_i) &= 1, & (45)\\ l(\nu_i) &= l_i. & (46) \end{aligned}$$

The location of the sink is $l(\sigma) = (x/2 + y, x/2 + y)$ and the battery capacity of the sink is irrelevant as the reception cost is zero. All relays have a battery capacity

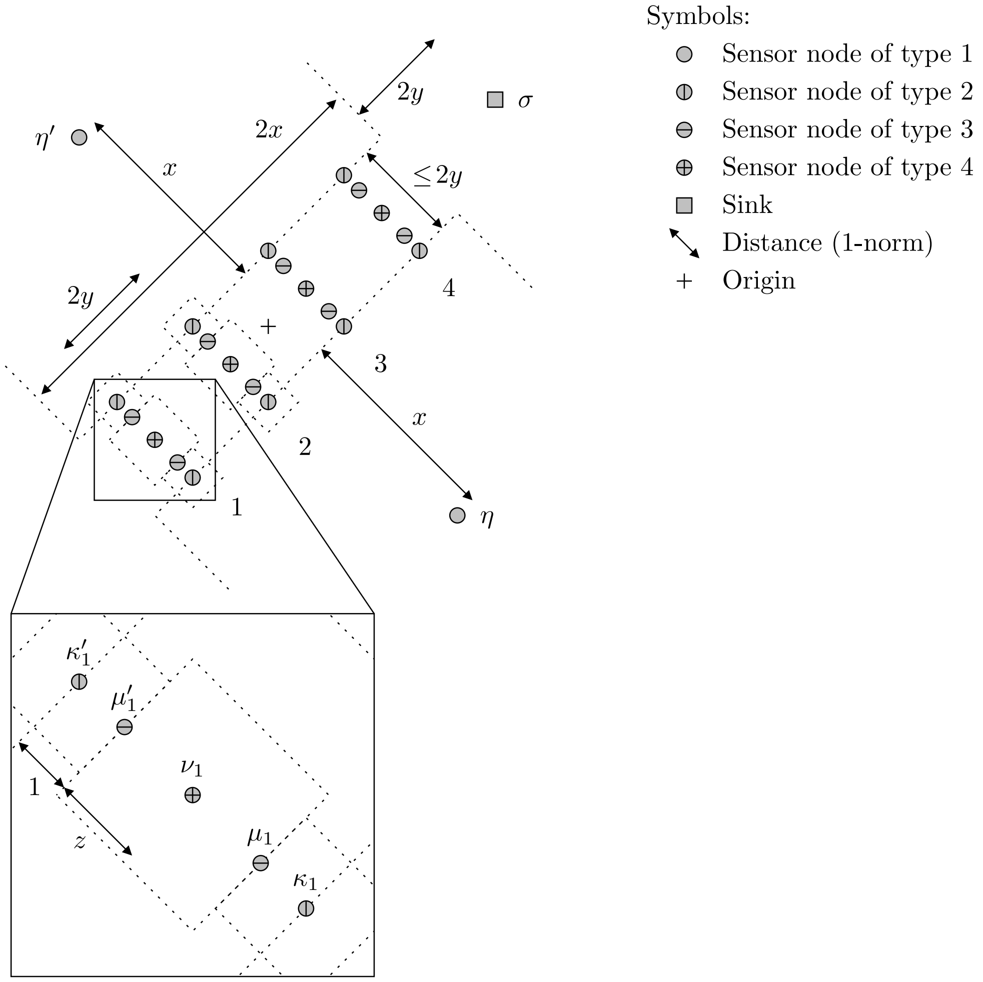


Figure 6: Reduction from Partition to $\mathcal{P}_{UFI}$ and $\mathcal{P}_{PFI}$. In this example, the corresponding Partition problem instance consisted of four integers. In the relay placement problem instance, diagonal rows labelled with numbers 1–4 correspond to the four integers in the Partition problem instance. One of the rows is shown in a larger scale.

satisfying $E \geq (a^* + 1)(2x + 2y)^\alpha$. One can use any rational constant satisfying this inequality.

The total number of sensor nodes is $m = 5n + 2$, and the total amount of available data is $2b + n$ units. The utility of any solution is thus at most $U^* = (2b + n)/m$.

We can now formulate the following decision problem instance (see Definition 3 on page 23): P is the relay placement problem instance constructed above, the number of relays N is n, and the utility requirement u equals U^*. We will show that this formulation is indeed a polynomial reduction from PARTITION to $\mathcal{P}_{UFI}$.

Lemma 1. *Constructing the problem instance is possible in polynomial time.*

Proof. We may choose $z = na^*$, $y = z + 1$, $x = ny$, and $E = (a^* + 1)(2x + 2y)^\alpha$. The total number of nodes in the constructed problem is $O(n)$. The parameters of each node can be calculated in polynomial time: keeping in mind that $\alpha = 2$, all expressions above only involve integer or rational numbers, and the size of each integer is polynomial in the size of the input. □

Lemma 2. *If the answer to the* PARTITION *problem instance is* yes*, the answer to the relay placement problem instance constructed above is* yes*, both in the $\mathcal{P}_{UFI}$ and in the $\mathcal{P}_{PFI}$ formulation.*

Proof. Let X be a feasible solution to the PARTITION problem. Denote the set $\{1, \ldots, n\} \setminus X$ by X'.

For each $i \in X$, place a relay on the type 3 node μ_i. For each $i \in X'$, place a relay on the type 3 node μ'_i. This is possible both in the sensor upgrade model and in the planar model. This way we have placed a total of n relays. Figure 7 illustrates this relay placement and the corresponding flow.

For each $i \in X$, transmit a_i units of data from η to κ_i. The distance is x, and each transmission uses $a_i x^\alpha$ units of energy. This way we can transmit a total of b units of data from η by using bx^α units of energy. Thus, all available data was transmitted from η, and the battery had enough capacity. Next, for each $i \in X$, forward a_i units of data from κ_i to the relay at μ_i. The distance is 1, and the transmission cost is thus a_i, again we have enough battery capacity for this transmission. Furthermore, transmit 1 unit of data from ν_i to the relay at μ_i, using z^α units of energy. Now, we have accumulated $a_i + 1$ units of data on the relay at μ_i, which can be transmitted directly to the sink. The distance is at most $2x + 2y$, and the transmission cost is thus at most $(a^* + 1)(2x + 2y)^\alpha$, which does not exceed the battery capacity.

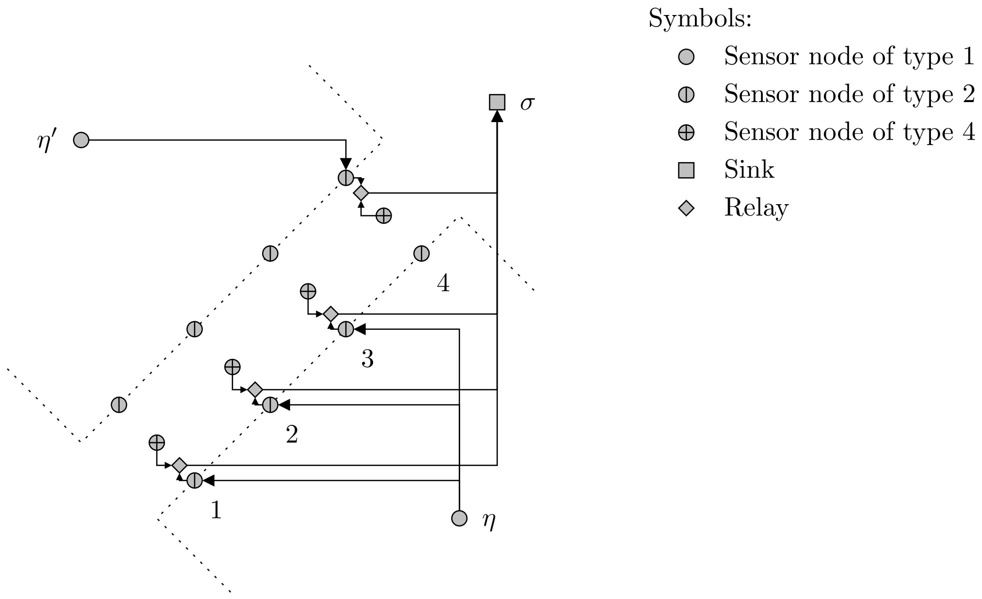


Figure 7: This figure illustrates a solution to the relay placement problem shown in Figure 6. Here we assume that $X = \{1, 2, 3\}$ is a solution to the corresponding Partition problem. This figure shows how to place 4 relays, relays 1–3 on the right side and relay 4 on the left side. For clarity, type 3 nodes are not shown as they do not have any energy resources and thus cannot contribute to routing data. Flows are shown with rectangular arrows in order to make it more clear that transmission costs use the Manhattan distance.

Similarly, for each $i \in X'$, transmit a_i units of data from η' to κ'_i and forward it to the relay at μ'_i. Then, transmit 1 unit of data from ν_i to the relay at μ'_i. Finally, send all data from the relay to the sink.

Thus, the flow is feasible. All available data is forwarded to the sink, and the value of the utility is U^*. Thus, the answer to the relay placement problem instance is *yes*. □

Lemma 3. *If the answer to the* Partition *problem instance is* no*, the answer to the relay placement problem instance constructed above is* no*, both in the $\mathcal{P}_{UFI}$ and in the $\mathcal{P}_{PFI}$ formulation.*

Proof. Let us first assume that the answer to the relay placement problem instance is *yes*. This is possible only if all available data from all sensor nodes of type 1 and type 4 is forwarded to the sink node.

Let us first study a sensor of type 4, say ν_i. If it sends all its data to nodes whose distance is more than z units from the sensor node, it would require more than z^α units of energy, which is not available. Thus, the sensor has to send at least some of its data to a node whose distance is at most z units. Furthermore, it cannot send data to nodes of type 3 as those nodes have no energy capacity for forwarding the data. No other nodes are available within this area: the distance to the type 2 node on the same diagonal is $z + 1$ units, distances to nodes on other diagonals are at least $2y \geq 2z + 2$ units, the distance to the closest type 1 node is $z + 1 + x$ units, and the distance to the sink is at least $2y \geq 2z + 2$. The only possible solution is to have at least one relay node on the area of distance z from ν_i. There are n such areas, one for each sensor of type 4. The areas are non-overlapping, and the number of relays is also n; thus, there must be exactly one relay node in each area.

Let us denote by X the indexes of the areas where the relay node is closer to η than η'. Denote $\{1, \dots, n\} \setminus X$ by X'. As the answer to the PARTITION problem was *no*, $\sum_{i \in X} a_i \neq \sum_{i \in X'} a_i$. Without loss of generality, we may assume that $\sum_{i \in X} a_i < \sum_{i \in X'} a_i$. Clearly $\sum_{i \in X} a_i < b$. As b and a_i are integral, $\sum_{i \in X} a_i \leq b - 1$.

The sensor η has to send b units of data to other nodes. The node has enough energy resources for transmitting b units of data to the distance of exactly x units. If some part of the data was sent over a larger distance, another part would have to be sent to a node whose distance is less than x units; however, no sensor or sink node is available closer than this, and all relays are already tied to the proximity of type 4 nodes. Thus, the only possibility is to send all data to type 2 nodes, each exactly x units from the source node. Let the amount of data transmitted from η to κ_i be c_i. Clearly $\sum_{i \in X \cup X'} c_i = b$ and each $c_i \geq 0$.

Now, $\sum_{i \in X} a_i \leq b - 1 = \sum_{i \in X \cup X'} (c_i - 1/n)$. At least one of the following holds: there is an i such that $i \in X$ and $a_i \leq c_i - 1/n$, or there is an i such that $i \in X'$ and $c_i \geq 1/n$. If neither holds, then $\sum_{i \in X} a_i > \sum_{i \in X} (c_i - 1/n) \geq \sum_{i \in X \cup X'} (c_i - 1/n)$, a contradiction.

Let us first assume that there is an i such that $i \in X$ and $a_i \leq c_i - 1/n$. In this case the node κ_i would have to transmit at least $a_i + 1/n$ units of data to some other node. The distance to the closest node is at least 1 unit. Thus, the transmission cost is at least $a_i + 1/n$, exceeding the available battery capacity a_i, a contradiction.

On the other hand, if there is an i such that $i \in X'$ and $c_i \geq 1/n$, the node κ_i would have to transmit at least $1/n$ units of data to some other node. As $i \in X'$, the distance to the closest relay node is at least $z + 1$ units. The only node less than $z + 1$ units from κ_i is ν_i, and it does not have any battery capacity for forwarding data. Thus, we need to transmit at least $1/n$ units of data to a distance of at least $z+1$ units, requiring at least $(1/n)\,(z+1)^{\alpha}$ units of energy. Here:

$$\begin{aligned}(1/n)(z+1)^{\alpha} &\geq (1/n)((na^{*})^{\frac{1}{\alpha}}+1)^{\alpha}\\ &> (1/n)((na^{*})^{\frac{1}{\alpha}})^{\alpha}\\ &= (1/n)(na^{*})\\ &= a^{*} \geq a_i = E(\kappa_i).\end{aligned} \tag{47}$$

Again, a contradiction. Thus the assumption must be false. □

Theorem 4. *The relay placement problem classes $\mathcal{P}_{UFI}$ and $\mathcal{P}_{PFI}$ are NP-hard.*

Proof. Follows directly from Lemmas 1, 2, and 3. □

5.2 Reducing Set Covering to k-optimal Relay Placement

In this section, we will prove that it is NP-hard to solve the relay-constrained optimisation problems of classes $\mathcal{P}_{DSI}$ (finite problems with a line-of-sight model and identical batteries) and $\mathcal{P}_{PSI}$ (planar problems with a line-of-sight model and identical batteries) k-optimally. We will show this by constructing a polynomial reduction from Set Covering, which is another well-known NP-complete problem [Kar72, GJ03]. We will see that even an *approximate* solution to these relay placement problems gives us an *exact* solution to the Set Covering problem.

In the Set Covering problem, one is given a collection of sets. One is asked to find m sets which cover all points. Formally:

Definition 14 (Set Covering). An instance of the problem Set Covering consists of a finite collection of finite sets, $\mathcal{A} = \{A_1, \ldots, A_n\}$, and a positive integer m. A subcollection $\mathcal{X} \subseteq \mathcal{A}$ is a feasible solution if $|\mathcal{X}| \leq m$ and $\bigcup \mathcal{X} = \bigcup \mathcal{A}$.

We will now develop a polynomial reduction from Set Covering to $\mathcal{P}_{DSI}$ and $\mathcal{P}_{PSI}$. As was the case in the previous section, the same construction applies to both relay placement problem classes. Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and m be given. Let a denote the total number of distinct elements in the sets, $a = |\bigcup \mathcal{A}|$. Without loss of generality we will assume that the elements are consecutive positive integers, $\bigcup \mathcal{A} = \{1, \ldots, a\}$.

Construct a relay placement problem instance P as follows. Choose $\lambda = 1$ so that we are interested in optimising the minimum amount of data gathered from each node. Choose the following radio propagation model:

$$p = 2, \tag{48}$$
$$\alpha = 2, \tag{49}$$
$$\rho = 0. \tag{50}$$

Then, define:

$$x = 4m, \tag{51}$$
$$y = 2x(a+n). \tag{52}$$

Construct the problem geometry as shown in Figure 8. This figure illustrates the case of $m = 2$, $n = 3$, and $a = 5$. Furthermore, in this example we assume that

$$A_1 = \{1, 2, 3\}, \tag{53}$$
$$A_2 = \{2, 4, 5\}, \tag{54}$$
$$A_3 = \{1, 2, 4\}. \tag{55}$$

In the construction, all obstacles consist of polygons with only 45° and 90° angles. All coordinates are integral. Let us first focus on part (a) of the figure. On the left-hand side, we have $a + 2n - 1$ triangular *nests*. The first $n - 1$ nests are empty. Then, there are a nests, Λ_1 to Λ_a, each corresponding to one element of $\bigcup \mathcal{A}$. The nest Λ_i contains the sensor node η_i with the following characteristics:

$$E(\eta_i) = 1, \tag{56}$$
$$s(\eta_i) = 1. \tag{57}$$

The next nest, Λ_σ, contains the sink node σ. The battery capacity of the sink is irrelevant as the reception cost is zero. The last $n - 1$ nests are empty. The spacing between the nests is y units.

On the right-hand side, we have n triangular *slots*, Υ_1 to Υ_n, again with a spacing of y units. Each slot corresponds to one element of $\mathcal{A}$. Let us now have a closer look at one of these slots, let it be slot Υ_j. See Figure 8 (b) for an illustration. On the leftmost side of the slot, we have $a + n - 1$ diamond-shaped obstacles, each of them x units wide and x units high.

Between the diamond-shaped obstacles, we have $a + n$ *holes*. The first $n - j$ holes are unused. The next a holes, Ξ_{1j} to Ξ_{aj}, correspond to the sensors η_1 to η_a, and the next hole, $\Xi_{\sigma j}$ corresponds to the sink σ. Finally, there are $j - 1$ unused holes.

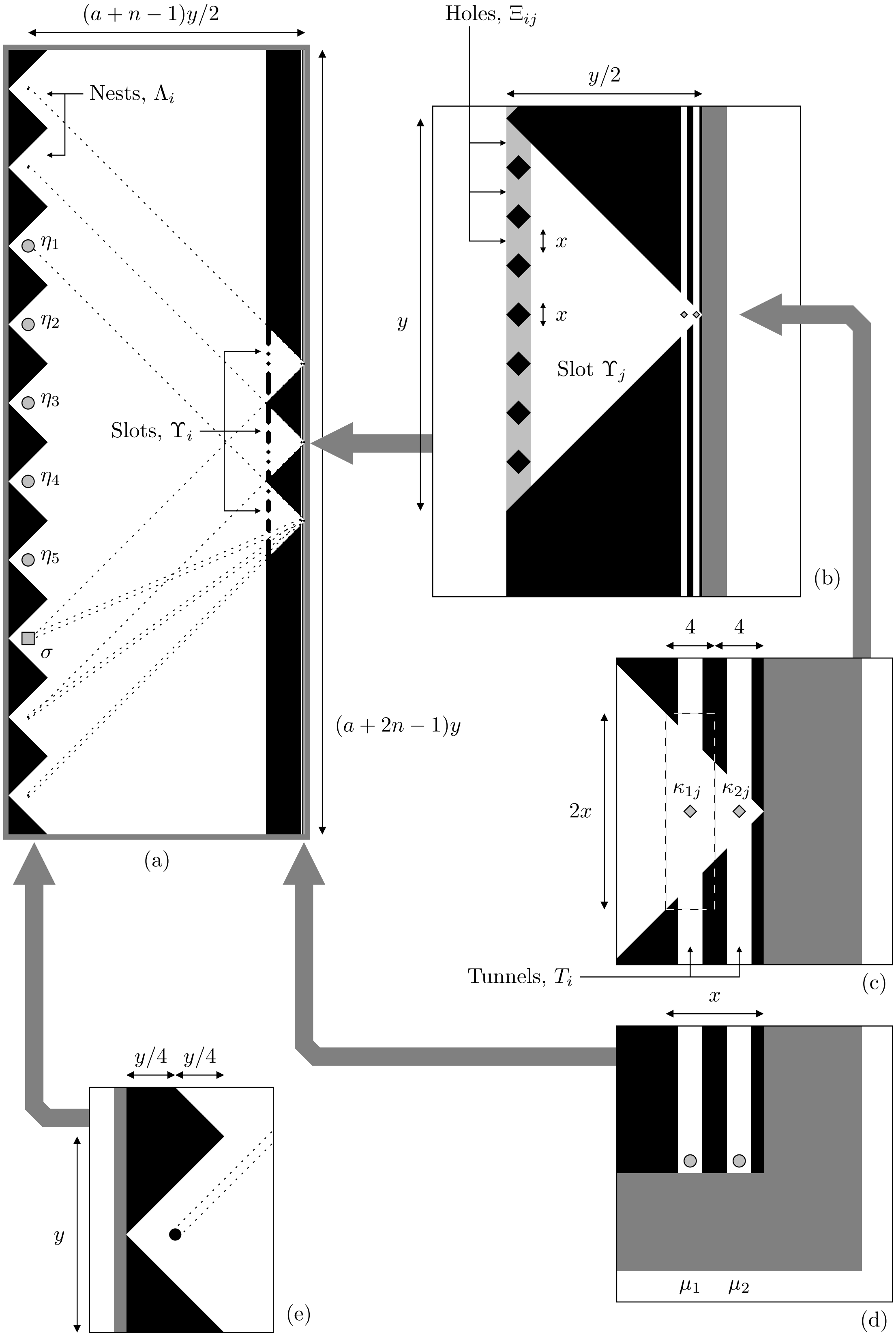


Figure 8: Reduction from Set Covering to $\mathcal{P}_{DSI}$ and $\mathcal{P}_{PSI}$. Some details are shown in a larger scale.

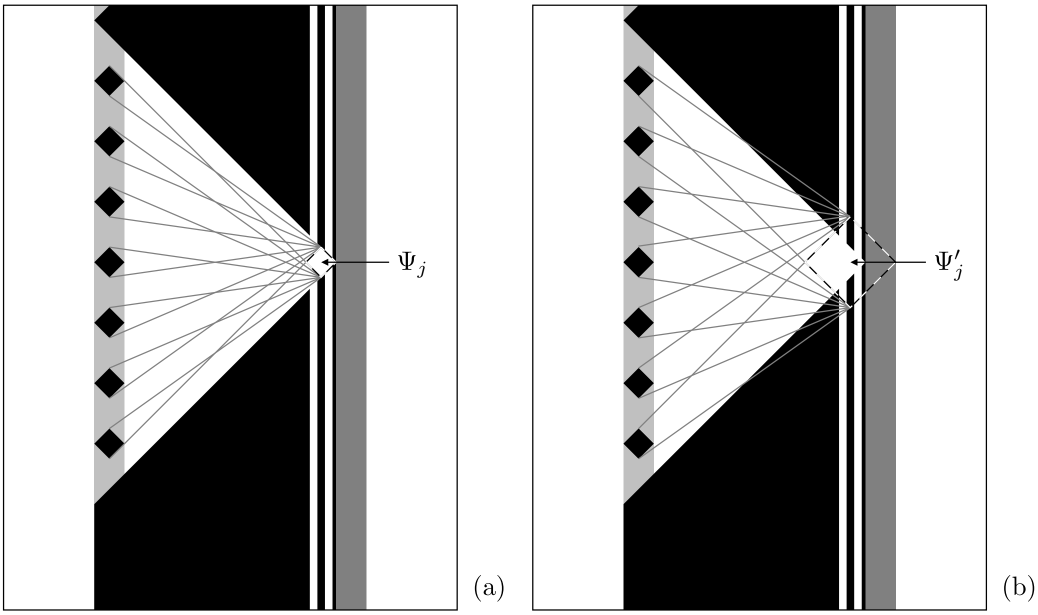


Figure 9: An illustration of areas Ψ_j and Ψ'_j.

Let us now construct two areas, Ψ_j and Ψ'_j. The first area contains points l which satisfy the following conditions:

1. For all i, there is a line of sight from l to η_i through Ξ_{ij}.
2. There is a line of sight from l to σ through $\Xi_{\sigma j}$.

This condition is illustrated in Figure 9 (a). We do not need to calculate the exact shape of this area, it is enough to note that these conditions hold for all points within an x-unit-wide and x-unit-high diamond-shaped area at the rightmost corner of the slot. We will refer to this area as Ψ_j. The second area contains points l which satisfy the following conditions:

1. For all i, if there is a line of sight from l to η_i, it necessarily passes through Ξ_{ij}.
2. If there is a line of sight from l to σ, it necessarily passes through $\Xi_{\sigma j}$.

This condition is illustrated in Figure 9 (b). Again, it is enough for the purposes of this construction to note that these conditions are clearly satisfied for all points on a diamond-shaped area which is $3x$ units high and wide. We will refer to this area as Ψ'_j.

Now, we will block some of the holes. The hole Ξ_{ij} is blocked if and only if $i \notin A_j$. The holes $\Xi_{\sigma j}$ are never blocked. Unused holes can be blocked; they do not matter.

We will also need m narrow, vertical tunnels, T_1 to T_m, in the rightmost part of the construction; see Figure 8 (c) and (d) for an illustration. Each tunnel consists of a 1-unit-wide wall, a 2-unit-wide tunnel, and a 1-unit-wide wall, and we will refer to the *interior* of this 4-unit-wide area as T'_i. We have a total of $4m = x$ units of tunnels and walls. At the bottom of each tunnel, one sensor node is placed. These sensor nodes, μ_1 to μ_m, have the following characteristics:

$$E(\mu_i) = 1, \tag{58}$$

$$s(\mu_i) = 1. \tag{59}$$

At locations where tunnels and triangular slots intersect, there are possible relay locations. The relay location at the intersection of the tunnel T_i and the slot Υ_j is denoted by κ_{ij}. Note that this relay location is inside the area Ψ_j.

Finally, the construction is surrounded by four walls, shown in the figure in grey colour. All relays have a battery capacity of 1 unit. Now we are ready to state a few lemmas.

Lemma 5. *Let us denote by X_l the set of indexes j such that η_j is visible from the location l. If $l \in \Psi_j$, then $X_l = A_j$. If $l \in \Psi'_j$, then $X_l \subseteq A_j$.*

Proof. Follows directly from the construction. □

Lemma 6. *All points in T_i are visible from the sensor node μ_i. No point outside T'_i is visible from the sensor node μ_i.*

Proof. The first part follows directly from the construction. As for the second part, we need to inspect more closely the geometry of the slots. Consider the rectangular area which is illustrated with dashed lines in Figure 8 (c). The area is 4 units wide and $2x$ units high. The vertical distance from the sensor node to the corresponding tunnel entrance at the bottom of the rectangle is more than $y/2 \geq 2x(a+n)/2 \geq 2x$ units. The horizontal distance from the sensor node to the corner of the tunnel entrance is 1 unit, and the horizontal distance to the left edge of the rectangle is 2 units. Thus, any line segment drawn from the sensor node to the left edge of this rectangle necessarily passes through an obstacle. The same applies to the right edge of the rectangle. There is no line-of-sight from the sensor node to any point outside T'_i. □

Lemma 7. *Constructing this relay placement problem instance is possible in polynomial time.*

Proof. The construction involves generating a problem instance with $O(a+n)$ sensors, $O(nm)$ possible relays, and $O((a+n)n)$ quadrilateral or triangular obstacles. Calculating the parameters of each node and each obstacle can be performed in a polynomial time. The calculations only involve integers. □

We can now formulate the following relay-constrained optimisation problem instance (see Definition 4 on page 24): P is the relay placement problem instance constructed above, and the number of relays N is m.

Lemma 8. *If the answer to the* Set Covering *problem instance is* yes*, the optimal solution to the relay placement problem instance constructed above has a positive utility, both in the $\mathcal{P}_{DSI}$ and in the $\mathcal{P}_{PSI}$ formulation.*

Proof. Let $\mathcal{X} = \{A_{c_1}, \ldots, A_{c'_m}\}$ be a solution to the Set Covering problem, with $m' \leq m$. Place relays 1 to m' as follows: the relay i is placed on the location of κ_{ic_i}. Next, place relays $m'+1$ to m as follows: the relay i is placed on the location of κ_{ij} where j is arbitrary (choose, for example, $j = 1$).

Now, for any sensor η_j, it holds that $j \in \bigcup \mathcal{X}$. Thus, there is an i such that $j \in A_{c_i}$. It follows that there is a relay at κ_{ic_i}. This location is on the area Ψ_{c_i}, and we have a line of sight to both the sensor node η_j and the sink node σ. We can transmit a positive amount of data from the sensor to the sink.

Similarly, for any sensor μ_i, there is a j such that there is a relay at κ_{ij}. This location is on the intersection of the tunnel T_i and the area Ψ_j, and we have a line of sight to both the sensor node μ_i and the sink node σ. Again, we can transmit a positive amount of data from the sensor to the sink.

By sharing battery capacities of the relays equally among all these flows, we can transmit a positive amount of data from all sensor nodes to the sink. Thus, there is a solution with a positive utility, and the utility of the optimal solution is also positive. □

Lemma 9. *If the answer to the* Set Covering *problem instance is* no*, there is no solution with a positive utility, either in the $\mathcal{P}_{DSI}$ or in the $\mathcal{P}_{PSI}$ formulation.*

Proof. Let us first assume that there is a solution with a positive utility. As the utility is positive, we are able to transmit a positive amount of data from all sensor nodes to the sink.

Each node μ_i is able to send some data to some other node. By Lemma 6, the target node needs to be located on T'_i. As there are no other nodes in this area, we need to have at least one relay in this area. As we have m non-overlapping areas and m relays, we need to have exactly one relay on T'_i for each i. Let us call this relay ν_i.

Denote by Y_i the set of indexes j such that η_j is visible from the relay ν_i. Let $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$. As each node η_j needs to transmit a positive amount of data to the sink, and there is no line of sight to any other node except possibly relays, we must have $\bigcup \mathcal{Y} = \bigcup \mathcal{A}$. Now there are two possibilities for each i:

1. There is a j such that the relay ν_i is located on Ψ'_j. By Lemma 5, $Y_i \subseteq A_j$.
2. Otherwise the relay is in the tunnel in a place where there is no line of sight to any nest. Now $Y_i = \emptyset \subseteq A_1$.

For each i there is now c_i such that $Y_i \subseteq A_{c_i}$. Define $\mathcal{Y}' = \{A_{c_1}, \ldots, A_{c_m}\}$. Now we have $\bigcup \mathcal{A} = \bigcup \mathcal{Y} \subseteq \bigcup \mathcal{Y}' \subseteq \bigcup \mathcal{A}$. Thus, $\mathcal{Y}'$ is a feasible solution to the Set Covering problem instance, a contradiction. □

Theorem 10. *Solving the relay-constrained optimisation problems of classes $\mathcal{P}_{DSI}$ and $\mathcal{P}_{PSI}$ k-optimally is NP-hard.*

Proof. Let us assume that for some k, we have an oracle for solving the relay-constrained optimisation problems of class $\mathcal{P}_{DSI}$ or $\mathcal{P}_{PSI}$ k-optimally in constant time. We may then use the construction presented above to solve Set Covering in polynomial time.

By Lemma 7, we may construct the relay placement problem instance in polynomial time. By Lemmas 8 and 9, the oracle will return a solution with a positive utility if and only if the answer to the Set Covering problem is *yes.* □

We also briefly note the following result:

Theorem 11. *Solving the utility-constrained optimisation problem of class $\mathcal{P}_{DSI}$ k-optimally is at least as hard as solving the optimisation version of* Set Covering *k-optimally.*

Proof. We may use a small positive value as the utility requirement. If we find a solution to the utility-constrained optimisation problem of class $\mathcal{P}_{DSI}$ with at most km relays, we have also found a set cover of at most km subsets. □

5.3 Bounding Computational Complexity

We will also prove the following upper bound for the computational complexity of some relay placement problems:

Theorem 12. *The decision version of the relay placement problem class $\mathcal{P}_D$ is in NP.*

Proof. Let (P, N, u) be an instance of the decision problem, $P \in \mathcal{P}_D$, parametrised like a generic $\mathcal{P}_D$ instance. By Definitions 1, 2, and 3, if and only if the answer to the decision problem is *yes*, there exists a set of relays $R \in \mathcal{R}$ and a flow f such that $f \in \mathcal{F}(B)$, $F(B, f) \geq u$, and $|R| = N$ where $B = B(P, R)$. Thus, (R, f) can be used as a *certificate* for a *yes* instance of the decision problem.

In order to prove that $\mathcal{P}_D$ is in NP, we need to show that the certificate can be checked in polynomial time. Let us first make sure that the size of the certificate is polynomial in the size of the input. The flow can be represented as a matrix with one element for each pair of nodes. On the other hand, the transmission cost matrix, which is a part of the problem description, contains one element for each pair of possible nodes. Thus, the size of the flow description is $O(|P|)$ values. The size of the cost matrix is $|\mathcal{V}|^2 > |R|^2 \geq |R|$. Thus, the set of relays, R, can also be described in $O(|P|)$ values.

Now we know that the *number* of values in the certificate is polynomial in the number of values of the input. We also need to bound the *size* of each value. It is trivial to represent R compactly, so we will focus on the flow f. The flow is a solution of an LP, and the size of the LP is polynomial in the input. All coefficients of the LP are rational, and we can multiply them by the product of all denominators. The coefficients of the new LP are integral and their total size is only polynomially larger than the size of the original LP. Finally, the size of the solution is only polynomially larger than the size of the new LP [PS98, Sections 2.2 and 8.7]. Thus, not only the number of values but also the size of each value in the certificate is polynomial in the size of the input.

Constructing $B(P, R)$ is polynomial for finite problems: one simply filters out possible relays which are not members of R, along with corresponding energy vector elements and transmission cost matrix rows and columns. Next, one can check equations (3)–(8) on page 21 and see if each of them holds. As there is only a constant number of arithmetic operations for each flow element, the number of arithmetic operations is $O(|P|)$. Finally, we may use equations (9) and (10) to see if the utility is at least u, again in $O(|P|)$ operations. Thus, checking the certificate is possible in polynomial time. □

The same proof clearly applies to simplified problem classes $\mathcal{P}_x \subset \mathcal{P}_D$ if the problem instance is described by giving a communication cost matrix. However, there are also problem classes where a much more compact parametrisation is possible. Further research is needed to determine if, for example, $\mathcal{P}_{DF}$ is in NP. This is not trivially true: a straightforward conversion to a cost matrix form may lead into both exponentially long integers and irrational numbers.

5.4 Summary of Computational Complexity

We will first make the following observation:

Lemma 13. *For any relay problem class $\mathcal{P}_x$ in $\mathcal{P}^*$, either $\mathcal{P}_{UFI} \subseteq \mathcal{P}_x$ or $\mathcal{P}_{PFI} \subseteq \mathcal{P}_x$.*

Proof. Follows directly from the hierarchy of problems, see Table 2 on page 20. □

We can now summarise the main results on computational complexity:

Theorem 14. *The decision versions of all relay placement problem classes in $\mathcal{P}^*$ are NP-hard.*

Proof. From Theorem 4 and Lemma 13. □

Theorem 15. *The decision version of the relay placement problem class $\mathcal{P}_D$ is NP-complete.*

Proof. From Theorems 12 and 14. □

Theorem 16. *Finding k-optimal solutions to the relay-constrained optimisation versions of problem classes $\mathcal{P}_x$ satisfying $\mathcal{P}_{DSI} \subseteq \mathcal{P}_x$ or $\mathcal{P}_{PSI} \subseteq \mathcal{P}_x$ is NP-hard.*

Proof. From Theorem 10. □

Note that Theorem 16 applies to all problem formulations with obstacles, except the sensor upgrade problems.

6 Algorithms

In this section, I will present algorithms for solving many classes of relay placement problems. As was proved in Section 5, relay placement problems are typically NP-hard. We will not see here algorithms which solve all instances in polynomial time.

As some simple heuristic solutions with no guarantee of optimality have already been studied by Falck *et al.* [FFK$^+$04], we will focus on algorithms which offer some guarantee, at the expense of computational complexity.

Most algorithms presented here share the following key property: The algorithms maintain bounds for the quality of the current solution. During computation, we have access to an upper bound of the utility. We may wait until the approximation ratio is low enough, or we may interrupt computation after a certain amount of time and have access to an estimate of the quality of the solution. This way we may find a practical balance between quality and computational resources for solving relay placement.

The main theme of this section is developing an algorithm for solving the planar relay placement problem. We will divide the task into subproblems as sketched in Figure 10. We remind the reader that Falck *et al.* [FFK$^+$04] have already developed an LP formulation for solving the balanced data gathering problem, and also an approximation algorithm is known [FKKO05a]. We will begin this section by showing various approaches to solving the finite relay placement problem, and we will then show how these algorithms can be used as a component when one is solving the planar problem. We focus primarily on approximation algorithms for the relay-constrained problem; see Sections 6.1.4 and 6.2.3 for some extensions.

6.1 Solving the Finite Relay Placement Problem

In this section, we will present three different approaches to solving the relay-constrained finite relay placement problem. We will see from the empirical results presented in Section 7 that each of these three methods is better than the others in certain problem instances. The methods presented here may be used for solving an arbitrary finite relay placement problem. The transmission cost model is not

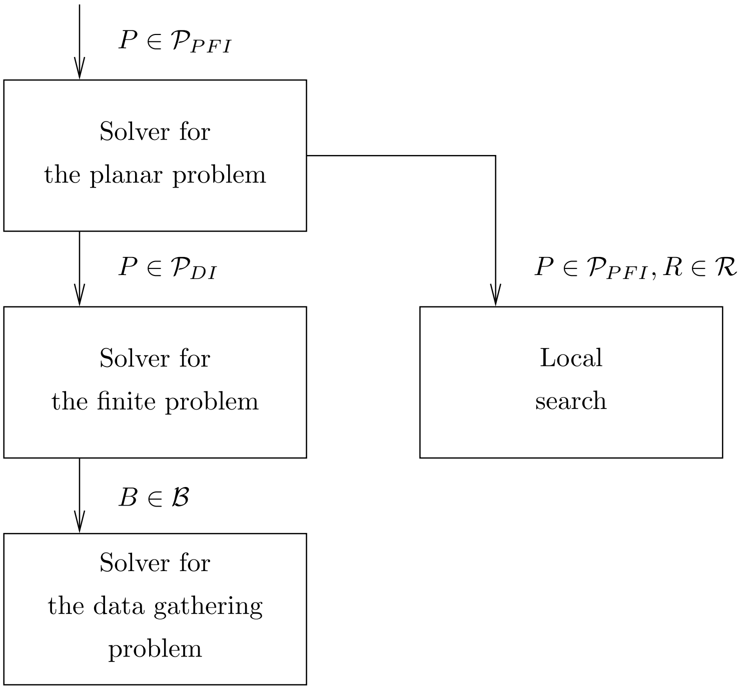

$P \in \mathcal{P}_{PFI}$
Solver for the planar problem
$P \in \mathcal{P}_{DI}$
$P \in \mathcal{P}_{PFI}, R \in \mathcal{R}$
Solver for the finite problem
Local search
$B \in \mathcal{B}$
Solver for the data gathering problem

Figure 10: Overview of algorithms. Each square represents a solver for a subproblem, and the arrows show the most relevant data flows.

restricted; an arbitrary cost matrix may be specified.

When using the finite solver as a component of the planar solver, we will encounter problems where there are multiple possible relays at the same location. In order to handle these cases efficiently, we will extend the finite relay placement problem by introducing a new input parameter, vector **a**. The value a_η specifies the number of relays one may assign at the location of the possible relay η. As the solution would be a multiset, it is presented as a vector **x** where $x_\eta \leq a_\eta$ is the number of relays assigned at this location.

6.1.1 Mixed Integer Linear Programming Formulation

One possibility is to formulate the finite relay placement problem as a *mixed integer linear program* (MIP or MILP). This way, one can use any existing MIP solver to solve the problem. A mixed-integer linear program is an LP where some variables are constrained to integral values. See, for example, Papadimitriou and Steiglitz [PS98] for an overview on integer linear programming. Unlike pure linear programs, solving an integer linear program is NP-complete [Kar72]. Algorithms for solving MIP problems are typically based on either so-called cutting-plane techniques, or intelligent enumeration of all possible combinations.

We will first make one assumption: there is no feasible flow f and relay η such that the relay consumes no energy at all, $\sum_{\kappa \in V} (\tau_{\eta\kappa} f_{\eta\kappa} + \rho f_{\kappa\eta}) = 0$, but that it is forwarding some data, $\sum_{\kappa \in V} (f_{\eta\kappa} + f_{\kappa\eta}) > 0$. This assumption is relatively reasonable: it is satisfied if the reception cost is non-zero, or if all transmission costs are non-zero. However, one must also maintain this condition when using the finite solver as a component in the planar solver.

Given this assumption, the MIP formulation is a straightforward extension of equations (3) to (8):

$$\text{Maximise } \lambda \check{q} + (1-\lambda)/|S| \sum_{\kappa \in V} f_{\kappa\sigma}, \text{ subject to:}$$

$$f_{\eta\kappa} \geq 0, \qquad \forall \eta, \kappa \in \mathcal{V}, \tag{60}$$

$$f_{\eta\eta} = 0, \qquad \forall \eta \in \mathcal{V}, \tag{61}$$

$$\check{q} \geq 0, \tag{62}$$

$$x_\eta \in [0, a_\eta], \qquad \forall \eta \in \mathcal{R}, \tag{63}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} f_{\sigma\kappa} = 0, \tag{64}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} (f_{\eta\kappa} - f_{\kappa\eta}) \in [0, s_\eta], \qquad \forall \eta \in S, \tag{65}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} (f_{\eta\kappa} - f_{\kappa\eta}) \geq \check{q}, \qquad \forall \eta \in S, \tag{66}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} (f_{\eta\kappa} - f_{\kappa\eta}) = 0, \qquad \forall \eta \in \mathcal{R}, \tag{67}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} (\tau_{\eta\kappa} f_{\eta\kappa} + \rho f_{\kappa\eta}) \leq E_\eta, \qquad \forall \eta \in S^+, \tag{68}$$

$$\sum\nolimits_{\kappa \in \mathcal{V}} (\tau_{\eta\kappa} f_{\eta\kappa} + \rho f_{\kappa\eta}) \leq E_\eta x_\eta, \qquad \forall \eta \in \mathcal{R}, \tag{69}$$

$$\sum\nolimits_{\eta \in \mathcal{R}} x_\eta \leq N, \tag{70}$$

$$f_{\eta\kappa} \in \mathbb{R}, \qquad \forall \eta, \kappa \in \mathcal{V}, \tag{71}$$

$$\check{q} \in \mathbb{R}, \tag{72}$$

$$x_\eta \in \mathbb{Z}, \qquad \forall \eta \in \mathcal{R}. \tag{73}$$

Here $\check{q}$ is an auxiliary variable which is used for calculating the minimum of the amounts of the data gathered from each sensor node. The integral variables x_η will contain the solution to the relay placement problem. The key point of this formulation is equation (69), where we use the above assumption: if $x_\eta = 0$, the energy constraint is zero, and by the assumption, there is no data flow either.

6.1.2 Heuristic Search with Local Search

Now we will develop an algorithm which solves the finite relay placement problem without relying on an MIP solver. The algorithm consists of two mostly independent parts: the first part keeps tightening the upper bound for the solution, while the second part keeps looking for a good solution.

The first part uses *heuristic search*; see Russell and Norvig [RN03] for an introduction to the topic. A search tree is formed as follows. Each tree node is presented as a tuple $(\mathbf{a}, \mathbf{b}, N)$. The value b_η specifies the number of relays we have already assigned at the location of the possible relay η. At the root of the tree, $\mathbf{a}$ and N are as given in the original problem, and $\mathbf{b} = \mathbf{0}$. Let κ be any possible relay with $a_\kappa > 0$. If no such possible relay exists, we have reached a leaf node. Otherwise, there are two possible branches at each node:

1. Form a new node $(\mathbf{a}', \mathbf{b}', N')$ by assigning $a'_\kappa = a'_\kappa - 1$, $b'_\kappa = b_\kappa + 1$, and $N' = N - 1$. In other words, add one more relay at the location κ.
2. Form a new node $(\mathbf{a}', \mathbf{b}, N)$ by assigning $a'_\kappa = 0$. In other words, fix the number of relays at the location κ to the value it has now. This branch may be pruned if $\sum a'_\eta < N$.

Now we need a heuristic function to guide our search in this tree. We will use an admissible heuristic in order to guarantee that we will find an optimal solution. To develop the heuristic function, we will use the well-known method of using a relaxed version of the original problem as the heuristic [RN03]. Here we may consider the above MIP formulation as the original problem, and we will relax it by removing the integrality constraint for $\mathbf{x}$. Furthermore, we need to take into account the current value of $\mathbf{b}$. The changes to the MIP formulation are thus:

$$\sum\nolimits_{\kappa \in \mathcal{V}} (\tau_{\eta\kappa} f_{\eta\kappa} + \rho f_{\kappa\eta}) \leq E_\eta x_\eta + E_\eta b_\eta, \qquad \forall \eta \in \mathcal{R}, \qquad (69')$$

$$x_\eta \in \mathbb{R}, \qquad \forall \eta \in \mathcal{R}. \qquad (73')$$

We are left with an LP, which may be solved by any LP solver.

This heuristic function is evaluated for each created node. The utility of the node is an upper bound for the utility of the entire branch. The nodes are stored in a priority queue, using the bounds as keys. At each iteration, the node with the highest bound is removed from the priority queue. Its child nodes are created, evaluated, and stored back to the queue. When the first leaf node is removed from the queue, we will know that there cannot be other leaf nodes with a higher utility. Thus, we may terminate our heuristic search and return the configuration which corresponds to the leaf node.

Now we have finished the first part of our algorithm. While this method was derived from the viewpoint of informed search as studied by the artificial intelligence community, it may also be seen as a way of implementing a MIP solver for this particular problem by using branch-and-bound techniques.

The second part tries to find a good solution by *local search*. This part could use an arbitrary local search method. Here we have chosen a method based on running a number of parallel hill-climbing algorithms, with random restarts after reaching a local maximum. Other possible approaches include methods like simulated annealing and genetic algorithms; experimenting with these is left for future research.

Our local search will proceed as follows. We will have a number of searchers. The internal state of each searcher is maintained as a relay assignment vector $\mathbf{x}$ which satisfies the constraints (63) and (70) above, along with the corresponding utility. Each searcher is initialised to a random configuration. While local search is running, each searcher executes one step, and passes the turn to the next one. At each step, the searcher attempts to modify the relay assignment by moving one relay to a new possible location such that the above constraints are still satisfied. If such a move is found, the utility of the new configuration is evaluated by solving the corresponding LP. If the utility improves, the new configuration is kept; otherwise, the next possible one-relay movement is attempted at the next step. When a local optimum is reached (i.e., no one-relay movement improves the solution), a new random configuration is generated. All searchers maintain a shared variable which stores the best configuration so far.

We will now combine these two parts into one search algorithm. We will run both search processes in an interleaved fashion, giving roughly equal amounts of computing resources to both parts. After each step, we will check if a termination criterion is met. We have two termination criteria:

1. The heuristic search terminates. It has found an optimal solution.
2. The current upper bound maintained by the heuristic search is at most k times the utility of the best configuration found by the local search so far. The local search has found a k-optimal solution.

We see that this process always terminates, as the heuristic search will eventually find an optimal solution; there are, after all, a finite number of possible configurations to check. However, the local search may help by letting us terminate much sooner, in case we are satisfied with an approximate solution.

6.1.3 Exhaustive Search

As a third alternative, we will consider uninformed exhaustive search. Here we will simply enumerate all possible relay assignment vectors $\mathbf{x}$ and choose the best one. We will eventually find an optimal solution. Again, one may interpret this approach as yet another method for solving the original MIP problem.

The reason why this approach is also considered will become evident in Section 7. We will see that, in the case of only a few sensors and a large number of possible relay nodes, evaluating the heuristic function may require solving a relatively large LP problem, while exhaustive search only needs to solve very simple LP problems at each node. If the number of relays to place is low enough, exhaustive search may enumerate all possible combinations much faster than what it takes for the heuristic search to converge.

In empirical experiments, exhaustive search and heuristic search will be combined into one interleaved process. However, other solutions are also possible: one could, for example, choose only one of these search methods based on the problem size.

6.1.4 Generalisation

The MIP formulation can be easily generalised to the problem of solving the utility-constrained problem. One just needs to interchange the roles of the objective function and the relay number constraint (70).

6.2 Solving the Planar Relay Placement Problem

Now we will turn our attention to solving planar relay placement problems. We will first assume that we are using the free-space transmission cost model and identical batteries. Some generalisations will be discussed in the end of this section.

In the free-space model, it is easy to see that we may confine ourselves to the problem of placing relays in a bounding rectangle which contains all sensors and the sink:

For any solution with relays outside this rectangle, we can find another solution with all relays inside the rectangle without making any of the transmission costs higher. Thus, the utility of the latter solution is no lower than the utility of the former solution.

6.2.1 Basic Algorithm

The algorithm will need two components. Firstly, it needs an approximate solver for the finite relay placement problem. This component may be seen as a black box. The only requirement is that the component returns not only a solution and its utility, but also an upper bound for the utility. Secondly, it needs a solver for the balanced data gathering problem; here we may simply use the LP formulation and any LP solver as this part is not performance-critical.

The basic approach is as follows. We will maintain an upper bound and a solution. We will keep making the upper bound tighter and the utility of the solution better until the ratio of these two values is at most k. Then we may return the solution, and it is guaranteed that the solution is k-optimal.

The upper bound is derived by dividing the bounding rectangle into a number of rectangular cells. The cells form a partition of the bounding rectangle, i.e., they are non-overlapping and they cover all points of the rectangle. Next, we will construct an instance of the finite relay placement problem. The sink and the sensors are the same as in the original planar problem. For each cell, we will add one possible relay node η with $a_\eta = N$. The battery capacities of the nodes, the amount of available data on each sensor, the balance parameter, and the reception costs are exactly as specified in the original problem.

Now we will specify the transmission costs between the nodes of the finite problem constructed above. We will assign a geometrical area to each node. For each sensor node and the sink, this area is the single point of the location of the node. For each relay, the area is the corresponding rectangular cell. The transmission cost between two nodes is specified as the *lowest possible transmission cost between their respective areas.* For the free-space model, we may easily calculate the shortest distance between two areas, and evaluate the corresponding lowest transmission cost. This may be zero, if the areas are adjacent; however, the assumption made in Section 6.1.1 is not violated as long as we have a non-zero reception cost.

Next, we will solve this finite relay placement problem by using any approximate solver. The solver will return a relay assignment, $\mathbf{x}$, and an upper bound for the utility. The utility of the assignment is called the *cell-view utility*, and the upper bound is called the *cell-view upper bound*. The key observation here is this: the cell-view upper bound is also an upper bound for the utility of the original planar problem.

The solution $\mathbf{x}$ also gives us a solution to the planar relay placement problem. We may place each relay at the centre point of the corresponding cell, assign the transmission costs using the free-space model, and evaluate the utility of this configuration by solving the corresponding balanced data gathering problem. This utility, U, is called the *point-view utility*.

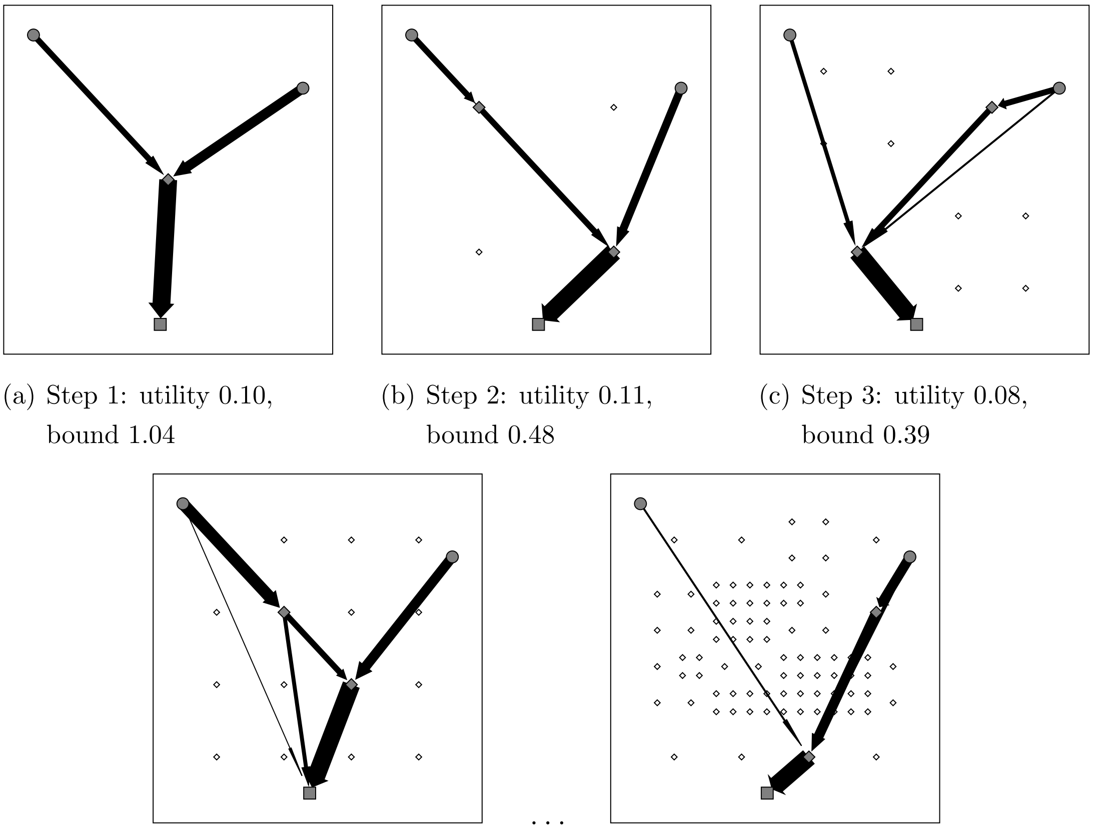
If the tightest cell-view upper bound is at most k times the best point-view utility, we are done. Otherwise, we will divide the bounding rectangle into a larger number of cells, and repeat the process. We still need to specify how we divide the bounding rectangle into cells. To avoid combinatorial explosion, we must make sure we do not make the number of possible relays in the finite problems too high. The following scheme is used here:

1. The first partition consists of one cell, covering the entire bounding rectangle.
2. After each iteration, split each cell which contains some relay nodes into four new rectangles of equal size. (In some rare cases, we may find an approximate solution where no relay nodes were assigned. In those cases, choose one cell randomly and split it.)

This approach gives us a guarantee of convergence while still generating only a moderate number of new cells. Even if the relays are assigned to the same areas, the smaller cells will mean that the point-view utility is closer to the cell-view utility than in the previous iteration. As the finite problem was solved by an approximation algorithm, this will also bound the ratio of the cell-view upper bound and the point view utility. The whole process is illustrated in Figure 11.

6.2.2 Post-processing

The solution returned by the above algorithm is k-optimal. However, it is not necessarily even a local optimum. Thus, it may be possible to improve the utility of the solution by local search. Here one may use, e.g., line search in a similar way as proposed by Falck *et al.* [FFK$^+$04] for their incremental relay placement algorithm.



(a) Step 1: utility 0.10, bound 1.04

(b) Step 2: utility 0.11, bound 0.48

(c) Step 3: utility 0.08, bound 0.39

(d) Step 4: utility 0.14, bound 0.27

(e) Step 15: utility 0.07, bound 0.17

Figure 11: Illustration of the algorithm. Here we are placing two relays in a very simple sensor network which consists of only two sensor nodes; the balance parameter λ is 0.5. The figures show how the plane is divided into smaller cells. Centre points of cells are illustrated by small diamonds. Here the fourth step already produced a 1.25-optimal solution. However, it took 15 steps to prove that the upper bound for the utility is as low as 0.17. The configuration shown in step 4 was the best, and it is returned as the solution.

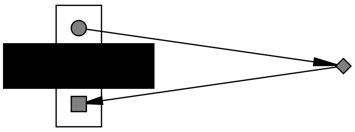


Figure 12: If there are obstacles, the bounding rectangle of the sensors and the sink does not necessarily contain the relay locations of any optimal solution.

Any heuristic method is safe here. As long as we check the utility of the final solution, and make sure we do not make the solution worse, we still have the guarantee of k-optimality. Again, various local search methods offer possibilities for future research.

6.2.3 Generalisation and Extensions

Generalising this algorithm to utility-constrained planar problems is possible by using a solver for the utility-constrained finite problem as a component.

We may also take into account the fact that relay placement may be imprecise. Instead of evaluating the utility of a configuration as presented above, we may replace each relay node by an area. The area could be, for instance, a disk, which represents the possible final locations of the relay node. Then, we may assign *worst case* transmission costs by measuring the *longest* distance between these disks. This way we can analyse how sensitive the configuration is to small variations in relay locations, and we may even include this test in the termination criteria of the algorithm.

In addition to the free-space model, other simple radio propagation models may be applied, too, as long as it is possible to analytically derive a lower bound for the transmission cost between two areas. Taking obstacles into account is more challenging. Figure 12 shows that the bounding box of the sensors and the sink does not necessarily contain the relay locations of any optimal solution. This example also shows that the area which needs to be checked may be made arbitrarily large. Thus, one needs to, for instance, further constraint the set of possible relay locations to some predefined area given by a human expert. This does not need to be a serious restriction as solutions where relays are placed at arbitrarily long distances are physically unrealistic.

Obstacles need to be considered also when evaluating the minimum communication cost between two areas. Here one may derive a lower bound as follows: If there is a line of sight between the areas, evaluate the communication cost as presented above for the free space model. Otherwise, the communication cost is infinite. For checking if there is a line of sight between two areas, one may use algorithms derived for solving similar problems in the field of computer graphics. See, for example, Teller and Séquin [TS91] for a method based on cells, portals, and so-called stabbing lines. Obstacles may also make local search in the post-processing phase more difficult as costs are no longer continuous functions of relay locations.

7 Implementation and Empirical Results

I have implemented the algorithms presented in the previous section and performed a number of experiments with them. The implementation, results, and some observations are presented in this section.

7.1 Implementation Details

The algorithms were implemented in the C programming language [ISO99]. The source code of the implementation is available[1] under a free software license. The implementation may be run in one of two major modes:

1. In the MIP mode, it uses the mixed integer linear programming formulation as described in Section 6.1.1.

2. In the heuristic mode, it uses the heuristic search and local search as described in Section 6.1.2, and exhaustive search as described in Section 6.1.3.

The implementation of the heuristic mode runs three algorithms in an interleaved fashion: heuristic search for finding an upper bound, local search for finding a solution, and exhaustive search. Both local search and exhaustive search use the same variables to maintain the best configuration seen so far; this way, heuristic search may benefit not only from local search but also from exhaustive search.

The search may be terminated for one of the following reasons: the heuristic search finds a solution; the upper bound maintained by the heuristic search becomes low enough compared to the best configuration seen this far; or the exhaustive search terminates.

A key issue here is deciding how to share computing resources between the three interleaved processes. One step of computation may be many orders of magnitudes

[1] The source code is available at `http://www.cs.helsinki.fi/u/josuomel/relays/` .

more expensive for heuristic search than for local and exhaustive search; furthermore, there is no known method of predicting accurately which of the termination criteria will be met first.

Memory usage is not an issue with the problem instances used in these experiments. The solver may spend several hours or even several days finding an optimal solution for a hard problem, while memory usage is typically around ten megabytes. We will thus focus on sharing processor time optimally.

In this implementation, we use three timers, each counting the amount of computer time used by one of the three processes. At each step, the turn is given to a process which has so far used less than one third of the total computer time. This way, each process will be given a roughly equal share of computer time.

The running time of this simple approach is, in a sense, worst-case optimal, assuming that there is no prior knowledge on the best way of sharing computing resources: if and only if we are using the equal assignment, we know that we are wasting computer time by at most a constant factor of 3 compared to the optimal assignment.

This time sharing scheme may also be seen as a way of simulating how the algorithm would behave if these three processes were assigned to separate processors. Even more parallelism could be achieved by dividing the local search algorithm: each searcher could be run on a dedicated processor. Experimenting with parallel implementations is left for future research.

7.2 MIP and LP Solvers

In the MIP mode, the implementation uses the GLPK library [Mak05] to solve MIP problems. The solver in the GLPK library is based on the branch-and-bound method.

The same library is also used in the heuristic mode for solving LP problems. There are two LP solvers in the GLPK library, one based on the simplex method and one which uses an interior-point method. The implementation allows experimenting with both methods.

The LP solver is used in the inner loop of our algorithms. Thus, being able to solve LP problems of moderate size rapidly is critical to the performance of the algorithms. During development and experiments, some observations were made.

Firstly, the interior-point method in the GLPK library spends a significant amount of time even on simple LP problems, while the simplex method solves small problems quickly. Thus, we will use the simplex method in all these experiments.

Secondly, the LP problem changes only slightly between iteration steps, and it may seem tempting to reuse the same problem instance and the solution, modify constraints, and restart the simplex solver. However, the total amount of time wasted on recovering a feasible solution was in some cases large compared to the gained speedup of finding the optimal solution. Even more importantly, this required the use of a suboptimal LP problem where some variables were, for example, constrained to a fixed value. The performance was significantly improved by rewriting the code so that the LP (as well as MIP) problem instances are constructed from scratch for each iteration step, and no trivially true constraint or fixed variable is included.

7.3 Results

The test environment was a PC with a 3.0 GHz Pentium 4 processor. On this platform, solving the problem in Figure 11 on page 55 took 15 seconds in the heuristic mode. Typically, exhaustive search terminated before heuristic search. Solving the same problem took 290 seconds in the MIP mode.

Figure 13 illustrates placing two relays in a sensor network. Solving these three problems took 470 seconds, 2 122 seconds, and 217 seconds, respectively, in the heuristic mode. In all cases, during the first steps, the heuristic solver found solutions to the finite subproblems, while during the last steps, the exhaustive search terminated before the heuristic search. In the MIP mode, the respective times were 495 seconds, 2 753 seconds, and 212 seconds.

The internal progress for case (b) in the heuristic mode is illustrated in Figure 14. This figure shows both the planar solver and the underlying finite solver. Each marker labelled as a “point view utility” corresponds to the termination of the finite solver. Between a pair of such markers, the finite solver is running: we can see how it is tightening the “cell-view upper bound” by heuristic search, and improving the “cell-view utility” by local and exhaustive search.

In this example, we can see that the “cell-view upper bound” may suddenly drop down to the level of the “cell-view utility”. This phenomenon is caused by the exhaustive search: when the exhaustive search terminates, we know that no better solution exists and we can immediately tighten our bounds. Later we will see examples where the heuristic search does converge to an approximate solution before

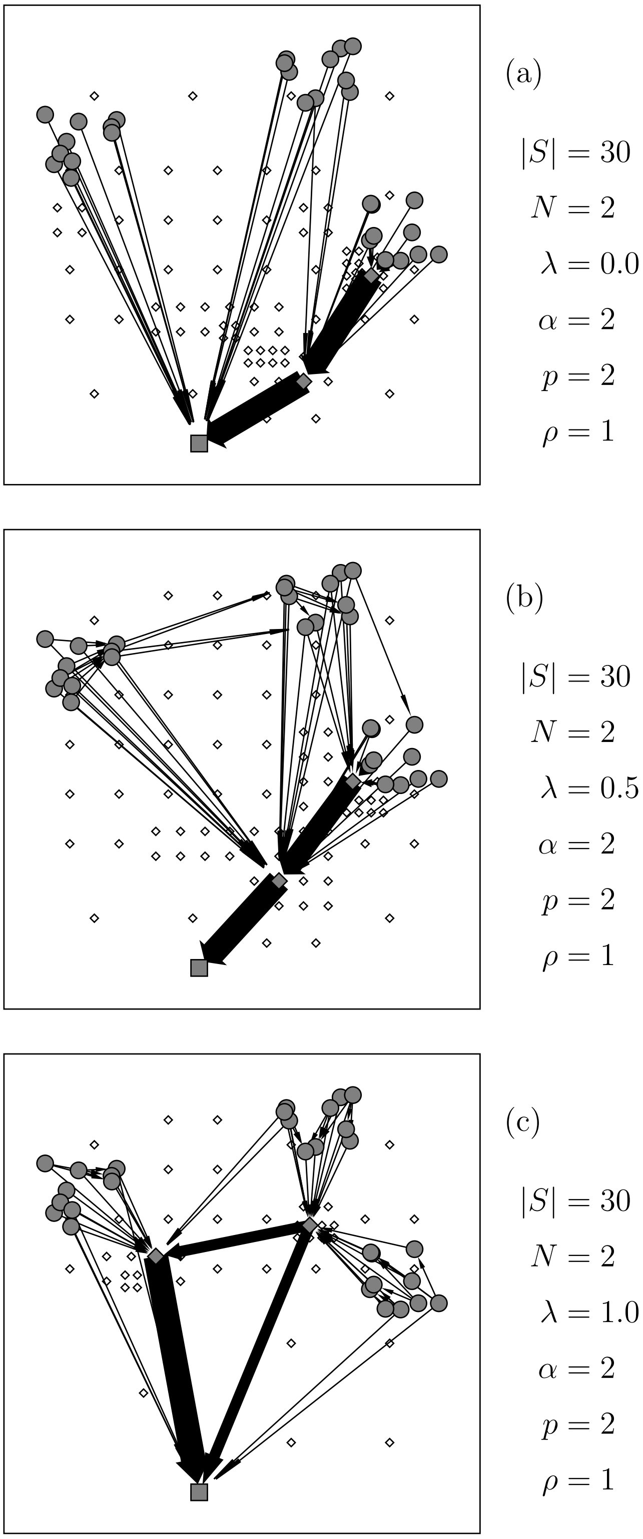


Figure 13: Placing two relays in a sensor network. The network consists of 30 sensors which are placed randomly in three clusters. The solutions are 1.25-optimal or better.

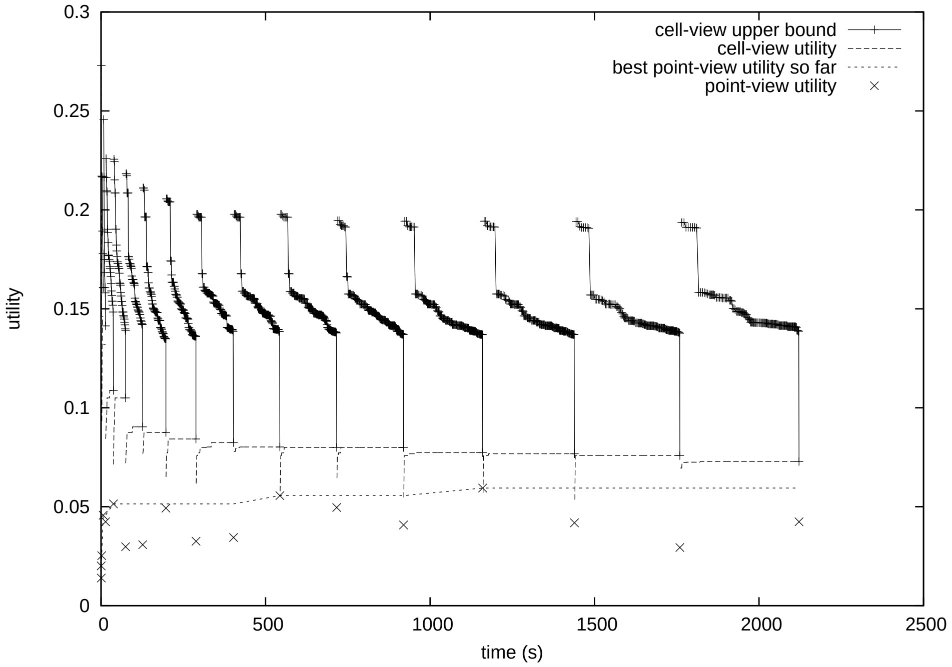


Figure 14: Timings for solving the problem in Figure 13 (b) in the heuristic mode.

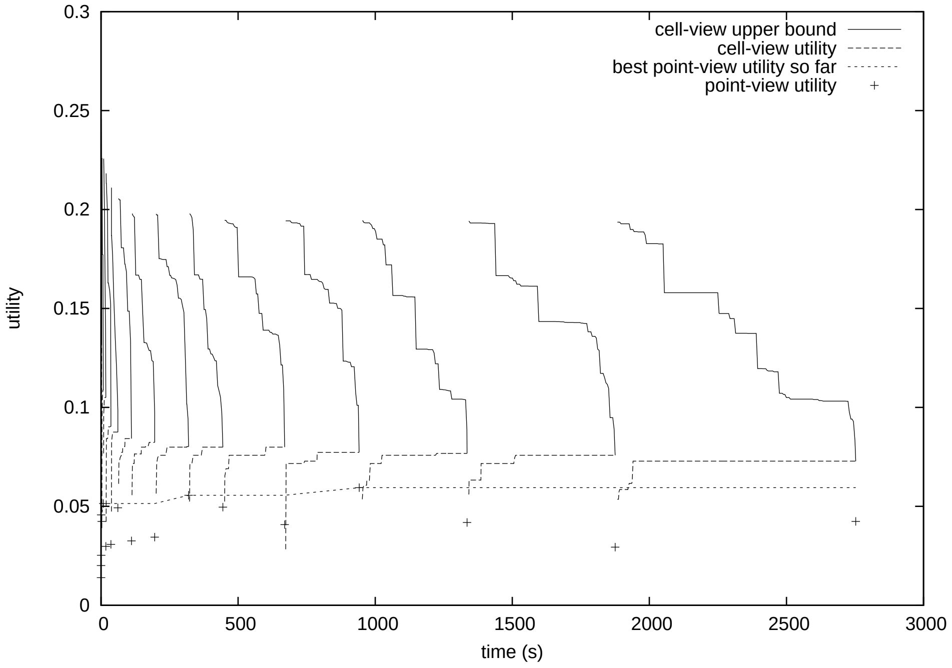


Figure 15: Timings for solving the problem in Figure 13 (b) in the MIP mode.

the exhaustive search terminates.

The *lowest* cell-view upper bound is used as the upper bound for the planar problem, and the solution with the *highest* point-view utility is used as the solution to the planar problem. The search terminates when the ratio of these two values is less than k, which was 1.25 in this example. Figure 15 illustrates the timings for the same problem instance in the MIP mode; this information is based on the logging facility of the GLPK MIP solver.

Figure 16 presents a collection of various relay placement problems. Solving the problems took 276 seconds, 10 seconds, and 18 710 seconds (approx. 5 hours), respectively, in the heuristic mode. In part (b), the exhaustive search always terminated first, while in parts (a) and (c), the heuristic search found solutions to the finite subproblems. The timings for case (c) are shown in Figure 17. The respective times were 6 seconds, 88 seconds, and 8 119 seconds (approx. 2 hours) in the MIP mode, and the timings for this mode are shown in Figure 18.

In summary, we see that the performance of the heuristic mode may be more than an order of magnitude better or worse than the performance of the MIP mode. Which is faster depends on the problem instance. Furthermore, we see that both the heuristic and exhaustive approaches are useful.

Figure 16 (c) also serves as an illustration of combinatorial explosion. One of the problem instances is similar to Figure 16 (c), but the balance parameter λ is 0.5 instead of 0.0. For this instance, finding a 2.0-optimal solution took 113 seconds (there were 58 cells in the last partition), finding a 1.5-optimal solution took 24 380 seconds (approx. 7 hours, 166 cells), and finding a 1.35-optimal solution took as much as 255 363 seconds (approx. 3 days, 226 cells). Clearly, it is not reasonable to expect to achieve, say, 1.25-optimal solutions for all problem instances of even moderate size. However, by maintaining upper bounds and by storing intermediate results, one may terminate the computation at any point and have a solution of known quality available.

In these experiments, the termination criterion for the heuristic mode was 1.1-optimality: the heuristic search is terminated when the cell-view upper bound is at most 1.1 times the cell-view utility. One example of changing this parameter is shown in Figure 19. In the MIP mode, the corresponding constant was 1.0 as the GLPK MIP solver never returns suboptimal solutions. Studying the choice of this parameter and experimenting with approximate MIP solvers is left for future research.

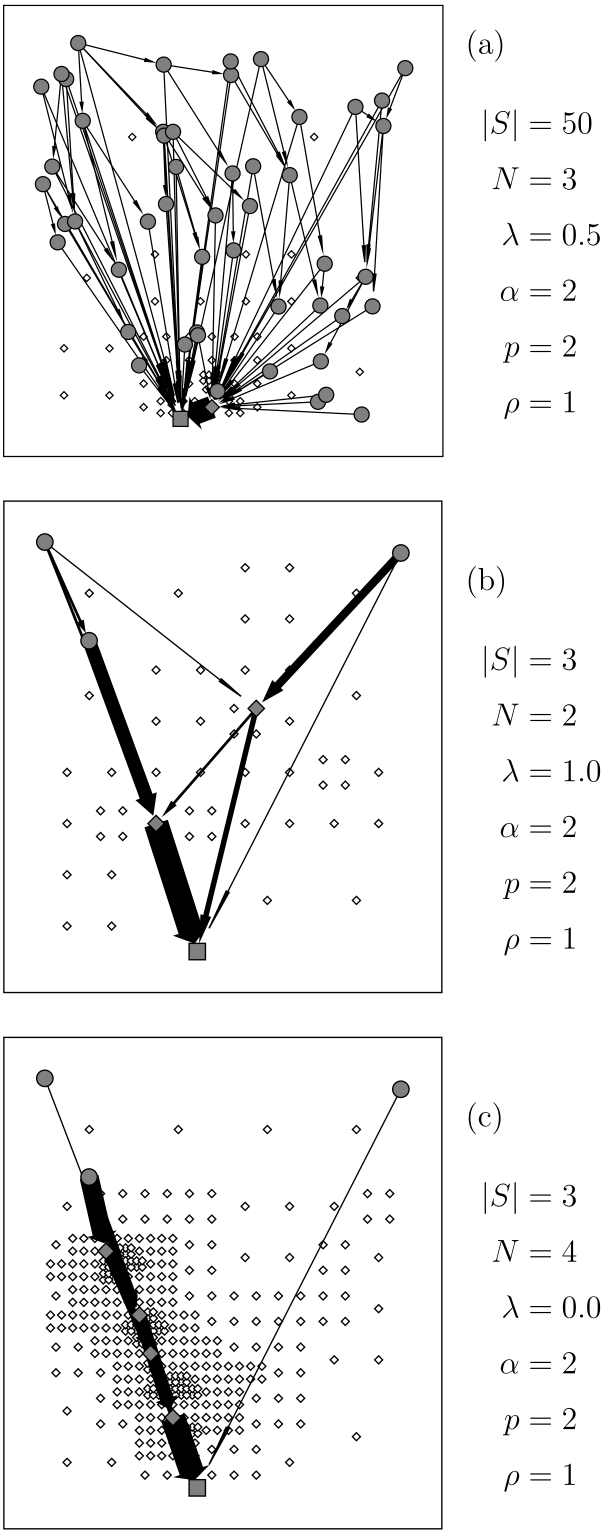


Figure 16: Various examples of relay placement. The solutions are 1.25-optimal or better.

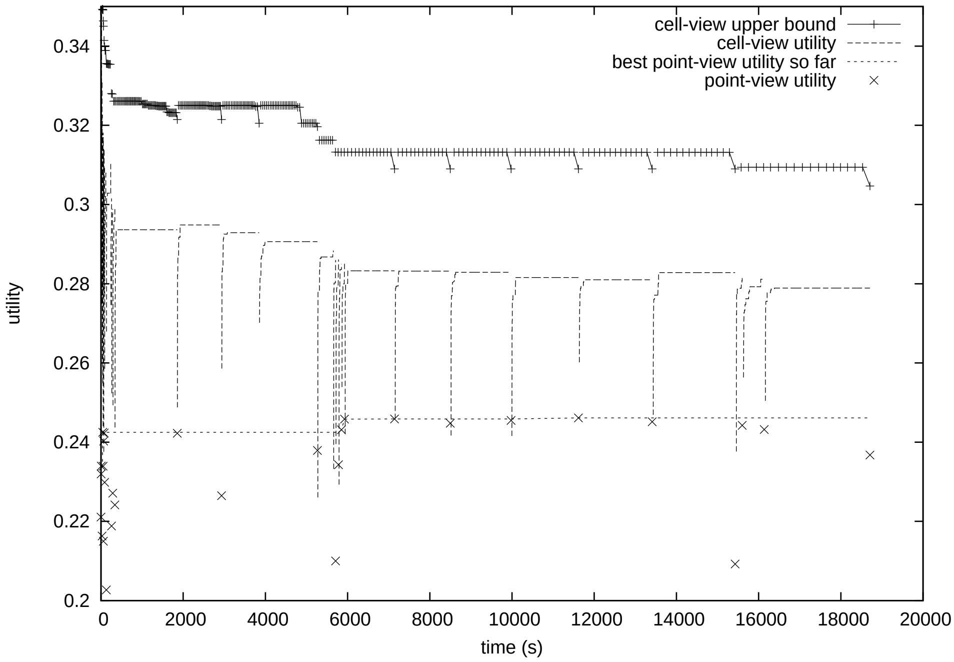


Figure 17: Timings for solving the problem in Figure 16 (c) in the heuristic mode.

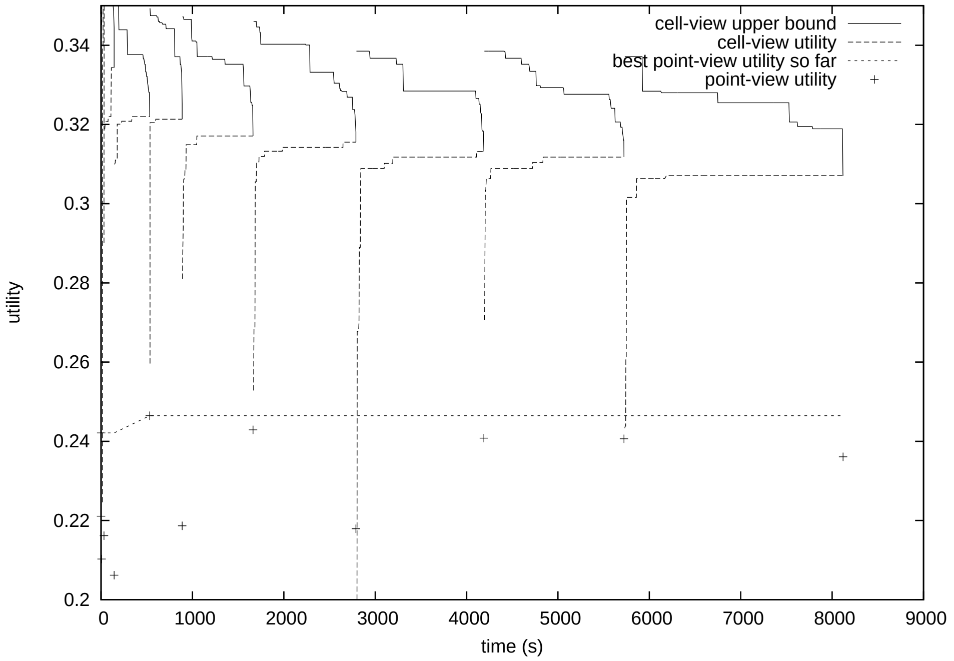


Figure 18: Timings for solving the problem in Figure 16 (c) in the MIP mode.

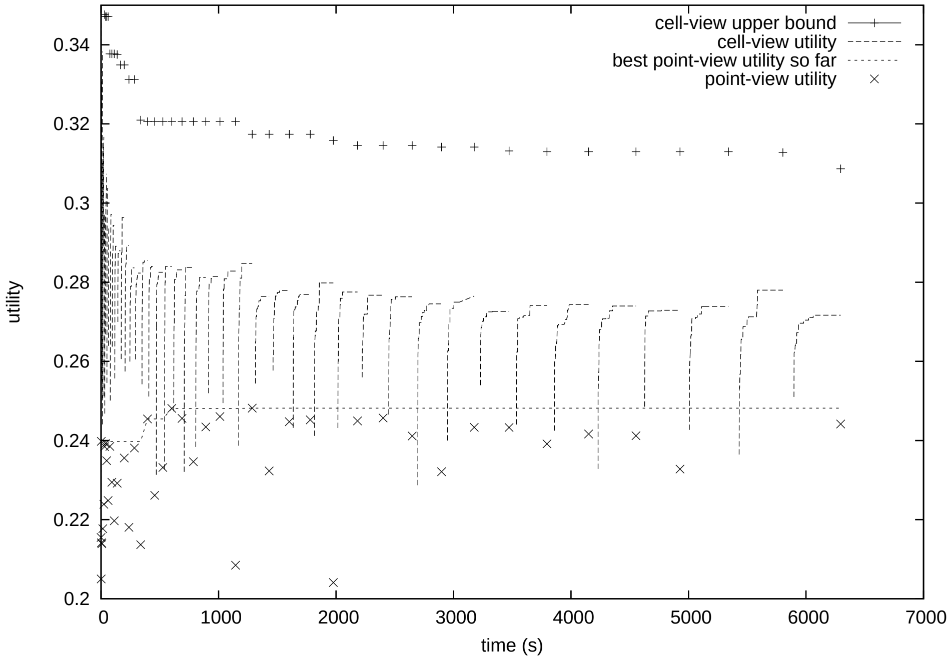


Figure 19: Timings for solving the problem in Figure 16 (c) in the heuristic mode. In this experiment, the finite subproblems were solved 1.2-optimally; compare with Figure 17, where the subproblems were solved 1.1-optimally. Here the heuristic solver typically evaluated only one branch for each finite subproblem. As cell-view upper bounds were looser, the number of steps was higher. However, the total time consumption turned out to be lower.

8 Conclusions and Further Research

In this thesis, I have defined a number of classes of relay placement problems. The computational complexity of these classes has been studied, and all classes have been proved NP-hard. Even approximation of some important problem classes is NP-hard.

I have developed algorithms which may be used for solving finite and planar relay placement problems approximately. While the problems are computationally difficult, the algorithms have been successfully used for solving some problem instances of moderate size. These algorithms provide us with a starting point for future development and a baseline for evaluating alternative approaches. One can also interpret these algorithms as methods for finding guaranteed upper bounds with a given tightness. These upper bounds can be used as a benchmark for new relay placement methods.

The following questions may serve as starting points for future research:

1. Are there efficient approximation algorithms for the free space model or is approximation NP-hard for these cases, too?
2. Is it possible to formulate a relay placement problem which is computationally tractable but still meaningful in practise? Do we need to consider other utility functions instead of the balanced data gathering formulation?
3. In practice, we may be interested in maximising the overall data quality by all possible means, not only by relay placement but also by sensor placement. How can we formulate the problem of optimising data quality, if we consider all relevant aspects such as sensor coverage, redundancy of the data, relay nodes, routing, radio communication, energy constraints, and monetary costs?

While the basic algorithms are now developed, further research needs to be done. We need to study different radio propagation models, imprecise relay placement, different algorithms for local search in the finite solver, different algorithms for local search in the post-processing phase of the planar solver, different implementations and algorithms for solving the LP and MIP problems, approximation algorithms for the LP and MIP problems, parallel implementations of the finite solver, and different approximation ratios for the underlying finite solver.

In general, there are various approaches to each subtask, and the algorithms have tunable parameters. We need empirical or theoretical studies in order to ideally configure the solver for each problem instance, or we need adaptive algorithms which automatically tune their behaviour based on the problem instance at hand.

Acknowledgements

Thanks to Patrik Floréen, Jyrki Kivinen, Tanja Säily, Jukka Kohonen, and Petteri Nurmi for discussions and comments.

References

ABN$^+$02 Adickes, M. D., Billo, R. E., Norman, B. A., Banerjee, S., Nnaji, B. O. and Rajgopal, J., Optimization of indoor wireless communication network layouts. *IIE Transactions*, 34,9(2002), pages 823–836.

AK04 Al-Karaki, J. N. and Kamal, A. E., Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11,6(2004), pages 6–28.

ARY95 Andersen, J. B., Rappaport, T. S. and Yoshida, S., Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, 33,1(1995), pages 42–49.

ASSC02 Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., Wireless sensor networks: a survey. *Computer Networks*, 38,4(2002), pages 392–422.

ČHE02 Čagalj, M., Hubaux, J.-P. and Enz, C., Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2002, ACM Press, pages 172–182.

CES04 Culler, D., Estrin, D. and Srivastava, M., Guest editors' introduction: Overview of sensor networks. *IEEE Computer*, 37,8(2004), pages 41–49.

DKK03 Dasgupta, K., Kukreja, M. and Kalpakis, K., Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. *Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, Washington, DC, USA, 2003, IEEE Computer Society, pages 341–348.

Eph02 Ephremides, A., Energy concerns in wireless networks. *IEEE Wireless Communications*, 9,4(2002), pages 48–59.

ERR94 Erceg, V., Rustako, Jr., A. J. and Roman, R. S., Diffraction around corners and its effects on the microcell coverage area in urban and suburban environments at 900 MHz, 2 GHz, and 4 GHz. *IEEE Transactions on Vehicular Technology*, 43,3(1994), pages 762–766.

FFK$^+$04 Falck, E., Floréen, P., Kaski, P., Kohonen, J. and Orponen, P., Balanced data gathering in energy-constrained sensor networks. *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSORS 2004, Turku, Finland, July 2004)*, Berlin Heidelberg, 2004, Springer-Verlag, pages 59–70.

FKKO05a Floréen, P., Kaski, P., Kohonen, J. and Orponen, P., Exact and approximate balanced data gathering in energy-constrained sensor networks, Theoretical Computer Science, Special Issue on Algorithmic Aspects of Wireless Sensor Networks, 2005. To appear.

FKKO05b Floréen, P., Kaski, P., Kohonen, J. and Orponen, P., Lifetime maximization for multicasting in energy-constrained wireless networks. *IEEE Journal on Selected Areas in Communications*, 23,1(2005), pages 117–126.

GG93 Goldsmith, A. J. and Greenstein, L. J., A measurement-based model for predicting coverage areas of urban microcells. *IEEE Journal on Selected Areas in Communications*, 11,7(1993), pages 1013–1023.

GJ03 Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 2003.

GP90 Grant, I. S. and Phillips, W. R., *Electromagnetism.* John Wiley & Sons Ltd, Chichester, second edition, 1990.

HHKK04 Hill, J., Horton, M., Kling, R. and Krishnamurthy, L., The platforms enabling wireless sensor networks. *Communications of the ACM*, 47,6(2004), pages 41–46.

HMS02 Howard, A., Matarić, M. J. and Sukhatme, G. S., Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02, Fukuoka, Japan, June 2002)*, 2002, pages 299–308.

HP04 Hong, B. and Prasanna, V. K., Constrained flow optimization with applications to data gathering in sensor networks. *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSORS 2004, Turku, Finland, July 2004)*, Berlin Heidelberg, 2004, Springer-Verlag, pages 187–200.

HSW$^+$00 Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K., System architecture directions for networked sensors. *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, New York, 2000, ACM Press, pages 93–104.

ISO99 ISO/IEC, Geneva, *International Standard ISO/IEC 9899: Programming Languages — C*, 1999.

Kar72 Karp, R. M., Reducibility among combinatorial problems. *Complexity of Computer Computations*, Miller, R. E. and Thatcher, J. W., editors. Plenum Press, New York, 1972, pages 85–103.

KDN03 Kalpakis, K., Dasgupta, K. and Namjoshi, P., Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks*, 42,6(2003), pages 697–716.

KEW02 Krishnamachari, B., Estrin, D. and Wicker, S. B., The impact of data aggregation in wireless sensor networks. *Proceedings of the 22nd International Conference on Distributed Computing Systems*, Washington, DC, USA, 2002, IEEE Computer Society, pages 575–578.

Kha79 Khachian, L. G., A polynomial algorithm for linear programming. *Soviet Mathematics. Doklady*, 20,1(1979), pages 191–194. English translation of the original Russian article published in *Doklady Akademii nauk SSSR*, 244,5(1979).

KKK83 Kahn, J., Klawe, M. and Kleitman, D., Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic and Discrete Methods*, 4,2(1983), pages 194–206.

Mak05 Makhorin, A., *GNU Linear Programming Kit, Reference Manual, Version 4.8.* Free Software Foundation, Boston, MA, USA, January 2005.

MKPS01 Meguerdichian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M., Coverage problems in wireless ad-hoc sensor networks. *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2001, pages 1380–1387.

MKQP01 Meguerdichian, S., Koushanfar, F., Qu, G. and Potkonjak, M., Exposure in wireless ad-hoc sensor networks. *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, New York, 2001, ACM Press, pages 139–150.

MS96 Melissen, J. B. M. and Schuur, P. C., Improved coverings of a square with six and eight equal circles. *Electronic Journal of Combinatorics*, 3,1(1996).

Nur00 Nurmela, K. J., Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles. *Experimental Mathematics*, 9,1(2000), pages 241–250.

NÖ00 Nurmela, K. J. and Östergård, P. R. J., Covering a square with up to 30 equal circles. Research Report A62, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, June 2000.

OK04 Ordóñez, F. and Krishnamachari, B., Optimal information extraction in energy-limited wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22,6(2004), pages 1121–1129.

PHC⁺03 Pan, J., Hou, Y. T., Cai, L., Shi, Y. and Shen, S. X., Topology control for wireless sensor networks. *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom*

'03, San Diego, California, USA, September 2003), New York, NY, USA, 2003, ACM Press, pages 286–299.

PS98 Papadimitriou, C. H. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, Inc., Mineola, New York, 1998.

Rap99 Rappaport, T. S., *Wireless Communications, Principles and Practice.* Prentice Hall, Inc., Upper Saddle River, 1999.

RN03 Russell, S. J. and Norvig, P., *Artificial Intelligence: A Modern Approach.* Pearson Education, Inc., Upper Saddle River, New Jersey, second edition, 2003.

RSPS02 Raghunathan, V., Schurgers, C., Park, S. and Srivastava, M. B., Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19,2(2002), pages 40–50.

SE96 Stamatelos, D. and Ephremides, A., Spectral efficiency and optimal base placement for indoor wirelessnetworks. *IEEE Journal on Selected Areas in Communications*, 14,4(1996), pages 651–661.

SK04 Sadagopan, N. and Krishnamachari, B., Maximizing data extraction in energy-limited sensor networks. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, March 2004, pages 1717–1727.

SMP99 Sohrabi, K., Manriquez, B. and Pottie, G. J., Near ground wideband channel measurement in 800–1000 MHz. *Proceedings of the 49th Vehicular Technology Conference*, 1999, pages 571–574.

SP01 Slijepcevic, S. and Potkonjak, M., Power efficient organization of wireless sensor networks. *IEEE International Conference on Communications*, 2001, pages 472–476.

SR92 Seidel, S. Y. and Rappaport, T. S., 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Transactions on Antennas and Propagation*, 40,2(1992), pages 207–217.

SWR98 Singh, S., Woo, M. and Raghavendra, C. S., Power-aware routing in mobile ad hoc networks. *Proceedings of the 4th Annual ACM/IEEE*

International Conference on Mobile Computing and Networking, New York, NY, USA, 1998, ACM Press, pages 181–190.

TS91 Teller, S. J. and Séquin, C. H., Visibility preprocessing for interactive walkthroughs. *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, New York, 1991, ACM Press, pages 61–70.

Urr00 Urrutia, J., Art gallery and illumination problems. In *Handbook of Computational Geometry*, Sack, J.-R. and Urrutia, J., editors, Elsevier Science B.V., Amsterdam, The Netherlands, 2000, chapter 22, pages 973–1027.

ZC03 Zou, Y. and Chakrabarty, K., Uncertainty-aware sensor deployment algorithms for surveillance applications. *IEEE Global Telecommunications Conference*, 2003, pages 2972–2976.

Appendix 1. Table of Symbols

The mathematical symbols used in this text are summarised in the following table.

B	instance of the balanced data gathering problem
$\mathcal{B}$	set of all balanced data gathering problems
P	instance of the relay placement problem
$\mathcal{P}, \mathcal{P}_x, \mathcal{P}_P, \mathcal{P}_D, \ldots$	classes of relay placement problem instances
$\mathcal{P}^*$	set of all relay placement problem classes
$B(P, R)$	converted instance
k	approximation ratio
λ	balance parameter
V	set of nodes
$\mathcal{V}$	set of possible nodes
S	set of sensor nodes
S^+	set of sensor and sink nodes
$R, R^*, \tilde{R}$	sets of relay nodes: any, optimal, approximated
$\mathcal{R}$	set of possible relay nodes
N	number of relay nodes
σ	sink node
$\eta, \eta_i, \kappa, \mu, \nu$	nodes
ρ	reception cost
$\tau(\eta, \kappa), \tau_{\eta\kappa}, \tau'(l_\eta, l_\kappa)$	transmission cost
$E(\eta), E_\eta, E$	energy supply of a node
s_η	available data at a sensor node
$l(\eta), l_\eta, l$	locations
$\overline{l_1 l_2}$	line segment
$w(\eta), w_\eta, w$	identities
W	set of identities
o_i, o	obstacles
O	set of obstacles
$\#o, \#O$	number of vertices in obstacles
$\Lambda, \Upsilon, \Xi, \Psi, T$	parts of geometry
$f, f^*, \tilde{f}$	flows: any, optimal, approximated

$\mathcal{F}(B)$	set of feasible flows
$q_\eta(B, f)$	amount of data collected from a sensor node
$F(B, f)$	utility of a balanced data gathering solution
$U(P, R)$	utility of a relay placement problem solution
u	utility requirement
$PL(l_1, l_2)$	radio communication path loss
α	exponent for the power law
$d(\mathbf{x}, \mathbf{y})$	Euclidean distance
$d_p(\mathbf{x}, \mathbf{y})$	distance with p-norm
p	exponent for the distance function
$a, b, c, i, j, m, n, t, x, y, z, \epsilon$	scalars
$\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{y}, \mathbf{z}$	vectors
z_i	vector elements
g, h	functions
A, X, Y, Z	sets
$\mathcal{A}, \mathcal{X}, \mathcal{Y}$	collections of sets