



**HAL**  
open science

# Power-Aware Protocols for Wireless Sensor Networks

Chuan Xu

► **To cite this version:**

Chuan Xu. Power-Aware Protocols for Wireless Sensor Networks. Distributed, Parallel, and Cluster Computing [cs.DC]. University Paris Sud, 2017. English. NNT: . tel-01758877v1

**HAL Id: tel-01758877**

**<https://hal.science/tel-01758877v1>**

Submitted on 4 Apr 2018 (v1), last revised 20 Feb 2018 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2017SACLS498

THÈSE DE DOCTORAT  
DE  
L'UNIVERSITÉ PARIS-SACLAY  
PRÉPARÉE À  
L'UNIVERSITÉ PARIS-SUD

Ecole doctorale n°580  
Sciences et technologies de l'information et de la communication  
Spécialité de doctorat : Informatique  
par  
**MLLE. CHUAN XU**  
Power-Aware Protocols for Wireless Sensor Networks

Thèse présentée et soutenue à Orsay, le 15 Décembre 2017 :

Composition du Jury :

M.	ABDEL LISSER	Professeur Université Paris-Sud	Président du jury
M.	LUÍS E. T. RODRIGUES	Professeur Université de Lisbonne	Rapporteur
M.	ALEXANDRE CAMINADA	Professeur Université de Technologie de Belfort-Montbéliard	Rapporteur
Mme.	JANNY LEUNG	Professeur Université chinoise de Hong Kong	Examinatrice
M.	JOFFROY BEAUQUIER	Professeur Université Paris-Sud	Directeur de thèse
Mme.	JANNA BURMAN	Maître de conférence Université Paris-Sud	Co-encadrante
M.	THOMAS NOWAK	Maître de conférence Université Paris-Sud	Invité



# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my Ph.D advisors, Prof. **Joffroy Beauquier** and Prof. **Janna Burman**. They have been supportive since the days I began working with them in the domain of distributed computing. They provided invaluable guidance throughout my three years' research, taught me the methodology to carry out the research and to write research papers as clearly as possible. I have learned a lot from them. They gave me a lot of moral supports when difficulties were encountered in my research. I am very grateful for their great efforts to train me in the scientific field. It was really a great privilege and honor to work and study under their guidances.

I would like to thank Prof. **Abdel Lisser**. He has guided me since my undergraduate project. He inspired me to pursue a research career in research, provided me the opportunities to explore interesting research topics and to collaborate with other professors. He gave me a very valuable advice and witnessed my growth as a foreign student in France. I am very grateful for his academical and emotional support during my seven years' stay in France.

I would like to express my gratitude to the reporters, Prof. **Luis E.T. Rodrigues** and Prof. **Alexandre Caminada**, for investing time and providing interesting and valuable feedback to this thesis. I am thankful to Prof. **Qianqiang Cheng**, Prof. **Alexei Gaivoronski**, Prof. **Shay Kutten**, Prof. **Janny Leung**, Prof. **Thérèse Malliavin**, Prof. **Thomas Nowak** and Prof. **Franz Rendl** for their valuable time, efforts and suggestions to the cooperative work.

I am extremely grateful to my **parents**, for their loves, caring and sacrifices to bring me up and give me support in every aspect of my life. I would like to thank also my friends, **Youfang Peng**, **Qiang Sun**, **Zhu Zhu**, **Jianzhi Fu**, **Gu Tang**, **Yijing Chen**, **Aygun Jamal**, **David Declerk**, **Alexandre Dambreville**, **Christian Hamelain**, **Marie Laveau**, **Fabien Dufoulon** for their supports, understandings and wonderful time spent together, and **Denis**, **Boubacar**, **Fati**, **Myriam Soumaré** for taking me as their family member since my first year in France. I am very pleased to have all of you by my side. Finally, I would also like to thank all the colleagues in the **ParSys group**, the **administration team** and the **technical support team** at Laboratoire de Recherche en Informatique.



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Main Results and Outlines . . . . .	6
1.2	Preliminaries . . . . .	7
1.2.1	Wireless sensor networks . . . . .	7
1.2.2	Population protocols . . . . .	8
1.2.3	Wireless body area networks . . . . .	12
<b>2</b>	<b>Power-Aware Population Protocols</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.1.1	Overview and outline . . . . .	18
2.1.2	Related work . . . . .	18
2.1.3	Assumptions . . . . .	19
2.2	Energy Consumption Scheme and Metrics . . . . .	20
2.2.1	Energy saving schemes in wireless sensor networks . . . . .	20
2.2.2	On-demand scheme for population protocols . . . . .	22
2.3	Energy Consumption of TTFM . . . . .	23
2.4	Energy-Balanced Protocol EB-TTFM . . . . .	25
2.4.1	Energy Consumption Analysis . . . . .	26
2.4.2	Special Case of Non-Bounded Memory ( $M \geq n - 1$ ) . . . . .	32
2.5	Lower bound on $Es_{max}$ . . . . .	36
2.6	Conclusion . . . . .	39
<b>3</b>	<b>Power-Aware Population Protocols under Non-uniformly Random Scheduler</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.1.1	Related work . . . . .	42
3.2	Model and Definitions . . . . .	43
3.3	Lower Bounds on the Expected Convergence Time . . . . .	44
3.4	Protocol “Transfer To the Faster” (TTF) . . . . .	46
3.5	Lazy TTF . . . . .	50
3.5.1	Convergence time of lazy TTF . . . . .	51
3.6	Numerical Results . . . . .	53
3.6.1	The relation between $T_{whp}(\text{TTF})$ and its upper bound . . . . .	53
3.6.2	TTF vs. Lazy TTF( $\hat{p}$ ) in terms of time complexity bounds . . . . .	53
3.6.3	Gaps on time and energy between TTF and Lazy TTF( $\hat{p}$ ) . . . . .	55
<b>4</b>	<b>An Optimization Model for Data Collection in Wireless Body Area Networks</b>	<b>57</b>
4.1	Wireless Body Area Networks . . . . .	57
4.2	Problem Formulation . . . . .	58
4.3	Near Optimal Solution: Meta-heuristics approach . . . . .	59
4.3.1	Variable neighborhood search . . . . .	60

4.3.2	Communication topology . . . . .	60
4.3.3	Design of VNS for WBAN problem . . . . .	61
4.4	Numerical Results . . . . .	63
4.4.1	Numerical results for the VNS algorithm . . . . .	63
4.4.2	Weighted objective function . . . . .	65
4.4.3	Average numerical results . . . . .	65
4.5	Conclusion . . . . .	66
<b>5</b>	<b>Stochastic Semidefinite Optimization Using Sampling Methods</b>	<b>69</b>
5.1	Introduction . . . . .	69
5.1.1	Chance constrained programs . . . . .	70
5.1.2	Methods to solve chance constrained program . . . . .	70
5.1.3	The problem . . . . .	72
5.1.4	Overview . . . . .	72
5.2	Chance Constrained Semidefinite Programs . . . . .	73
5.3	Simulation-based Approximation . . . . .	73
5.3.1	Scenario approach . . . . .	73
5.3.2	Sample average approximation - Big-M sampling approach . . . . .	73
5.3.3	Combination of Big-M and constraints discarding . . . . .	74
5.4	Numerical Experiments . . . . .	74
5.4.1	Control system problem . . . . .	75
5.4.2	Experiment approaches . . . . .	76
5.4.3	Design of the experiments . . . . .	77
5.4.4	Numerical results . . . . .	78
5.5	Conclusion . . . . .	81
<b>6</b>	<b>Conclusion</b>	<b>83</b>
6.1	Summary . . . . .	83
6.2	Perspectives . . . . .	84
	<b>Appendix</b>	<b>101</b>
<b>A</b>	<b>Semidefinite Programming</b>	<b>103</b>
A.1	Positive semi-definite matrix . . . . .	103
A.2	Semi-definite programming . . . . .	103
A.2.1	Linear matrix inequality . . . . .	103
A.2.2	Maximum eigenvalue minimization . . . . .	104
<b>B</b>	<b>Synthèse en français</b>	<b>105</b>





# Introduction

---

Nowadays, with the development of standardizations on networking, the ideas of "Internet of Things"<sup>1</sup>, "Smart City"<sup>2</sup> and "ubiquitous computing"<sup>3</sup> have been proposed and attracted great attentions from researchers and industries. The common point of these innovations is to make use of our physical world to facilitate daily lives of human beings. Computing and communicating units are implanted into physical objects (as, e.g., cars, lamps, electronic devices, buildings and products), to collect and exchange data through wireless networks without any human intervention. The collected information allows to interact with the environment in real time.

For instance, in the Libelium Smart Cities design, sensors are implanted in the tunnels to monitor the air quality and traffic, put into containers of rubbish to optimize the trash collection routes, integrated into buildings to monitor real time noises and air pollutions, installed on bridges and historical monuments to monitor material conditions. Similar ideas and techniques have already been tested and applied in cities like Oslo, San Francisco and Singapore.

For deploying such real time information systems, well performing *wireless sensor networks* (WSNs) are essential and indispensable. Small, rugged, inexpensive and low power wireless sensors are required to reach any kind of environment at reasonable costs. At the same time, the sensed data has to be collected efficiently in a central unit, called the *base station*, where it is analyzed or delivered to the Internet for a real time sharing. A crucial property for such systems is to be power aware, that is to consume energy as little as possible.

In most of the available sensors in the market, the primary power supply is two fixed typical AA batteries, like Telos, Iris or Mica. However, even the ultra low power sensor Telos can only achieve a maximum of 241 hours of node operation [83]. From a long term perspective, the notion of power-awareness helps to reduce the operations and maintenance costs. Moreover, in some cases, when the moves of the sensors are not under control, it may be difficult or undesirable to access these sensors frequently (cf., Bird Species Recognition [29], ZebraNet [80]). Therefore, once a sensor depletes its available energy, to obtain additional energy is almost impossible. This may cause a significant deterioration of the performance of the network. For instance, the sensors out of energy can break the network into two disconnected parts. If this happens, neither broadcasting nor gathering of information could ever be achieved. In recent years, the techniques for harvesting energy from the

---

<sup>1</sup>The Internet of Things is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.

<sup>2</sup>A smart city is an urban development vision to integrate information and communication technology and the Internet of Things technology in a secure fashion to manage city's assets.

<sup>3</sup>Ubiquitous computing aims to make computing to appear anytime and everywhere.

environment (e.g. solar or wind energy) have been developed for wireless sensor networks, enabling sensors to increase their lifetime [83]. However, such techniques are highly constrained by the weather conditions, which introduce uncertain factors to the performance. Hence, power-aware designs are required and studied at every level of operation (like sensing or communicating). For example, compressed sensing has been studied to preserve energy when the signal is sparse and compressible [112]. Global Positioning System (GPS) is integrated to provide location information for sensors, which allows them to adjust the transmission power for a communication [80].

In this thesis, a general interest is to design power-aware protocols for wireless sensor networks. More precisely, we design protocols, which consider energy consumption, for the fundamental task of *data collection*. The goal of data collection is to collect at the base station all the data (e.g., temperature, pressure or altitude) sampled by the sensors, for further data processing. For example, gathering the data of soil moisture tension in the fields allows to better regulate irrigation [39]. Collecting the infrasonic signals generated by volcanic activities helps to predict paroxysmal events and triangulate precise locations of individual eruptions [139]. Having a real-time physiological data of a patient, enables disease diagnosis and remote assistance [4]. Note that data collection differs from *data aggregation*, which aims at eliminating redundant data transmissions and provides fused information to the base station [110]. Data aggregation is not within the scope of this thesis. Refer to Sect. 1.2.2 for a formal definition of data collection used here.

The first and essential step in our study is to choose a proper network model. In common designs, factors that may affect the network performance (e.g., noise of channels, delays of packets, mobility patterns of the nodes, network faults) are involved as many as possible, in order to make the model more realistic. These factors are presented by parameters, obtained from experiments or experiences. They could be deterministic, stochastic or just "totally" randomized. Such a network model is complex and complicated, making the pure theoretical analyses hard or even impossible to perform. Therefore, in most cases, designs are validated by simulations, i.e., tested on benchmarks to demonstrate their performances on computations. Indeed, simulations could be useful for intuitive comparisons among different approaches. However, these results are generally not truly objective, because they are highly affected by the computing resources, the coding levels, the lack of considerations of other scenarios, etc. Moreover, sometimes, it is hard to explain the reasons for the good performances appearing in the simulations.

Contrary to the simulation approach, we consider at first a distributed computing model, called *population protocols*, which allows purely analytical analysis for a specific type of WSNs. In this model, nodes have limited power resources and communicate only in pairs. They have no identifiers and their memory size is uniformly bounded (i.e., independent of the number of nodes). This model has a wide domain of applications, especially in the case where the mobility of nodes are unpredictable, such as in mobile sensor networks with resource-limited communicating devices attached to animals, humans or vehicles (cf., habitat monitoring applications of Bird Species Recognition [29] or ZebraNet [80]). However, this model does not have an energy framework, considering in some way the energy consumption of its entities

during executions. Hence, for being able to design power-aware protocols, we enhance the population protocols model with an energy consumption scheme. Even though this enhancement considers and models energy consumption of real mobile WSNs, it is still compact and elegant enough to allow a pure analytical analysis of energy or other aspects of efficiency. Then, using this enhanced model, we develop and analyze new power-aware data collection protocols and establish explicit upper and lower bounds on energy consumption. This demonstrates the usefulness of the proposed energy model.

Note that for making the protocols power-aware, parameters (related to energy consumption, evolving during an execution) should be integrated into the design to save energy when an abnormal consumption is observed. Therefore, in our power-aware design, choosing such proper parameters for protocols, is an important consideration. However, when a large number of parameters are involved, this may cause large problem dimensions and a lot of dependencies between parameters and bounds. This may make the problem hard, or even impossible, to be solved analytically. That is why we adopt optimization techniques: giving a mathematical model taking parameters as variables and optimizing the analytical bound for the performance.

In the second part of this thesis, we try to apply the same methodology for analyzing energy consumption, to a completely different type of networks. We choose the *wireless body area network* (WBAN) model, as energy is a crucial factor to be considered in its design. WBAN consists of bio-sensors implanted into the body of human beings for health-care. Power-awareness is the main issue because humans are sensitive to the heat generated by the sensors and recharging the battery is definitely not comfortable.

Together with that, there are many differences between WBAN and networks modeled by population protocols. The former are small and almost static sensor networks using neighboring broadcast communications, while the latter are large scale mobile peer-to-peer networks. Especially due to this fact, WBAN has a particular interest for this thesis. The interest is in performing a broader study of power-aware data collection concerning very different types of WSNs. This is for learning the differences and the similarities between their protocol design approaches as well as between the methods useful for analyzing these protocols.

Compared to population protocols, data collection in WBAN can be managed off-line, i.e., the path for delivering values to the base station can be programmed in advance. It stems from the fact that WBAN is a small WSN of no more than 256 sensors, whose memory sizes are of kilobytes [85]. Hence, each sensor unit in a WBAN could have its own unique identifier and the identifiers of its neighbors (contrary to population protocols). In addition, since the communication range of each sensor in WBAN is adjustable by changing its transmitting power, the communications between sensors are not as opportunistic as in population protocols (which are almost completely unpredictable). Therefore, to accomplish data collection in a WBAN, optimization techniques can be applied to have an optimal strategy for each sensor, by modeling the routing problem and solving this model centrally. Such an off-line solution obtained by combinatorial optimization is generally impossible in the case of population protocols. Nevertheless, and maybe somewhat surpris-

ingly, we find such an approach applicable and useful also in the case of population protocols (See Chapter 3).

One of the main insights of this thesis is that, although population protocols and WBAN are networks of different natures, having different characteristics, there is a common point in both approaches for power-awareness. This common point is the benefit to use optimization techniques. The main reason is that power-awareness takes into account a large number of parameters, which can be dealt with by optimization.

## 1.1 Main Results and Outlines

**Main results.** In the first part of this thesis, we study the complexities of data collection in population protocols, in terms of time and energy. The contribution is fourfold. First, a model for analyzing energy consumption in networks of mobile sensors is introduced (Chapter 2). This model can be considered as an extension to the classical model of population protocols, which takes into account energy consumption. The interest of the extended model is to allow a purely analytical analysis of the energy complexity of a protocol, in the same spirit as for time and space complexity, without appealing to simulations. Second, this model is used for studying energy complexity of distributed protocols for the task of data collection. Starting from analyzing the energy complexity of the existed time-optimal data collection protocol, called TTFM (given in [26]), we propose and analyze a new power-aware version of it, called EB-TTFM. Then, lower bounds on energy complexity for any data collection protocol are established. This demonstrates also the cases where these lower bounds are reached by the proposed protocols (Chapter 2). Third, the study of non-uniform random fairness (a generalized random fairness) is initiated in the context of population protocols, where a meeting pair is independently selected from the population with a discrete probability (Chapter 3). Fourth, a power-aware population protocol for data collection is then developed to justify the relevance and operability of this enhanced model, which integrates also the optimization approach. This demonstrates that optimization can be applied to the design of parametrized distributed algorithms.

In the second part of this thesis, we adopt optimization approach to study power-aware data collection in wireless body area networks. While, to have a better and profound comprehension on methods for solving optimization models, we study a stochastic semi-definite program which is complex but applicable to many realistic problems. In addition, this stochastic model is a variant of the one proposed previously, in the first part. Studying it in depth helps to combine mathematical optimization and distributed computing for future research. The contribution in the second part is twofold. First, a minmax multi-commodity netflow formulation is proposed to optimally route data packets in a wireless body area network (Chapter 4). Second, a simulation-based approximation method is studied to solve a stochastic semi-definite program (Chapter 5). This method can be easily applied to solve other optimization problems casted into the same model.

**Outlines.** Chapter 2 consists of three parts. First, the formal energy model designed for population protocols is presented. Second, the energy worst-case complexity of an extended version of the existed data collection protocol TTFM is studied. Then, a power-aware version of TTFM, called EB-TTFM (Energy-Balanced TTFM), is presented with analytical formulas for energy consumption. EB-TTFM takes residual energy into account and improves TTFM in terms of the maximum energy spent by an agent. At the end, lower bounds for energy consumption are established, and the cases where these lower bounds are reached, are presented.

Chapter 3 enhances the classical population protocols with the novel non-uniform random fairness. Explicit lower bounds on the expected convergence time of any data collection protocol are given. Followed by an average-case analysis on time for a simple data collection protocol, the relevance of this enhanced model in protocol analysis and its operability are justified. Moreover, a new parametrized power-aware protocol is presented and a polynomial-time optimization problem is formulated to obtain optimal parameters. At last, numerical results are presented to show the efficiency of this power-aware protocol.

Chapter 4 presents a minmax multi-commodity netflow optimization model for the data collection in wireless body area networks, which is transformed then into a mixed integer linear program. After solving this program, each sensor node obtains a decision on its activation and the flows to transmit during one data collection. A variable neighborhood search approach is proposed to obtain a near-optimal solution to this program. Numerical experiments on small instances are conducted in the end to show the efficiency of our approach and the quality of our solutions.

Chapter 5 studies the methods to solve chance constrained semi-definite programs. It can be seen as an invariant of the model applied in Chapter 3 for designing distributed algorithms. A novel simulation-based method is then proposed, which obtains less conservative solutions within reasonable cpu time. Experiments of our methods on a real control theory problem are conducted then, which demonstrate the usefulness of the proposed method.

## 1.2 Preliminaries

### 1.2.1 Wireless sensor networks

Wireless sensor networks consist of tiny, lower powered sensors, distributed in the environment to achieve functionalities like monitoring, detection and communication. The research of WSNs starts in the 80's for military applications, including battlefield surveillance, forces monitoring, battle damage assessment et al.

In the 90's, civilian applications of sensor networks became possible with the advances in the fabrication of sensors. The sizes of sensor can range from a pack of cards to dust particle within an affordable price. In addition, network research and developments on highly dynamic ad hoc environments make the wireless sensor networks applicable to more and more realistic scenarios. In industry, sensors

are attached to the goods and the products, to track their positions during the transportation. In a smart home, sensors are installed into vacuum cleaners, microwave ovens, and refrigerators, allowing owners to control home devices locally and remotely.

WSNs are also applied for habitat monitoring [90], like helping biologists to settle large-scale land-use issues affecting animals like birds, cows and whales. It prevents the potential disturbances of human presence during the research of life sciences. For instance, audio sensors are used to monitor bird species [29]. In the ZebraNet project [80], sensors are attached to wild-life zebras to track their movements in central Kenya. In the Pigeon Air Patrol Network project<sup>4</sup>, sensors are carried by the pigeons in London to measure nitrogen dioxide and the real time measurement is accessible to the public by the Internet. For this kind of networks, the communication graph is always changing and the network is thus opportunistic.

### 1.2.2 Population protocols

In this thesis, the first model studied, *population protocols* [14], is a theoretical distributed computing model proposed in 2004. It can simulate collections of molecules undergoing chemical reactions and passively mobile sensor networks (e.g., a flock of birds equipped with sensors, WSNs for habitat monitoring). In this model, tiny indistinguishable agents (sensors) with bounded memory<sup>5</sup> move unpredictably. They have no identifiers and execute all the same code. As their communication power is limited, they do not communicate by broadcasting, but only, in pairs. That is, when two agents are sufficiently close to each other, they can communicate (i.e., an *interaction* happens). During an interaction, they exchange and update their respective states according to a *transition rule* (the protocol). Such successive interactions contribute to the realization of some global tasks.

Though many restrictions are imposed on the agents in population protocols (like limited memory size or the lack of identifiers), the computational power of this model is not trivial. In [15], the authors showed that any predicate in the class of Presburger arithmetic<sup>6</sup> can be computed. Later in [16], it was proven that in a probabilistic population protocol model, where all pairs of agents are equally to be chosen for each interaction, semilinear predicate can be computed in  $O(n \log^5 n)$  interactions, provided a unique leader in the initial population. Then, the fundamental tasks and topics of distributed computing, like leader election [10, 51], majority [11], counting [22], fault tolerance [18] and self-stabilization [19] have been studied. Their performances in the probabilistic population protocol model are evaluated by the expected (parallel) interactions (time complexity) and by the number of states available at each agent (space complexity). In [6], the authors showed that any leader election or majority protocol converges in  $o(n/\text{polylog } n)$  parallel time using  $\Omega(\log \log n)$  states. Then, a space optimal leader election was presented,

---

<sup>4</sup><http://www.pigeonairpatrol.com/>

<sup>5</sup>The size of their memory is uniformly bounded, independently of the total number of sensors in the network.

<sup>6</sup>The language of Presburger arithmetic contains constants 0 and 1 and a binary function +, interpreted as addition, relations <, >, =, ≤, ≥ and the standard connectives and quantifiers of first-order predicate calculus.

which requires only  $O(\log \log n)$  space usage and converges in  $O(\log^2 n)$  parallel time. Another trade-off between time and space complexity in population protocols was shown later in [8]. It stated that any majority protocol which converges in  $O(n^c)$  ( $c < 1$ ) parallel time, requires  $\Omega(\log n)$  space. Then, the authors proposed a space optimal majority protocol, using  $O(\log n)$  states and converging in  $O(n \log n)$  parallel time. The counting problem, where a distinguished agent (called the base station) counts a non-initialized population, can be solved in time  $O(n \log n)$ , when using an optimal space of only one bit [22]. The community detection problem, asking for each agent to recover its community in a regular clustered communication graph, can be solved in  $O(\log^2 n)$  parallel time using  $O(\log^2 n)$  states [28]. On the other hand, self stabilizing leader election (SSLE) in population protocols was proven to be impossible for arbitrary graphs [19]. By applying the oracle  $\Omega?$ , which is able to detect the presence or the absence of leaders, SSLE can be solved in the complete graphs using just one bit of memory space [57]. Moreover, by using a composition of two copies of  $\Omega?$ , there exists a solution for SSLE for arbitrary graphs [24]. However, without an oracle, for solving SSLE in a complete graph, under global fairness, at least  $n$  states are needed and the best known protocol converges in  $O(n^2)$  parallel time in the probabilistic model [35].

### Formal definition

Formally, a system consists of a collection  $\mathbf{A}$  of pairwise interacting agents, also called a population. The size of the population  $|\mathbf{A}| = n$  is unknown to the agents. Among the agents, there is a distinguishable agent called the *base station* (BST), which can be as powerful as needed, in contrast with the resource-limited non-BST agents. The non-BST agents are also called *mobile*, interchangeably. Each agent has a state taken from a finite set of states. When two agents  $i$ , in state  $p$ , and  $j$ , in state  $q$ , interact (meet), they execute a *transition*  $(p, q) \rightarrow (p', q')$ . As a result,  $i$  changes its state from  $p$  to  $p'$  and  $j$  from  $q$  to  $q'$ . The table  $\mathbf{T}$  of all the transition rules defines the population protocol. A protocol is called *deterministic*, if for every pair of states  $(p, q)$ , there is exactly one  $(p', q')$  such that  $(p, q) \rightarrow (p', q')$ . Otherwise, they are *non-deterministic*. Note that, as interactions are supposed to be asymmetric (with one agent acting as the *initiator* and the other as the *responder*), the transition rules for  $(p, q)$  and  $(q, p)$  may be different.

A *configuration* of the system is defined by the vector of agents' states. If, in a given configuration  $C$ , a configuration  $C'$  can be obtained by executing one transition of the protocol (between two interacting agents), it is denoted by  $C \rightarrow C'$ . An *execution* of a protocol is a sequence of configurations  $C_0, C_1, C_2, \dots$  such that  $C_0$  is the *initial configuration* and for each  $i \geq 0$ ,  $C_i \rightarrow C_{i+1}$ . The sequence of the corresponding interactions in an execution is provided by an external entity called *scheduler*, whose behavior will be discussed later in Sect. 1.2.2.

An execution is said to be *finite*, if and only if from some point onwards, no applicable transition changes the configuration. In this case, this non-changing configuration is said to be *terminal*. When a terminal configuration is reached, we say that the *termination* has occurred. If an execution  $e$  is finite, its length  $|e|$  is

the number of interactions until termination.

Usually, the length of an execution is considered as the time reference, i.e., each interaction adds one time unit to the global time. This is similar to the *step complexity*, a common measure in population protocols (cf. [7, 15]) and in distributed computing in general [128].

### Communication graph

Since the mobilities of agents may have physical constraints, it is possible that, due to the physical distances, an agent never meets some other agents far away from itself. A communication (directed) graph  $G = (\mathbf{A}, \mathbf{E})$  illustrates the possible interactions in population protocols, where  $\mathbf{A}$  is the set of nodes and  $\mathbf{E}$  is the set of edges. A directed edge  $(i, j) \in \mathbf{E}$  intuitively represents a possible interaction between two agents, where agent  $i$  is the initiator and agent  $j$  is the responder.

### Scheduler and fairness

In population protocols, the interactions are unpredictably chosen from  $\mathbf{E}$  and ordered by the *scheduler*. Whatever the order of interactions is, a correct population protocol should reach a terminal configuration.

However, if the scheduler chooses to isolate agents in groups and orders only the interactions between agents from the same group, no protocol could ever realize any non-trivial task correctly. Thus, a *fairness* condition is introduced for the scheduler to restrict its behavior. In the following, three different fairness conditions appeared in the literature of population protocols, are presented: *global fairness*, *random fairness* and *cover time fairness*.

**Global Fairness.** Global fairness imposes constraints on configurations appearing in an execution. The basic idea is that if in an execution there is an infinitely often reachable configuration, then it is infinitely often reached [15]. More formally, an execution is said *globally fair*, if for every two configurations  $C$  and  $C'$  such that  $C \rightarrow C'$ , if  $C$  occurs infinitely often in the execution, then  $C'$  also occurs infinitely often in the execution.

Notice that this fairness does not ensure that every interaction happens infinitely often, it requires only that certain configurations appear in a fair execution no matter which transitions are used. Global fairness is certainly the most common fairness assumption appearing in the literature on population protocols. For instance in [15], the authors show that any predicate in the class of Presburger arithmetic<sup>7</sup> can be computed under global fairness. Or in [25], the authors design a counting protocol, which requires only one bit of memory under global fairness.

**Random Fairness.** Random fairness imposes to the scheduler to choose interacting pairs independently and uniformly at random from  $\mathbf{E}$  [15]. More formally, an

---

<sup>7</sup>The language of Presburger arithmetic contains constants 0 and 1 and a binary function  $+$ , interpreted as addition.



execution is said *randomly fair*, if for each interaction in the execution, a pair of agents in  $\mathbf{E}$  is chosen uniformly at random.

Random fairness gives a natural notion of *parallel time* equal to the total number of interactions divided by the number of agents. Indeed, as each agent participates in an expected number of  $\Theta(1)$  of interactions per parallel time unit, this fairness allows an average-case analysis on the parallel time of a protocol (cf. leader election protocols [10, 51], exact majority protocols [11]).

In chapter 3, a novel generalized random fairness is introduced, where each interacting pairs  $(i, j)$  is chosen with a discrete (not necessarily uniform) probability  $P_{i,j}$  such that  $\sum_{(i,j) \in \mathbf{E}} P_{i,j} = 1$ . Such a scheduler is called *non-uniformly random scheduler*, which is one of our main results in this thesis. In the real mobile networks, some agents may meet more frequently than others (due to their moving speeds or limited areas), i.e., the probabilities for choosing interacting pairs are no more identical. Under our designed fairness, the inter-contact time<sup>8</sup> of two agents is distributed geometrically, which corresponds to the experimental observation in practical mobile networks. More motivations and details are given in chapter 3.2.

**Cover Time Fairness.** Cover time fairness introduces an idea similar to partial synchrony in [52], in the sense that constraints are imposed on the relative (activation) speeds of agents. An agent interacts with the others with a certain “frequency”, expressed by a *cover time* parameter [26]. This is an upper bound on the “time”, counted in number of global interactions (pairwise meetings), for an agent to interact with all the others. The “faster” an agent meets the others, the smaller its cover time is. As shown by recent experimental and analytical studies [74, 91], such an assumption of bounded cover times may model well the mobility in many practical sensor networks (e.g., where agents have different communication capabilities and move within a bounded area).

Formally, every agent  $i \in \mathbf{A}$  has (an unknown to agents) a parameter  $cv_i \in \mathbf{N}$  called *cover time* s.t. during any  $cv_i$  time units (consecutive interactions in an execution), agent  $i$  interacts with every other agent at least once.  $cv_i$  is the minimum of such a number of interactions.<sup>9</sup> For two agents  $x$  and  $y$ , if  $cv_x < cv_y$ , then we say that  $x$  is *faster* than  $y$ , and  $y$  is *slower* than  $x$ . The minimum cover time value is denoted by  $cv_{min}$  and the maximum one by  $cv_{max}$ . A *fastest* (resp. *slowest*) agent  $z$  has  $cv_z = cv_{min}$  (resp.  $cv_z = cv_{max}$ ). We denote by  $F$  the set of fastest mobile agents, and by  $NF$  the set of non-fastest ones.

It should be noted that cover time fairness enables a worst-case analysis on the global time (the number of interactions until termination) of a protocol [26, 141].

<sup>8</sup>the time period between two successive interactions of the same two mobile agents

<sup>9</sup>Note that there are mappings from agents to their cover times for which the cover time fairness cannot be satisfied (e.g, if  $cv_1 = 4$ ,  $cv_2 = 6$ ,  $cv_3 = 9$  and  $cv_4 = 16$ ). From now on, we assume mappings allowing at least one possible fair execution. For an additional discussion on the validity of the cover time values, refer to [26].

### Population protocols for the task of data collection

**Data Collection.** We assume that each agent, except the base station (BST), owns initially an input value (which is constant during one execution of the protocol). Eventually, every input value has to be delivered to BST, and exactly once (as a multi-set). When this happens, we say that the protocol (its execution) has *converged*. The *convergence time* of a protocol is the maximum length of a possible execution (until convergence). We denote by  $M$  the number of values that a non-BST agent can receive from other agents (on top of its own initial value). When describing or analyzing a considered protocol, the term “to transfer a value” from agent  $x$  to  $y$  means to copy it to  $y$ ’s memory, and erase it from the memory of  $x$ .

There are four existing population protocols under cover time fairness for the task of data collection, studied by Beauquier et al. [26]. The first one, SIMPLE, makes every agent transfer its value directly to the base station. Thus, the convergence time of SIMPLE depends on the speed of the slowest agent, which is  $cv_{max}$ . The second one, STRONG, supposes that every agent  $i$  could check a predicate  $2 \cdot cv_{min} < cv_x$  at the start of the protocol. This predicate gives each agent an idea about its speed compared with others. So, if the predicate is true, agent  $i$  transfers its value to the fastest agent. Otherwise, agent  $i$  waits to meet the base station. The worst case (convergence) time complexity decreases to  $\min\{2cv_{min}, cv_{max}\}$  in the case where the memory is unbounded. However, this protocol requires that every agent knows the value of the cover time of a fastest agent and the value of its own cover time, which is a very strong assumption. Then, the third one, "Transfer To the Faster" (TTF), is proposed, where the agent transfers its values only to a faster agent or the base station. The agents are not assumed to know the cover times, instead, two interacting agents can compare their respective  $cvs$ . Then, the worst case complexity of TTF is  $m \cdot cv_{min} - |F|$ , where  $m$  is the number of different  $cvs$  in the system and  $|F|$  is the number of fastest agents. Last but not least, a time-optimal population protocol, called TTFM (Transfer To the Faster Marked), is introduced. The basic strategy for data transfer in TTFM, is that a mobile agent only transfers its values either to BST, or to a faster agent that has never met (yet) another mobile agent faster than itself. This protocol will be presented in more details in Sect. 2.3. The authors showed that when  $cv_{max} \geq 2cv_{min} - |F|$  and the memory is unbounded, TTFM has optimal worst case (convergence) time complexity (Corollary 18 in [26]).

However, none of these protocols is designed with energy consumption in mind. So, in the first part of this thesis, we first built an energy consumption framework for population protocols. Then we design power-aware population protocols for the task of data collection and study their performances in terms of time and energy.

### 1.2.3 Wireless body area networks

The second model studied in this thesis, is *Wireless Body Area Networks* (WBAN). With the advances of electrical sensors and the development of intelligent health-care, WBAN was designed in 2001 [132], for taking care of human body’s health using wireless biological sensors. So far, WBAN has received lots of attentions from the researchers [79, 133, 135], as it enhances the current health care practices

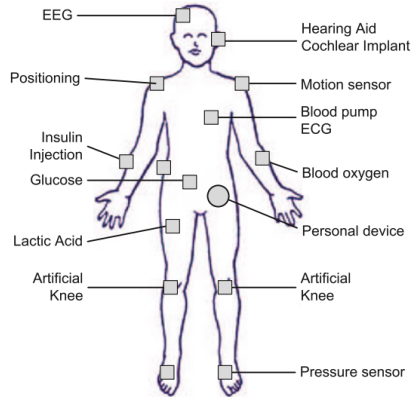


Figure 1.1: Example of distributions of WBAN nodes on a patient [85]

(e.g., MobiHealth [134], CodeBlue[120]) and copes with the high costs of health care system. Besides, it can also be applied for sports training [122] and military operations [75] as well.

A wireless body area network (WBAN) is composed of tiny biological sensors (*bio-sensors*), which are placed in or on the body of a person in order to remotely monitor health-care status such as fever, blood pressure, body temperature, heart rate, blood glucose concentration, among many others. Each intelligent sensor has enough capability to process and forward information to a *base station* for diagnosis and prescription. The base station can be a personal device, such as smart phone, acting as a sink for the data. In addition, there is another kind of nodes in WBAN, called *actuators*, who take actions according to the data received from the bio-sensors or the base station. For a better understanding, we use the example given in [85] to illustrate the roles of nodes in WBAN. In the example of Fig. 1.1, the bio-sensors are placed for measuring heart rate (ECG), brain activities (EEG), blood pressure, glucose level, etc. Next to these sensors, the patient has actuators, such as insulin injection, acting as drug delivery systems. The medicine can be delivered by the predefined program, once the actuator is triggered by an event. The event can be either the order sent by a doctor after analyzing the data or the signal from the bio-sensor when it detects abnormal statistics. For instance, when the sensor of glucose monitors a sudden drop of glucose level, it activates the injection of insulin.

To integrate a WBAN (a patient) into health care system where the patient can be remotely consulted, monitored and treated, a communication between the base station (the patient's personal device) and an external network should be constructed. Take the complete real-time infrastructure for patient's rehabilitation (Fig. 1.2) as an example, it enables the connection between patients and doctors. The health data of patients is transferred and stored in a medical server where doctors can access. In the mean time, a patient can require real time diagnostic recommendations given by doctors. Moreover, when the data of a patient exceeds a predefined threshold, the base station can make an emergency call for the hospital with the

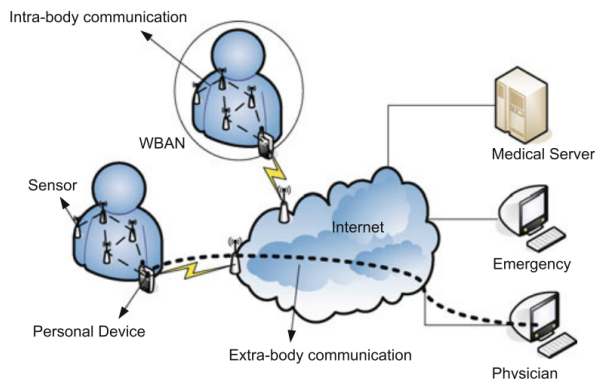


Figure 1.2: Infrastructure for a real health care system [85]

real time monitoring data.

**WBAN vs. Population Protocols.** There are three major differences between WBAN and population protocols. Firstly, WBAN are small sensor networks with the expected number of sensors up to 50 [102] and a maximum number up to 256 defined in IEEE 802.15.6. While population protocols assumes a large population (hundreds, even thousands) and the number of agents is unknown to the network. Secondly, sensors in WBAN are heterogeneous in terms of available energy, computing power and identifiers. So in WBAN, a predefined lower power consumption routing could be scheduled as sensors distinguish their neighbors. However, the agents in population protocols are anonymous and homogeneous. Thirdly, population protocols are asynchronous networks, while WBAN could be synchronous networks under some designs in MAC layer (See Chapter 2.2 in [87]). Finally, in population protocols, agents are supposed to communicate in a pair-wise (peer-2-peer) fashion, while in WBAN they communicate by neighboring broadcast.

For the common parts, the first one is that, for both, communications are in short range of distance. A maximum operating range of 3 meters is required in WBAN [94]. In population protocols, the communication is established only when two agents are close enough. Second, in both designs of data collection protocols, network control packets are limited since both nodes have bounded memory and limited energy resources. In addition, in WBAN, since the propagation of the radio in human body suffers from the fading, noise and interference, the available bandwidth for the communication channel is narrow. All in all, for data collection protocols in both networks, the energy consumption is a necessary and crucial concern.

### Data collection protocols in WBAN

Unlike typical WSN, WBAN suffer from very limited energy resources and hence preserving the energy of the nodes is of great importance. Additionally, an extremely low transmit power per node is required in order to minimize interference and to cope with health concerns such as avoiding tissue heating of skin on patients. One

possible approach to minimize power consumption as well as tissue heating of skin problems is by improving the performance of routing protocols. In WBAN, routing protocols must have self-configuration features and must be capable of finding the best route for communication in order to increase delivery insurance and decrease energy consumption between nodes.

So far, there have been several algorithmic approaches to control bio-effects for WBAN. The authors in [127] propose a Thermal Aware Routing Algorithm (TARA) that balances the communication over the sensor nodes in order to route data away from high temperatures. The algorithm achieves better energy efficiency levels and low temperatures, however it requires that all nodes have complete knowledge about the temperatures of all remaining nodes in the network. Another attempt is a protocol known as Anybody [138]. The underlying idea of Anybody is to form clusters and a backbone network with selected cluster heads in order to reduce the number of direct transmissions to the base station. This algorithm also achieves energy savings, but does not consider other aspects such as reliability of messages for example. In [84], the authors compare and analyze different protocols from WBAN requirements to energy efficiency whereas in [76] the authors propose a weighted random value protocol for multiuser WBANs (WRAP). In [33], authors show that the proper placement of relay nodes can permit a better energy performance for the nodes far away from the base station. Then an integer linear programming model is proposed in [54], which optimizes the number and location of relays to be deployed and the data routing towards the sinks. Other efforts consider explicit mathematical programming formulations in order to efficiently design optimal routing protocols in WBANs ([2],[3]). For more details on recent advances in data collection protocols, readers can refer to the survey [53].

In this thesis, we consider a multi-commodity net-flow problem to optimally route packets in WBAN, which minimizes the worst power consumption and subjects to flow conservation and maximum capacity energy constraints.



# Power-Aware Population Protocols

---

## 2.1 Introduction

In this chapter, we enhance population protocol (PP) with an energy consumption framework. Basically, an agent consumes energy when: (1) interacting and exchanging data, and (2) signaling its presence and sensing the wireless medium continuously (to allow detection of other agents in proximity). An interacting agent is said to be in *awake* mode and spends an amount  $E_{wkp}$  of energy. When it does not interact, but senses the medium, it is in *sleep* mode, and spends  $E_{slp}$  for every global interaction (that can be seen as a time unit). We also consider an additional mode called *terminated*, in which an agent consumes no energy. This mode can allow saving energy of agents having terminated their part in a task. The interest of this extended model is to allow a purely analytical analysis of the energy complexity of a protocol without simulations.

To adapt this general energy model to population protocols, we make some additional considerations and assumptions detailed in Sect. 2.2. For instance, as the data exchanged during an interaction is bounded by the small agent's memory, we can assume that any transmitted data fits in one packet and thus the energy spent in a communication is constant. In addition, as the agents communicate only in pairs when they come close to each other, it is likely to assume that the sensing of the medium is done by proximity sensors (cf. [58, 106]) and thus, agents signal their presence only passively, spending no energy.

In order to illustrate the power and the usefulness of this enhanced model, we consider the issue of determining the amount of agents' initial energy necessary and sufficient for being able to perform a given task. This issue is crucial for choosing, in practice, a category of sensors (in respect with their power capacities) adapted both to the task and to the number of times it should be repeated (before the sensors are replaced or recharged).

In this context, we analyze the maximum amount of energy spent by an agent for performing a one-time collaborative task, of data collection in our case. This metric is clearly related to the lifetime of the network and to the amount of the required initial energy for each agent. The formal analysis of this metric allows, in advance, to adjust the network both in terms of size and battery power allocated to the agents. It is particularly important in the case of networks where it is difficult or undesirable to access the sensors frequently (cf., Bird Species Recognition [29]). Due to the nature of the considered problem, a worst case analysis has to be done. However, it is important to note that, if no guarantees are imposed on the agents' interactions,

such analysis is in general impossible (for non-trivial protocols). For example, in the case of global fairness, protocols converge only eventually, consuming an arbitrary energy till convergence. On the contrary, with random fairness where pairs of agents interact according to some probability distribution, an average complexity analysis of energy consumption can be computed.

Here, for having a bounded convergence and still being able to perform a worst case analysis of the maximum energy spent, we do not adopt the probabilistic approach, but we assume that the enhanced model satisfies cover times fairness (Sect. 1.2.2).

### 2.1.1 Overview and outline

The chapter is organized as follows. The framework allowing an analytical computation of energy consumption is presented in Section 2.2. Using this energy model, we start by analyzing energy consumption of the data collection protocol, TTFM, proposed and proven to be time optimal in [26] (Section 2.3). In this analysis, our approach is motivated by the observation that the energy spent is proportional to the length of an execution. However, further investigation shows that this is not the only factor that can affect the energy spent, and other factors, as the residual energy of an agent, if considered by a solution, can improve energy performance. This gives an idea for developing a better solution in terms of energy consumption, and we propose a power-aware protocol. We prove its correctness and analyze its energy complexity (Section 2.4). Not surprisingly, it appears that it needs less energy than the first protocol when correctly adjusting the protocol's parameters. Having analytical formulas for energy consumption under the form of functions allows to draw graphs and plots, on which remarkable points, zones and limits appear immediately (Sect. 2.4.2 and 2.4.2). This is particularly advantageous when a great number of parameters are involved and when the exhibited functions are hybrid (i.e., composed of several functions). Getting the same or approximative results from simulations would certainly be longer and more costly, while here exact results can be obtained almost immediately. This chapter contains such plots as an illustration of this methodology. Section 2.5 presents lower bounds concerning energy consumption of any possible data collection protocols. Cases where this lower bound is reached by the presented protocols are also given. Finally, Sect. 2.6 is a summary of the results in this chapter.

This work has been published in ICDCS 2017 [141].

### 2.1.2 Related work

There is a very large literature on power-aware network protocols. However, most of the works on energy consumption do not present an analytical study of energy complexity, and the validation of the approaches is made by simulation. In this section, we will only mention the studies directly linked to our approach by restricting our attention to those for which a formal model of energy consumption is presented and its mathematical analysis is done.

In this category, we should first mention [101] where authors study the problem of



energy balancing between non finite state agents (interacting uniformly at random) in a model inspired by population protocols. The considered model is not an energy consumption one. An interacting agent can transfer some part of its energy to the other agent during the interaction (by using a corresponding energy equipment) and there may be a constant energy loss for each unit of the transferred energy. However, contrary to our study, energy consumption for interacting and sensing is not considered. Three protocols, for balancing (averaging) the distribution of energy given to agents, are presented and their analytical and simulation analysis is presented.

Then, one can mention [140], which studies a synchronized scheduling of wake-up times for improving the data collection in a non-mobile sensor network. It proposes protocols with energy consumption of at most twice the optimum. Synchronization of the wake-up times is also studied in [72], where several self-stabilizing protocols to solve the problem of temporal partition are presented and analyzed.

In [146], a 2-approximation (energy consumption) centralized heuristic and a distributed power-aware heuristic are developed for wireless sensor networks with unique identifiers, to ensure that all packets are delivered with the minimum energy cost within a required deadline. Authors develop an evaluation function of energy cost for one transmission, considering residual energy, quality of link and the type of nodes.

In [23], authors propose a randomized clustering algorithm for non-mobile wireless sensor networks which minimizes the energy spent for transmissions. It is proven formally that the algorithm has a time complexity of much less than  $O(n)$ .

In [137], authors give an approximated cone-based topology scheme to increase ad-hoc network lifetime with adaptive transmission power control, while maintaining connectivity for routing. In an analytical way, it is shown that the power consumption of each route can be made arbitrarily close to the optimal.

[44] studies the performance of a class of simple and local algorithms for energy-efficient construction of minimum spanning trees in a wireless ad hoc setting. Bounds on the performance of these algorithms, in instances obtained by uniformly distributed points in the unit square, are given.

For being complete, we should also mention that there exist studies presenting a mathematical model for optimizing the energy-consumption of a single entity, independently of the protocol run, which is one of the main focus of the second part in this thesis, but not in our preoccupations here.

Finally, even if some approaches considering energy consumption are in spirit similar to ours, none of them is devoted to the specific type of networks that we consider: anonymous resource-limited mobile sensors moving unpredictably and constituting a highly dynamic opportunistic network.

### 2.1.3 Assumptions

In the protocol analysis, we consider only the case where  $cv_{BST} > cv_{min}$ , that maximizes time and energy consumption (for data collection). In some cases, for simplicity, we also assume that  $cv_{min} \gg O(n^2)$ . This assumption holds for a large and natural family of cover time vectors. It is especially natural for the case of

passively mobile sensor networks, which is exactly the type of networks modeled by population protocols. Intuitively, as  $cv_{min}$  (in this family) is much larger than the number of all possible pair-wise interactions, agents are free to interact in many different ways, as it would be naturally expected with passive mobility. Still very particular schedules are possible and there are guarantees that agents interact according to the cover times.

We emphasize that agents are not assumed to know cover times (to conform with the finite state population protocol model). Instead, we do assume that two interacting agents can only compare their respective  $cvs$ . This can be implemented by comparing categories instead of  $cvs$ , in applications where the overall number of categories is likely to be uniformly bounded.

## 2.2 Energy Consumption Scheme and Metrics

Up to now, energy does not appear in the literature on population protocols. Although energy is crucial, its introduction in a formal model is difficult, because it involves a lot of parameters. That can be seen in the studies using simulations. In the present work, we realize this introduction. For being realistic, we studied energy saving schemes already developed in wireless sensor network. We summarize them below and discuss their adaptability to population protocols (Sect. 2.2.1). Finally, we propose the design of energy consumption scheme for population protocols (Sect. 2.2.2).

### 2.2.1 Energy saving schemes in wireless sensor networks

Generally speaking, there are two ways to save energy in WSN. One is to recharge directly the battery with outer supply such as magnetic resonance, reflected solar energy. Another way is to apply efficient energy saving schemes. Readers can refer to [13, 48] for a well-organized survey on energy management in WSN.

Figure 2.1 shows the taxonomy of energy saving schemes for WSN. In the sequel, a brief introduction to each kind of schemes is presented. And the reason why *on-demand* schemes (marked in red color in Fig. 2.1) are chosen as the energy framework for population protocols, is followed.

There are three main categories in energy saving schemes: *duty cycling*, *data driven* and *mobility*. Duty cycling enables the nodes to switch between active and sleep periods depending on network activity, and thus to reduce the energy consumption on idle listening. *Duty cycle* is defined as the fraction of time during which nodes are active, during their lifetime. Data driven techniques are designed to reduce the energy consumption on sending redundant or unneeded sensing data. Finally, mobility techniques are used to add mobile nodes to relay data and thus avoid energy depletion on a specific node.

Among duty cycling techniques, *topology control* refers to find the optimal subset of nodes (to be active) that guarantees connectivity of the network. *MAC protocols with low duty cycle* aim to schedule duty cycles such that the corresponding channel accesses have as few collisions as possible. While *Sleep/Wakeup protocols* turn the idle agents into sleep for a certain period of time (on top of the MAC protocols).

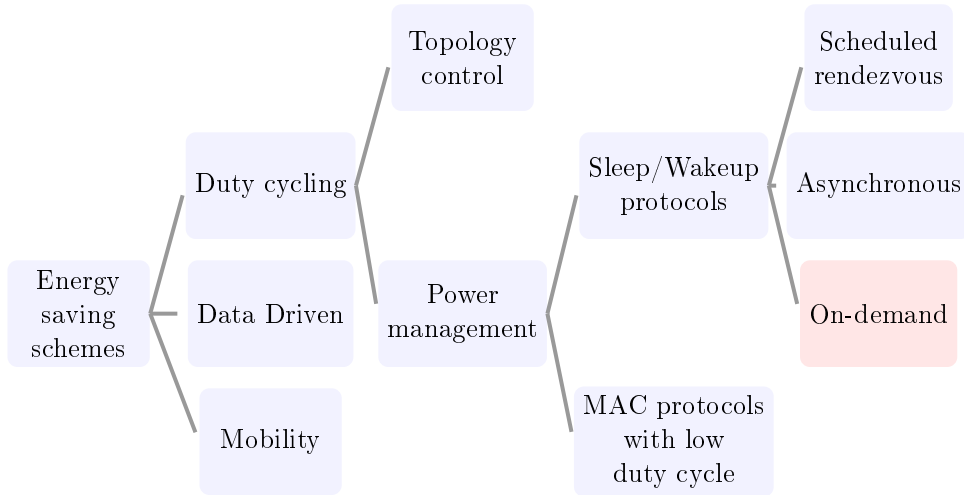


Figure 2.1: The taxonomy of energy saving schemes for WSN [13]

For designing as general as possible energy framework for population protocol, data driven and mobility schemes are not within the scope of the interest. At the same time, as agents have high mobility, which causes frequent changes in the topology of the network, topology control is not applicable neither in the design. The Sleep/Wakeup protocols are investigated furthermore in the following.

**Sleep/Wakeup protocols.** There are three different strategies for deciding how long an agent sleeps and when it wakes up: *scheduled rendezvous*, *on-demand* and *asynchronous* schemes.

Scheduled rendezvous schemes assume that agents agree on a specific moment to communicate (e.g, S-MAC [147], T-MAC [133]). This kind of approach is adopted in ZebraNet [148], where agents are synchronized through a Global Positioning System (GPS). However, it seems hard to add synchronization in population protocols, since it would cost a lot of time and energy.

On-demand schemes consider that a node wakes up only when it needs to communicate. This basic idea is realized by adding a low power stand-by radio for signaling ([69],[119],[108]) or applying radio-triggered techniques ([66],[81]) where sleeping nodes are triggered to be woken by a wakeup radio. All on-demand protocols require a short communication distance in WSN because of the supplementary radios, restricted to a dozen meters in most cases. For instance, Radio Frequency Identification (RFID) systems [65] require a read range less than 6 meters and Near Field Communication (NFC) [45] requires a few centimetres. Since population protocols assume that agents communicate only when they are close to each other, on-demand schemes can be introduced in the framework design (details are given in the next subsection).

Asynchronous schemes (like periodic listening [78], wakeup scheduling [149]) allow each node to wake up independently of the others by guaranteeing that neighbors always have overlapped active periods with duty cycles. This scheme allows

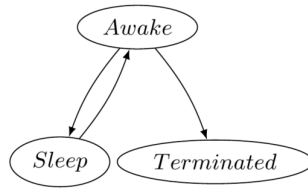


Figure 2.2: Transitions between modes.

a longer communication distance than the on-demand ones, but it is less energy efficient. Since, in population protocols, the communication distance between two agents is supposed to be small, to have the most energy efficient design, we adopt the on-demand schemes in stead of the asynchronous ones.

### 2.2.2 On-demand scheme for population protocols

In the sequel, we present in details our energy consumption scheme for population protocols, which belongs to the category of on-demand schemes as explained above. During an execution, each non-BST agent consumes energy according to three different modes: *sleep*, *awake* and *terminated*. The transitions between modes are depicted in Fig. 2.2 and the overall consumption scheme can be summarized by the following points:

1. Each agent starts with the same amount of initial energy  $E_0$  (e.g., all agents are fully charged). It is in *sleep* mode, and consumes  $E_{slp}$  per time unit (every interaction). During this mode, an agent consumes a reduced amount of energy. It senses the vicinity for detecting other agents in proximity, and signals its presence. The signalling can be done either actively, by emission of small beacon messages (as, e.g., in IEEE 802.11), or passively, only by agent's presence (radio triggered techniques). The latter can be carried out by sensing the vicinity using proximity sensors (cf. [58, 106]). In this work, we consider this latter case of sensing, as it is the most energy efficient one which can be applied to population protocols.
2. When two agents in *sleep* mode have successfully detected each other, the interaction happens and both turn into *awake* mode to proceed with a communication. For that, they have to switch on their radio transceivers, which is generally a high energy consuming operation (cf. [111]). During the interaction, both agents stay *awake* and each consumes  $E_{wkp} = E_{sw} + E_{tx} + E_{rx} + E_{slp}$ , where  $E_{sw}$  is the energy consumed for switching to the main radio,  $E_{tx}$  and  $E_{rx}$  are consumed during the sending and the reception of data, and  $E_{slp}$  is due to point 1.

Since in the considered types of networks (and in PP) non-BST agents have a small memory, independent of the network size, transmitted data can fit in very few, and even one, packets in every communication. Thus, we can assume that the energy consumed by the communication,  $E_{tx} + E_{rx}$ , is constant.

3. During an interaction, two *awake* agents decide to turn either to *sleep* or to *terminated* mode, according to the protocol. No energy is consumed by a *terminated* agent. This mode is useful for saving energy of agents that have terminated their protocol activities. Together with that, the scheme can be adapted in such a way that a *terminated* agent could be restored into *sleep* mode, whenever it is required to restart a task. This can be done, for example, by a global signal from BST, which, at the same time, can also recharge the mobile agents, to prepare them for the next task [126].<sup>1</sup>
4. We assume that when a sleeping agent  $x$  meets a *terminated* agent  $y$ ,  $x$  wakes up and thus, spends  $E_{wkp}$ . This assumption makes sense in networks deploying proximity sensors for sensing close agents. In this case, the *terminated* agent  $y$  is detected by the proximity sensor, and as  $x$  is not aware of  $y$ 's mode in this step,  $x$  wakes up and only then detects that  $y$  is *terminated*.<sup>2</sup>

**Evaluation Metrics.** There are many different power-aware metrics in the related literature on sensor networks. The term lifetime of a network is generally used and can be evaluated, e.g., as the maximum time until one or all sensors have no energy, or until there are more than a ratio  $\beta$  of surviving nodes, or until "connectivity" is lost. Alternatively, to analyze the lifetime, one can evaluate the maximum energy spent per node for a given task (e.g., per data collection from all nodes to BST, or per packet routing to the next node, or to the final destination). Refer to surveys on such metrics [42, 121].

In this work, we choose to evaluate and minimize the maximum energy spent per node across the network in each data collection, thereby enhancing the lifetime performance. Given a protocol  $\mathcal{P}$ , let  $\mathcal{E}(\mathcal{P})$  be the set of all executions in  $\mathcal{P}$  and  $E_{smax}(e)$  the maximum energy spent by an agent at the end of the execution  $e \in \mathcal{E}(\mathcal{P})$ . Then we denote by  $E_{smax}(\mathcal{P})$  the maximum energy spent by an agent in an execution of  $\mathcal{P}$ , i.e.,  $E_{smax}(\mathcal{P}) = \max_{e \in \mathcal{E}(\mathcal{P})} E_{smax}(e)$ . This metric is one of the most popular metrics studied in power-aware routing protocols for MANET [121]. Minimizing it can be also seen as balancing the distribution of energy consumption among agents. Moreover, as already mentioned in the introduction (Sect. 2.1), evaluation of this metric can allow, in advance, to adjust the network both in terms of size and battery power allocated to the agents.

## 2.3 Energy Consumption of TTFM

Our starting point is TTFM [26], which was designed without energy consumption in mind, and we perform its energy consumption analysis, using the proposed energy model.

The basic strategy for data transfer in TTFM (Protocol 1), is that a mobile agent only transfers its values either to base station, or to an agent that has never

<sup>1</sup>In this work, we do not consider this possibility, as we analyze energy consumption for only one data collection.

<sup>2</sup>However, with a beacon based approach for agents' detection (described in the first point), neither  $x$ , nor  $y$  spend  $E_{wkp}$  in such an interaction.

**Protocol 1** TTFM

---

(for a mobile agent  $i$  interacting with an agent  $j$ )

---

- 1:  $fastest\_mark_i \in \{0,1\}$  (\* initialized to 1 \*)
  - 2: **if** ( $j$  is not base station) **then**
  - 3:   **if** ( $cv_j < cv_i$  **and**  $fastest\_mark_j := 1$ ) **then**
  - 4:      $fastest\_mark_i := 0$
  - 5:   **if** ( $fastest\_mark_j > fastest\_mark_i$  **and** there is a free place in  $j$ 's memory) **then**
  - 6:     agent  $i$  transfers to  $j$  as many values as would fit in the free part of  $j$ 's memory
  - 7: **if** ( $j$  is base station) **then**
  - 8:   agent  $i$  transfers all values to BST
- 

met (yet) another mobile agent faster than itself. When two fastest agents meet, there is no transfer of data. The idea of TTFM is to make use of the fastest agents, which are more likely to bring sooner the values to base station. In protocol 1,  $fastest\_mark_i$ , initialized to one, is a bit mark to record if agent  $i$  has already met a faster agent. Lines 3-4 are to affect the value of  $fastest\_mark_i$  during the meetings of agents. Lines 5-6 ensure that agent  $i$  transfers its values only to an agent that has never met (yet) another faster agent.

For illustrating the usage of *terminated* mode, we adapt TTFM in such a way that a mobile agent turns into *terminated* mode once it has transferred all of its values to a faster agent (excluding BST). Indeed, according to TTFM, once this happens, the slower agent does not receive any value till the end of the execution. Therefore, turning into *terminated* mode saves energy. For the fastest agents, as they never transfer values to other agents except to BST, they stay *non-terminated* until convergence. This adapted protocol, called E-TTFM, is presented below (Protocol 2).

**Protocol 2** E-TTFM

---

(for a mobile agent  $i$  interacting with an agent  $j$ )

---

- 1:  $fastest\_mark_i \in \{0,1\}$  (\* initialized to 1 \*)
  - 2: **if** ( $j$  is not BST) **then**
  - 3:   **if** ( $cv_j < cv_i$  **and**  $fastest\_mark_j := 1$ ) **then**
  - 4:      $fastest\_mark_i := 0$
  - 5:   **if** ( $fastest\_mark_j > fastest\_mark_i$  **and** there is a free place in  $j$ 's memory) **then**
  - 6:     agent  $i$  transfers to  $j$  as many values as would fit in the free part of  $j$ 's memory
  - 7:     **if** (agent  $i$  has no value in its memory) **then**
  - 8:       agent  $i$  turns into *terminated* mode
  - 9: **if** ( $j$  is BST) **then**
  - 10:   agent  $i$  transfers all values to BST
- 

E-TTFM has the same time complexity as TTFM, because the adaption we did

does not influence the worst case convergence time of data collection. Therefore, the longest execution in E-TTFM is of length  $2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1$  (Th. 11 in [26]). The worst case on the maximum energy spent by an agent is reached in the longest execution, in which one fastest agent participates in a maximum number of interactions (being in *awake* mode as long as possible). This implies the result:

**Proposition 1.** *An upper bound on the maximum energy spent by an agent in E-TTFM is  $(2cv_{min} \lceil \frac{|NF|}{M \times |F|} \rceil - 1)E_{wkp}$ . This bound is reached when  $cv_{min} \gg O(n^2)$ .*

*Proof.* The upper bound is easy to obtain, as the longest execution in E-TTFM is of length  $(2cv_{min} \lceil \frac{|NF|}{M \times |F|} \rceil - 1)$  [26]. Thus, no agent can consume more energy than  $(2cv_{min} \lceil \frac{|NF|}{M \times |F|} \rceil - 1)E_{wkp}$ .

Next, we prove that this bound can be reached when  $cv_{min} \gg O(n^2)$ . Consider the following schedule which repeats the segment  $X_1X_2$ , where  $X_1X_2$  is of length  $cv_{min}$ . We denote by  $f$  one of the fastest agents.

$X_1 \equiv$  [ repeat as much as possible all interactions between  $f$  and every agent in  $A \setminus \text{BST}$  ],

$X_2 \equiv$  [ all the other necessary interactions to satisfy  $cv$ s constraints ].

Consider the execution  $e$  of E-TTFM corresponding to this schedule. This execution reaches the upper bound on time presented in [26] and then  $|e| = (2cv_{min} \lceil \frac{|NF|}{M \times |F|} \rceil - 1)$ .

In E-TTFM, a fastest agent never turns into *terminated* mode. Therefore in  $e$ , it consumes the most ( $E_{wkp}$  in each interaction in  $X_1$ ). Thus,  $Es_{max}(e) = (|e| - \epsilon|X_2|)E_{wkp} + \epsilon|X_2|E_{slp}$  where  $\epsilon|X_2|$  is the number of (necessary) interactions in  $e$  not involving agent  $f$ .

Notice that  $|X_2| \leq \frac{n(n-1)}{2} = O(n^2)$  and as  $cv_{min} \gg O(n^2)$ , the length of  $X_2$  is negligible compared with  $|e|$ . Therefore,

$$Es_{max}(\text{E-TTFM}) = Es_{max}(e) = |e|E_{wkp} = (2cv_{min} \lceil \frac{|NF|}{M \times |F|} \rceil - 1)E_{wkp}.$$

□

## 2.4 Energy-Balanced Protocol EB-TTFM

A first step towards a protocol balancing the distribution of energy consumption would be to consider the following strategy: When an agent notices that it becomes "low" in energy, it tries to transfer its values to a higher energy level agent and turns into *terminated* mode, as soon as possible. The issue raised by this strategy is that the convergence time for data collection could be possibly considerably augmented, especially if some values are transferred to a slow agent. An increased convergence time means more energy spent.

Therefore, the level of the amount of residual energy that is considered to be "low" should be carefully set. Below, we discuss the possible choices of such a level. We make these choices depend on a parameter  $\lambda$ , by defining the "low" level by  $\frac{E_0}{\lambda}$  where  $\lambda \geq 1$ . To conform with the finite state population protocol model, we assume that

$\frac{E_0}{\lambda}$  can only take a finite number of values. In the same way as for cover times, we assume that the agents do not know their residual energies but can compare them when they interact. Adding the test of the energy level to E-TTFM yields the protocol EB-TTFM( $\lambda$ ) (for Energy Balanced TTFM, Protocol 3). In the code below,  $el_i$  denotes the residual energy of the mobile agent  $i$ .

---

**Protocol 3** EB-TTFM( $\lambda$ )(for a mobile agent  $i$  interacting with an agent  $j$ )

---

- 1: **if** ( $j$  is not BST) **then**
  - 2:     **if** ( $el_i < \frac{E_0}{\lambda}$  **and**  $el_i < el_j$  **and** there is a free place in  $j$ 's memory) **then**
  - 3:         agent  $i$  transfers to  $j$  as many values as would fit in the free part of  $j$ 's memory
  - 4:     **if** (agent  $i$  has no value in its memory) **then**
  - 5:         agent  $i$  turns into *terminated* mode
  - 6:     **else**
  - 7:         execute E-TTFM
  - 8: **if** ( $j$  is BST) **then**
  - 9:     agent  $i$  transfers all values to BST
- 

### 2.4.1 Energy Consumption Analysis

Intuitively, by observing the code of EB-TTFM( $\lambda$ ), it can be seen that, if the threshold  $\frac{E_0}{\lambda}$  is too high, many interactions result in the execution of lines 3-5, since the condition in line 2 is satisfied more easily. That concerns especially the fastest agents, since they interact more frequently. As a consequence, all the fastest agents switch quickly to *terminated* mode and the values are carried to BST by slower agents, even possibly the slowest agent. Then worst case convergence time and energy consumption depend on  $cv_{max}$  instead of  $cv_{min}$ , which is worse than for E-TTFM (Prop. 1).

On the other hand, if the threshold  $\frac{E_0}{\lambda}$  is too small, EB-TTFM( $\lambda$ ) executes line 7 more often, i.e., behaves like E-TTFM, since the condition in line 2 is difficult to satisfy. Thus, in the perspective of improving the performance of EB-TTFM, we first study the conditions excluding the two previous (uninteresting) cases. More precisely, we study the corresponding upper and lower bounds for  $E_0$  with respect to  $\lambda$  (Lem. 1 and 2). Then, for  $E_0$  and  $\lambda$  given inside these bounds, we study the worst-case performance of EB-TTFM( $\lambda$ ) in terms of maximum energy spent by an agent (Th. 1). Next, we try to figure out, for a given  $E_0$ , the best choice for  $\lambda$  (Sect. 2.4.2), according to the results for the worst-case. Then, plots are drawn to show the performance of EB-TTFM( $\lambda$ ) (Sect. 2.4.2 and 2.4.2).

As explained above, the number of fastest agents turning into *terminated* mode during an execution plays an important role in the energy performance of EB-TTFM( $\lambda$ ). Therefore, in the sequel, we classify the executions by the number of fastest agents in *terminated* mode at the end of the execution. Then, we study the length of the longest execution in each classification set (Prop. 2, 3 and 4), we use these results to obtain bounds for  $E_0$  (Lem. 1 and 2) and we determine the worst-case performance



of EB-TTFM( $\lambda$ ) (Th. 1).

**Notations.** We denote by  $\Xi_d$  the set of executions of EB-TTFM( $\lambda$ ) with exactly  $d$  fastest agents in *terminated* mode at their ends ( $d \in \{0, 1, \dots, |F|\}$ ). For a non-empty set  $\Xi_d$ ,  $mni(d)$  is the minimum number of interactions until  $d$  fastest agents are in *terminated* mode in the executions of  $\Xi_d$ , and  $mnu(d)$  is the minimum number of values that have been delivered to BST at the  $mni(d)^{th}$  interaction. We denote by  $e_d^{max}$  a longest execution in  $\Xi_d$  and by  $\theta$  the ratio between  $E_{slp}$  and  $E_{wkp}$  ( $\theta = E_{slp}/E_{wkp} < 1$ ).

If  $\cup_{i=1}^{|F|} \Xi_i = \emptyset$ , no fastest agent ever turns into *terminated* mode and fastest agents only execute E-TTFM (line 7). In this case, EB-TTFM( $\lambda$ ) has the same performance as E-TTFM. On the contrary, if  $\Xi_{|F|} \neq \emptyset$ , there exists at least one execution, at the end of which all fastest agents are in *terminated* mode. Then, some values could be delivered to BST by a slow agent, increasing the convergence time of EB-TTFM( $\lambda$ ) (to be dependent on  $cv_{max}$ ). The case  $\cup_{i=1}^{|F|} \Xi_i = \emptyset$  is uninteresting and the case  $\Xi_{|F|} \neq \emptyset$  is undesirable. That is why we are looking now for conditions on  $E_0$  and  $\lambda$  ensuring that  $\cup_{i=1}^{|F|} \Xi_i \neq \emptyset$  and  $\Xi_{|F|} = \emptyset$ .

Prop. 2 presents a lower bound on  $mni(d)$  for any non-empty set  $\Xi_d$ , denoted by  $l(d)$ . This bound is tight when  $cv_{min} \gg O(n^2)$ . Therefore, to ensure  $\cup_{i=1}^{|F|} \Xi_i \neq \emptyset$ ,  $\min_{d \neq 0} l(d)$  must be smaller than or equal to the length of the longest execution in E-TTFM (see proof of Lem. 1). To ensure  $\Xi_{|F|} = \emptyset$ ,  $l(|F|)$  must be greater than the length of the longest execution in the set  $\cup_{i=0}^{|F|-1} \Xi_i$ , i.e., the set of executions converging before all the fastest agents are in *terminated* mode (see proof of Lem. 2). In addition, in order to obtain an upper bound on the length of the longest execution in Prop. 4, we compute the values of  $mnu(d)$  (Prop. 3).

**Proposition 2. [ $mni(d)$ ].** *A lower bound on the minimum number of interactions before  $d$  fastest agents are in terminated mode, taken over all executions in a non-empty set  $\Xi_d$  of EB-TTFM( $\lambda$ ), is  $l(d) = \left\lceil (E_0 - \frac{E_0}{\lambda}) [1 - (1 - \theta)^{\lceil \frac{d}{2} \rceil}] / E_{slp} \right\rceil$ . This bound is reached when  $cv_{min} \gg O(n^2)$ .*

*Proof.* Consider an execution and a fastest agent  $i$ . In EB-TTFM( $\lambda$ ), agent  $i$  turns into *terminated* mode when the condition (in line 2) is satisfied, which means that its residual energy is less than  $\frac{E_0}{\lambda}$ . During each interaction, agent  $i$  cannot spend more energy than  $E_{wkp}$ , corresponding to the cost of the *awake* mode. Then the number of interactions until agent  $i$  turns into *terminated* mode is at least  $\lceil \frac{E_0 - \frac{E_0}{\lambda}}{E_{wkp}} \rceil$ . As both participants in one interaction consume energy,  $mni(2) = mni(1)$ ,  $mni(4) = mni(3)$ , ..., and more generally  $mni(2m) = mni(2m - 1)$ . In addition, we have the following recursive equations when  $d > 2$ :  $(mni(d) - mni(d - 2)) \times (E_{wkp} - E_{slp}) + mni(d) \times E_{slp} \geq E_0 - \frac{E_0}{\lambda}$ . Recursively, we obtain the following inequality:

$$mni(d) \geq \left\lceil \frac{E_0 - \frac{E_0}{\lambda}}{E_{slp}} [1 - (1 - \theta)^{\lceil \frac{d}{2} \rceil}] \right\rceil = l(d). \quad (2.1)$$

For the same reason as in the proof of Prop. 1, when  $cv_{min} \gg O(n^2)$ , the number

of necessary interactions not involving these  $d$  fastest agents is negligible in every  $cv_{min}$  interactions. In this case, the lower bound on  $mni(d)$  is reached.  $\square$

**Proposition 3. [m $nv(d)$ ].** *When  $cv_{min} \gg O(n^2)$ , the minimum number of values that have been delivered to BST at the  $mni(d)^{th}$  interaction, taken over all executions in a non-empty set  $\Xi_d$  of EB-TTFM( $\lambda$ ), is*

$$mnv(d) = M(|F| - 2\beta) \times \lfloor \frac{mni(d)}{2cv_{min}} \rfloor + 2M \times \lfloor \frac{mni(1)}{2cv_{min}} \rfloor \times [1 + \frac{\beta}{\theta} - \frac{1-\theta}{\theta^2}(1 - (1-\theta)^\beta)],$$

where  $\beta = \lfloor (d-1)/2 \rfloor$ .

*Proof.* It has been shown that in any segment of  $2cv_{min}$  consecutive interactions of an execution of TTFM, at least  $M|F|$  values (tight bound) are transferred to the fastest agents (Th. 11 of [26]). Similarly, it is easy to see that every  $2cv_{min}$  interactions, at least  $M|F|$  values (tight bound) are delivered to BST. However, in EB-TTFM( $\lambda$ ), since the fastest agents can turn into *terminated* mode (line 5), this property is no longer true. But it still holds for the segments of  $2cv_{min}$  consecutive interactions where all fastest agents are *non-terminated*. According to the definition of  $mni(d)$ , before the  $mni(1)^{th}$  interaction, all fastest agents are surely *non-terminated* in any execution of EB-TTFM( $\lambda$ ). So, we obtain the minimum number of values that have been collected at the  $mni(1)^{th}$  interaction in EB-TTFM( $\lambda$ ), which is:  $mnv(1) = M|F| \times \lfloor \frac{mni(1)}{2cv_{min}} \rfloor$ . As  $mni(1) = mni(2)$ , we have  $mnv(1) = mnv(2)$ .

Then, we consider the segment  $[mni(1), mni(3))$  in an execution. There are at least  $|F| - 2$  fastest agents which are *non-terminated* in this segment. Then at least  $M(|F| - 2) \times (\lfloor \frac{mni(3)}{2cv_{min}} \rfloor - \lfloor \frac{mni(1)}{2cv_{min}} \rfloor)$  values are delivered to BST in this segment. Recursively, we obtain the following relation between  $mnv(d)$  and  $mnv(d-2)$ :

$$\begin{aligned} mnv(3) &= mnv(4) \\ &= mnv(1) + M(|F| - 2) \left( \lfloor \frac{mni(3)}{2cv_{min}} \rfloor - \lfloor \frac{mni(1)}{2cv_{min}} \rfloor \right) \\ &= M(|F| - 2) \times \lfloor \frac{mni(3)}{2cv_{min}} \rfloor + 2M \times \lfloor \frac{mni(1)}{2cv_{min}} \rfloor ; \\ mnv(5) &= mnv(6) \\ &= mnv(3) + M(|F| - 4) \left( \lfloor \frac{mni(5)}{2cv_{min}} \rfloor - \lfloor \frac{mni(3)}{2cv_{min}} \rfloor \right) \\ &= M(|F| - 4) \times \lfloor \frac{mni(5)}{2cv_{min}} \rfloor + 2M \times (\lfloor \frac{mni(1)}{2cv_{min}} \rfloor + \lfloor \frac{mni(3)}{2cv_{min}} \rfloor) ; \\ &\vdots \\ mnv(d) &= mnv(d-2) + M(|F| - 2\lfloor \frac{d-1}{2} \rfloor) \times \left( \lfloor \frac{mni(d)}{2cv_{min}} \rfloor - \lfloor \frac{mni(d-2)}{2cv_{min}} \rfloor \right) \\ &= M(|F| - 2\lfloor \frac{d-1}{2} \rfloor) \times \lfloor \frac{mni(d)}{2cv_{min}} \rfloor + 2M \sum_{j=1}^{\lfloor \frac{d-1}{2} \rfloor} \lfloor \frac{mni(2j-1)}{2cv_{min}} \rfloor . \end{aligned}$$

Let  $\beta = \lfloor \frac{d-1}{2} \rfloor$ . From Prop. 2, we have, when  $cv_{min} \gg O(n^2)$ :

$$mnu(d) = M(|F| - 2\beta) \times \lfloor \frac{mni(d)}{2cv_{min}} \rfloor + 2M \times \lfloor \frac{mni(1)}{2cv_{min}} \rfloor \times [1 + \frac{\beta}{\theta} - \frac{1-\theta}{\theta^2}(1 - (1-\theta)^\beta)].$$

□

**Proposition 4.**  $[|e_d^{max}|]$ .

When  $cv_{min} \gg O(n^2)$  and  $d < |F|$ , the length of the longest execution  $e_d^{max}$  in a non-empty set  $\Xi_d$ , is at most  $\rho(d) =$

$$\lceil \frac{|NF| - 2M \lfloor \frac{mni(1)}{2cv_{min}} \rfloor [1 + \frac{1}{\theta} \lfloor \frac{d}{2} \rfloor - \frac{1-\theta}{\theta^2}(1 - (1-\theta)^{\lfloor d/2 \rfloor})]}{M(|F| - d)} \rceil \times 2cv_{min}. \quad (2.2)$$

*Proof.* A longest execution  $e_d^{max}$  in  $\Xi_d$  is obtained by turning  $d$  fastest agents into *terminated* mode as fast as possible, i.e., at the  $mni(d)^{th}$  interaction. Moreover, the data is delivered to BST in the slowest way. It means that there are  $|NF| - mnu(d)$  values not delivered to BST at the  $mni(d)^{th}$  interaction of  $e_d^{max}$ . And from the  $mni(d)^{th}$  interaction until termination, only  $M(|F| - d)$  values are delivered to BST during every  $2cv_{min}$  consecutive interactions. Thus, the longest execution is of length:

$$|e_d^{max}| = mni(d) + \lceil \frac{|NF| - mnu(d)}{M \times (|F| - d)} \rceil \times 2cv_{min}.$$

Therefore, when  $cv_{min} \gg O(n^2)$ , according to Prop. 2 and Prop. 3, we have:

$$\begin{aligned} |e_d^{max}| &\leq \lceil \frac{|NF| - 2M \lfloor \frac{mni(1)}{2cv_{min}} \rfloor [1 + \frac{1}{\theta} \lfloor \frac{d}{2} \rfloor - \frac{1-\theta}{\theta^2}(1 - (1-\theta)^{\lfloor d/2 \rfloor})]}{M(|F| - d)} \rceil \\ &\times 2cv_{min} = \rho(d). \end{aligned}$$

It appears that  $\rho(d)$  is an increasing function of  $d$ . Therefore, the longest execution in  $\cup_{i=0}^{|F|-1} \Xi_i$  is of length at most  $\rho(|F| - 1)$ . □

Using the results above, to exclude the two previously described uninteresting cases for energy analysis, we compute the corresponding lower and upper bounds on the initial energy  $E_0$  in EB-TTFM( $\lambda$ ).

**Lemma 1.** [Upper Bound on  $E_0$  ( $\cup_{i=1}^{|F|} \Xi_i \neq \emptyset$ )].

When  $E_0 \leq \left(2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1\right) \frac{\lambda}{\lambda-1} E_{wkp}$  and  $cv_{min} \gg O(n^2)$ , there exists at least one execution of EB-TTFM( $\lambda$ ), in which a fastest agent turns into *terminated* mode.

*Proof.* We know that the time complexity of E-TTFM is  $2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1$ , when  $cv_{min} \gg O(n^2)$  (Prop. 1). In addition, when  $cv_{min} \gg O(n^2)$ , for a non-empty set  $\Xi_d$ ,  $mni(d) = l(d)$ .

Therefore, to ensure that  $\cup_{i=1}^{|F|} \Xi_i \neq \emptyset$ , i.e., there exists at least one execution of EB-TTFM( $\lambda$ ) in which a fastest agent  $i$  turns into *terminated* mode,  $\min_{d \neq 0} l(d)$

must be less than or equal to  $2cv_{min}\lceil\frac{|NF|}{M|F|}\rceil - 1$ . Elsewhere, EB-TTFM( $\lambda$ ) would behave as E-TTFM. Thus, according to (2.1), we have:

$$\begin{aligned} \min_{d \neq 0} l(d) &= l(1) \leq \left(2cv_{min}\lceil\frac{|NF|}{M|F|}\rceil - 1\right) \\ \Rightarrow E_0 &\leq \left(2cv_{min}\lceil\frac{|NF|}{M|F|}\rceil - 1\right) \frac{\lambda}{\lambda - 1} E_{wkp} . \end{aligned}$$

□

**Lemma 2. [Lower Bound on  $E_0$  ( $\Xi_{|F|} = \emptyset$ )].**

When  $cv_{min} \gg O(n^2)$ ,  $\theta \leq 1/2$  and  $E_0 \geq \lceil\frac{|NF|}{M \times |F|}\rceil \frac{\lambda}{\lambda - 1} cv_{min} E_{wkp}$ , in any execution of EB-TTFM( $\lambda$ ), there is at least one fastest agent, which is not in terminated mode at the end of the execution. <sup>3</sup>

*Proof.* We know from the proof of Prop. 4 that, when  $cv_{min} \gg O(n^2)$ , the longest execution in  $\cup_{i=0}^{|F|-1} \Xi_i$  is of length  $\rho(|F| - 1)$ .

According to the definition of  $mni(d)$ , the configuration where all fastest agents are in *terminated* mode, can appear only at or after the  $mni(|F|)^{th}$  interaction. Then, if all executions in  $\cup_{i=0}^{|F|-1} \Xi_i$  are of length less than the lower bound of  $mni(|F|)$ , no execution reaches such a configuration, i.e., there is no execution  $e$  such that all fastest agents are in *terminated* mode at the end of  $e$ . Thus,  $\rho(|F| - 1) < l(|F|)$  implies  $\Xi_{|F|} = \emptyset$ .

When  $cv_{min} \gg O(n^2)$ , substituting (2.1) and (2.2) in  $\rho(|F| - 1) < l(|F|)$ , we obtain a lower bound of  $E_0$ :

$$E_0 > \lceil\frac{|NF|}{M} \times \frac{\theta^2}{\theta^2 + (\lceil\frac{|F|+1}{2}\rceil + 1/2)\theta - 1}\rceil \frac{\lambda}{\lambda - 1} cv_{min} E_{wkp}.$$

This lower bound can also be seen as a function of  $\theta$ , denoted by  $lb(\theta)$ .

Since  $lb(\theta)$  is an increasing function of  $\theta$  and  $\theta \leq \frac{1}{2}$ , we have  $lb(\theta) \leq lb(\frac{1}{2}) \leq \lceil\frac{|NF|}{M} \times \frac{1}{|F|}\rceil \frac{\lambda}{\lambda - 1} cv_{min} E_{wkp}$ . So, we have a sufficient condition for  $\Xi_{|F|} = \emptyset$ , which is:

$$E_0 \geq \lceil\frac{|NF|}{M \times |F|}\rceil \frac{\lambda}{\lambda - 1} cv_{min} E_{wkp} . \quad \square$$

Next, we consider the worst-case performance of EB-TTFM( $\lambda$ ), supposing that every agent has an initial energy  $E_0$  inside the bounds established above (Th. 1) and outside the bounds (Th. 2 and 3). Then from the analysis of the worst-case performance on energy consumption, we will determine the best choice for  $\lambda$  in EB-TTFM.

Notice that if there is only one fastest agent ( $|F| = 1$ ), the conditions of Lem. 1 and 2 are incompatible. This case is not considered by Th. 1. However, with the conditions of Lem. 2, this fastest agent stays *non-terminated* during all the executions and executes only E-TTFM. In this case, the upper bound in Prop. 1 (depending on  $cv_{min}$ ) holds. Otherwise, the bound is worse, as it depends on a cover time greater than  $cv_{min}$ .

<sup>3</sup>Recall that  $E_{wkp} > E_{slp}$ . For simplicity and following the study in [111], we choose  $\theta \leq 1/2$ .

**Theorem 1. [Worst Case Energy Analysis I].** Assume  $\lceil \frac{|NF|}{M \times |F|} \rceil \frac{\lambda}{\lambda-1} cv_{min} E_{wkp} \leq E_0 \leq (2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1) \lambda E_{wkp} / (\lambda - 1)$ ,  $cv_{min} \gg O(n^2)$  and  $|F| > 1$ . Then

$$Es_{max}(\text{EB-TTFM}(\lambda)) \leq \frac{\lambda-1}{\lambda} E_0 + (\rho(|F|) - 1) E_{wkp},$$

where  $l(d)$ ,  $\rho(d)$  are the functions defined in (2.1) and (2.2), respectively.

*Proof.* Consider an execution  $e$  of EB-TTFM( $\lambda$ ). As  $E_0 \geq \lceil \frac{|NF|}{M \times |F|} \rceil \frac{\lambda}{\lambda-1} cv_{min} E_{wkp}$ ,  $\Xi_{|F|} = \emptyset$  (Lem. 2). Moreover, for a non-empty set  $\Xi_d$ ,  $|e_d^{max}| \leq \rho(d)$  when  $cv_{min} \gg O(n^2)$  (Prop. 4). Therefore, since  $\rho(d)$  is an increasing function of  $d$ , we have  $|e| \leq \max_{d < |F|} |e_d^{max}| \leq \rho(|F| - 1)$ . Let  $i \in F$  be the agent which consumes the most in  $e$ . There are two possibilities at the end of  $e$ :

1. The energy spent by agent  $i$  is smaller or equal to  $E_0 - \frac{E_0}{\lambda}$ . Thus, the maximum energy spent by one agent in  $e$  is smaller or equal to  $\frac{\lambda-1}{\lambda} E_0$ .
2. The energy spent by agent  $i$  is greater than  $E_0 - \frac{E_0}{\lambda}$ . Let  $t_i$  be the number of past interactions when the residual energy of agent  $i$  reaches  $\frac{E_0}{\lambda}$ . As when  $cv_{min} \gg O(n^2)$ ,  $l(1)$  is the minimum number of interactions when a fastest agent turns into *terminated* mode (Prop. 2), we have  $t_i \geq l(1)$ . Thus, the length of the interval  $[t_i, |e|]$  is  $|e| - t_i \leq |e| - l(1)$ . Consequently, the maximum energy spent by an agent in  $e$  is smaller or equal to  $E_0 - \frac{E_0}{\lambda} + (|e| - t_i) E_{wkp} \leq \frac{\lambda-1}{\lambda} E_0 + (|e| - l(1)) E_{wkp}$ .

In conclusion, when  $cv_{min} \gg O(n^2)$ , the maximum energy spent by an agent in the worst case of EB-TTFM( $\lambda$ ),  $Es_{max}(\text{EB-TTFM}(\lambda))$ , is at most  $\frac{\lambda-1}{\lambda} E_0 + (\rho(|F|) - 1) E_{wkp}$ .  $\square$

**Theorem 2. [Worst Case Energy Analysis II].** Assume  $E_0 > (2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1) \lambda E_{wkp} / (\lambda - 1)$  and  $cv_{min} \gg O(n^2)$ . Then  $Es_{max}(\text{EB-TTFM}(\lambda)) = Es_{max}(\text{E-TTFM})$ .

*Proof.* We know from the proof of Lemma. 1 that, when  $cv_{min} \gg O(n^2)$  and  $E_0 > (2cv_{min} \lceil \frac{|NF|}{M|F|} \rceil - 1) \lambda E_{wkp} / (\lambda - 1)$ , EB-TTFM( $\lambda$ ) behaves like E-TTFM. So we have  $Es_{max}(\text{EB-TTFM}(\lambda)) = Es_{max}(\text{E-TTFM})$ .  $\square$

**Theorem 3. [Worst Case Energy Analysis III].** Assume  $E_0 < \lceil \frac{|NF|}{M \times |F|} \rceil \frac{\lambda}{\lambda-1} cv_{min} E_{wkp}$  and  $cv_{min} \gg O(n^2)$ . Then  $Es_{max}(\text{EB-TTFM}(\lambda)) = \Omega(cv_{max} E_{wkp})$ .

*Proof.* According to Lemma. 2, when  $cv_{min} \gg O(n^2)$  and  $E_0 < \lceil \frac{|NF|}{M \times |F|} \rceil \frac{\lambda}{\lambda-1} cv_{min} E_{wkp}$ , there exists one execution  $e = X_1 X_2$ , composed of two segments  $X_1$  and  $X_2$ , where at the end of  $X_1$ , all fastest agents are in *terminated* mode and some values are transferred to the slowest agent. Then we can construct another execution  $e' = X_1 X_3 X_4$ . In  $X_3$ , the slowest agent meets as much as possible the *terminated* agents. When  $cv_{min} \gg O(n^2)$ ,  $|X_3| \approx cv_{max}$ . Therefore, we know that the maximum energy spent by an agent in  $e'$  is at least  $cv_{max} E_{wkp}$ .  $\square$

A natural issue is to determine the best value for  $\lambda$ , that is the value that minimizes the upper bound in Th. 1. This determination is not easy because of the ceiling and floor functions appearing in Th. 1. Thus, in order to get an exploitable expression, we drop, in the next subsection, the assumption of bounded memory and we consider that an agent can hold an arbitrary number of values. Then we obtain a better upper bound using such a  $\lambda$ .

### 2.4.2 Special Case of Non-Bounded Memory ( $M \geq n - 1$ )

The following improved bound for  $E_0$  (with respect to Lem. 2) is obtained when the lower bound of  $\text{mni}(|F|)$  is larger or equal to  $cv_{\min}$ . This condition ensures that at least one fastest agent is not *terminated* during the first  $cv_{\min}$  interactions (and not until convergence as in Lem. 2). During this period, the fastest agents meet all the non-fastest ones, collect all their values (as  $M$  is large enough), and turn them into *terminated* mode, as it is stated by Lem. 6. This certainly ensures that  $\Xi_{|F|} = \emptyset$ .

**Lemma 3. [Lower Bound on  $E_0$  when  $M \geq n - 1$  ( $\Xi_{|F|} = \emptyset$ )].**

When  $E_0 \geq \frac{\lambda}{\lambda-1} \left( \frac{\theta}{1-(1-\theta)^{\lceil \frac{|F|}{2} \rceil}} \right) cv_{\min} E_{wkp}$  and  $M \geq n-1$ , in any execution of EB-TTFM( $\lambda$ ), all non-fastest agents are in *terminated* mode after the first  $cv_{\min}$  interactions.

*Proof.* We know that a configuration where all fastest agents are in *terminated* mode can appear only at or after the  $\text{mni}(|F|)^{\text{th}}$  interaction. Thus, if  $l(|F|) \geq cv_{\min}$ , in any execution of EB-TTFM( $\lambda$ ), there is at least one *non-terminated* fastest agent at the end of the first  $cv_{\min}$ . Moreover, all the non-fastest mobile agents are in *terminated* mode, since every non-fastest mobile agent interacts with a *non-terminated* fastest agent during the first  $cv_{\min}$  and has transferred all of its value (as the memory is unbounded) in this interval.

Substituting (2.1) in  $l(|F|)$  and expressing the formula in function of  $E_0$  yields the formula appearing in Lemma 6.  $\square$

Then, similarly to the bounded case in the previous section, we compute  $E_{s_{\max}}(\text{EB-TTFM}(\lambda))$  for  $E_0$  satisfying the bounds of Lemmas 1 and 6.

**Theorem 4. [Worst Case Energy Analysis when  $M \geq n - 1$ ].**

If  $\frac{\lambda}{\lambda-1} \left( \frac{\theta}{1-(1-\theta)^{\lceil \frac{|F|}{2} \rceil}} \right) cv_{\min} E_{wkp} \leq E_0 \leq (2cv_{\min} - 1) \frac{\lambda}{\lambda-1} E_{wkp}$  and  $cv_{\min} \gg O(n^2)$ , then  $E_{s_{\max}}(\text{EB-TTFM}(\lambda)) \leq \frac{\lambda-1}{\lambda} E_0 + (1 + \frac{1}{2-\theta}) cv_{\min} E_{wkp} = \Theta(cv_{\min} E_{wkp})$ .

*Proof.* Assume that for an execution  $e$ , the energy of agent  $i$  decreases to  $\frac{E_0}{\lambda}$  at time  $t_i$ . Then, there are two possible states for agent  $i$  after  $t_i + cv_{\min}$ :

- Agent  $i$  is in *terminated* mode: It means that agent  $i$  has already transferred its value. Therefore, in this case, its residual energy at the end of execution,  $el_i(e) = el_i(t_i + cv_{\min}) \geq \frac{E_0}{\lambda} - cv_{\min} E_{wkp}$ .
- Agent  $i$  is not in *terminated* mode:  
If agent  $i$  is still alive after  $t_i + cv_{\min}$ , for any interaction  $(i, j)$  at time  $t \in$

$[t_i, t_i + cv_{min}]$  with an agent  $j$  still *non-terminated*,  $el_j(t) < el_i(t) \leq \frac{E_0}{\lambda}$ . Thus, after  $t_i + cv_{min}$ , agent  $i$  is the only *non-terminated* agent. And agent  $i$  is the agent that transfers the last value to the base station.

Let agent  $j$  be the last agent turning into *terminated* mode during  $[t_i, t_i + cv_{min}]$  and let  $t_j$  be the time where its energy decreases to  $\frac{E_0}{\lambda}$ . Let  $t_{ij} \in [t_i, t_i + cv_{min}]$  be the time when *non-terminated*  $j$  interacts with  $i$ . We know that  $t_{ij} > t_j$  since  $el_j(t_{ij}) < el_j(t_j) = \frac{E_0}{\lambda}$ . Denoting by  $x_i$  the number of interactions of agent  $i$  during  $[t_i, t_{ij}]$  and by  $x_j$  the number of interactions of agent  $j$  during  $[t_j, t_{ij}]$ :

1. First, according to the definition of  $x_i$  and  $x_j$ , we have  $x_i \leq t_{ij} - t_i$ ,  $x_j \leq t_{ij} - t_j$ .
2. As  $el_j(t_{ij}) < el_i(t_{ij})$ ,  $\frac{E_0}{\lambda} - x_j(E_{wkp} - E_{slp}) - (t_{ij} - t_j)E_{slp} < \frac{E_0}{\lambda} - x_i(E_{wkp} - E_{slp}) - (t_{ij} - t_i)E_{slp}$ . Thus, we have  $x_i < x_j + (t_i - t_j) \frac{\theta}{1 - \theta}$  where  $\theta = \frac{E_{slp}}{E_{wkp}}$ .
3. There is no interaction between  $i$  and  $j$  during the interval  $[\min(t_i, t_j), t_{ij})$ . Indeed, suppose that there is one at time  $t' \in [\min(t_i, t_j), t_{ij})$ ,  $k_1 = \arg \min_{x \in \{i, j\}} t_x$  and  $k_2 = \{i, j\} \setminus k_1$ . We know that  $el_{k_1}(t') \leq el_{k_1}(t_{k_1}) = \frac{E_0}{\lambda}$ . If  $el_{k_1}(t') < el_{k_2}(t')$ , agent  $k_1$  transfers its values to agent  $k_2$ . If  $el_{k_1}(t') > el_{k_2}(t')$ , as  $el_{k_2}(t') < el_{k_1}(t') \leq \frac{E_0}{\lambda}$ , agent  $k_2$  transfers its values to agent  $k_1$ . Then, after the interaction at time  $t'$ , either agent  $k_1$  or agent  $k_2$  is in *terminated* mode which contradicts the interaction of  $(i, j)$  at time  $t_{ij}$  where both agents are *non-terminated*.

Since there is no interaction between agent  $i$  and  $j$  during the interval  $[\min\{t_i, t_j\}, t_{ij})$  and since there is at least one interaction between  $i$  and  $j$  for each  $cv_{min}$  time interval, according to the definition of a cover time, we have  $t_{ij} \leq \min\{t_i, t_j\} + cv_{min}$ . Thus,  $x_i + x_j \leq cv_{min}$ .

Taken all the inequalities obtained above into consideration,  $x_i$  reaches its maximum value  $\frac{1}{2-\theta}cv_{min}$  when  $t_{ij} - t_j = cv_{min}$ ,  $t_{ij} - t_i = \frac{1}{2-\theta}cv_{min}$ ,  $x_j = \frac{1-\theta}{2-\theta}cv_{min}$ . Figure 2.3 gives an illustration of this case.

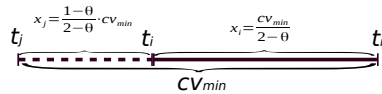


Figure 2.3: Illustration of the case where  $x_i$  has the maximum value.

As the execution ends before  $t_{ij} + cv_{min}$ , agent  $i$  interacts at most  $cv_{min}$  times after  $t_{ij}$ . Thus, from  $t_i$  to the end of execution, agent  $i$  interacts at most  $x_i + cv_{min} = \frac{3-\theta}{2-\theta}$  times. Therefore, in the worst case, agent  $i$  will consume  $\frac{3-\theta}{2-\theta}cv_{min}E_{wkp}$  energy at the end of execution  $e$ .

In conclusion, for any agent  $i \in F$  whose energy decreases to  $\frac{E_0}{\lambda}$  before the end of an execution  $e$ , we have proved that agent  $i$  should consume at most  $\frac{3-\theta}{2-\theta}cv_{min}E_{wkp}$

energy. Therefore, we have:

$$Es_{max}(\text{EB-TTFM}(\lambda)) \leq \frac{\lambda - 1}{\lambda} E_0 + \left(1 + \frac{1}{2 - \theta}\right) cv_{min} E_{wkp}.$$

□

### Best Choice for $\lambda$ .

Now, given  $E_0$  (large enough for accomplishing the task), we study the best choice for  $\lambda$  for minimizing  $Es_{max}(\text{EB-TTFM}(\lambda))$ . We can see from Th. 4 that  $Es_{max}(\text{EB-TTFM}(\lambda))$  decreases with  $\lambda$ , provided that  $\lambda$  satisfies, in particular,  $E_0 \geq \frac{\lambda}{\lambda - 1} (\theta / (1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil})) cv_{min} E_{wkp}$ .

Therefore, the smallest value of  $\lambda$  is obtained when  $\frac{\lambda}{\lambda - 1} \left(\frac{\theta}{1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}}\right) cv_{min} E_{wkp} = E_0$ , which is

$$\tilde{\lambda} = E_0 / \left(E_0 - \left(\frac{\theta}{1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}}\right) cv_{min} E_{wkp}\right).$$

Now, we study the consequences of setting  $\lambda$  to  $\tilde{\lambda}$  in EB-TTFM( $\lambda$ ). The best energy level is:

$$\frac{E_0}{\tilde{\lambda}} = E_0 - \left(\frac{\theta}{1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}}\right) cv_{min} E_{wkp}$$

, that is the initial energy minus a fixed amount, which is precisely

$$E_{fix} = \left(\frac{\theta}{1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}}\right) cv_{min} E_{wkp}.$$

What is the explanation for that? Actually, in the analysis of EB-TTFM( $\lambda$ ), an important point is the relation between  $E_0$  and  $\lambda$  ensuring that there is at least one *non-terminated* fastest agent at the end of the first  $cv_{min}$ , i.e.,  $\Xi_{|F|} \neq \emptyset$ .  $E_{fix}$  is the minimum energy consumption of an agent that ensures that  $\Xi_{|F|} \neq \emptyset$ . The interpretation of  $E_{fix}$  in the process of EB-TTFM( $\tilde{\lambda}$ ) is the following. Once an agent has consumed more than  $E_{fix}$  energy, it transfers its values to more powerful agents and turns into *terminated* mode.

So, we can say that  $E_{fix}$  is the amount of energy to activate the condition in line 2 of EB-TTFM( $\tilde{\lambda}$ ). In other words, the best strategy to increase the lifetime in the worst case of EB-TTFM( $\lambda$ ) is to activate the condition as soon as an agent has consumed  $E_{fix}$  energy, i.e., when it is certain that there will be at least one *non-terminated* fastest agent at the end of the first  $cv_{min}$ .

Thus, the best worst-case performance of EB-TTFM( $\lambda$ ) is when  $\lambda$  is set to  $\tilde{\lambda}$ :

$$\begin{aligned} Es_{max}(\text{EB-TTFM}(\tilde{\lambda})) &= E_{fix} + \left(1 + \frac{1}{2 - \theta}\right) cv_{min} E_{wkp} \\ &= \left(1 + \frac{\theta}{1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}} + \frac{1}{2 - \theta}\right) cv_{min} E_{wkp}. \end{aligned}$$



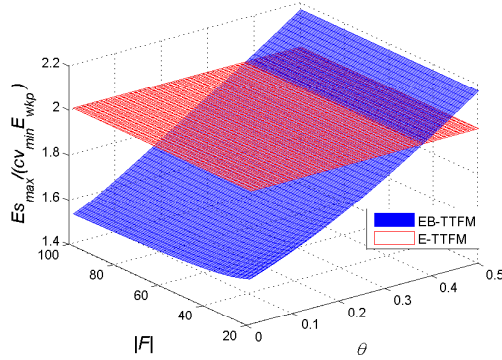


Figure 2.4: The comparison of  $E_{s_{max}}$  between EB-TTFM( $\tilde{\lambda}$ ) and E-TTFM

### Interpretation of Results: Comparison of Energy Performance between EB-TTFM( $\tilde{\lambda}$ ) and E-TTFM

We give a meaningful example of the interest to have analytical functions for describing the energy consumption performance. This example considers the unbounded memory case and illustrates conditions under which EB-TTFM( $\tilde{\lambda}$ ) outperforms E-TTFM.

When the memory is not bounded and  $cv_{min} \gg O(n^2)$ , by Prop. 1,  $E_{s_{max}}(\text{E-TTFM}) = (2cv_{min} - 1)E_{wkp}$ , and by Th. 4, when  $\lambda$  is set to  $\tilde{\lambda}$ ,  $E_{s_{max}}(\text{EB-TTFM}(\tilde{\lambda})) \leq (1 + \theta / (1 - (1 - \theta)^{\lceil \frac{|F|}{2} \rceil}) + \frac{1}{2-\theta})cv_{min}E_{wkp}$ . Then, we obtain that for  $|F| \geq 10$  and  $\theta \leq (3 - \sqrt{5})/2 \approx 0.38$ ,  $E_{s_{max}}(\text{EB-TTFM}(\tilde{\lambda})) \leq E_{s_{max}}(\text{E-TTFM})$ , i.e., EB-TTFM( $\tilde{\lambda}$ ) outperforms E-TTFM.

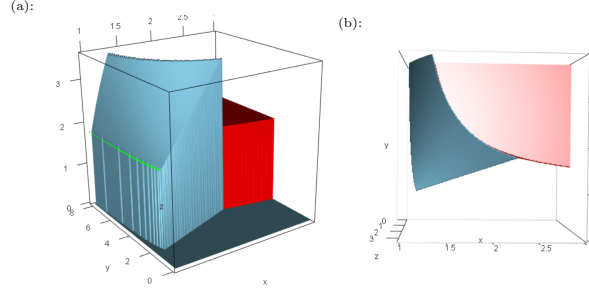
Figure. 2.4 represents the comparison of  $E_{s_{max}}$  between EB-TTFM( $\tilde{\lambda}$ ) and E-TTFM with different  $|F|$  and  $\theta$ . The red (lighter) plane shows  $E_{s_{max}}(\text{E-TTFM})$  and the blue (darker) one represents  $E_{s_{max}}(\text{EB-TTFM}(\tilde{\lambda}))$ . We can see that EB-TTFM( $\tilde{\lambda}$ ) is more energy balanced when  $\theta$  is small, i.e., when the energy spent in *sleep* mode is much less than the energy spent in *awake* mode.

### Interpretation of Results: Relationship between $\lambda$ , $E_0$ and $E_{s_{max}}$ in EB-TTFM( $\lambda$ ).

In Section 2.4.2, we discussed the best choice for  $\lambda$  in EB-TTFM( $\lambda$ ). But what if the value of  $\lambda$  is not set to  $\tilde{\lambda}$  and what is the performance of EB-TTFM( $\lambda$ ) in this case? In this section, we use 3D plots to show the relation among  $\lambda$ ,  $E_0$  and  $E_{s_{max}}(\text{EB-TTFM}(\lambda))$ .

Firstly, from Lemma 6, we have a lower bound for  $E_0$  depending on  $\lambda$ . Then, 3D plots of  $\lambda$ ,  $E_0$  and  $E_{s_{max}}$  can be constructed following Th. 4. Figure 2.5 shows the performance of EB-TTFM( $\lambda$ ) where  $|F| = 10$ ,  $\theta = 0.2$ . x axis represents  $\lambda$ , y axis  $E_0$  and z axis  $E_{s_{max}}(\text{EB-TTFM}(\lambda))$ .

The blue polytope in Figure 2.5a presents the possible values of  $\lambda$  and  $E_0$  for which EB-TTFM( $\lambda$ ) does not behave like E-TTFM. As we can see,  $\lambda$  should be carefully chosen to a small value for EB-TTFM( $\lambda$ ) to perform better than E-TTFM in the


 Figure 2.5: 3D plot on the relationship of  $\lambda$ ,  $E_0$  and  $E_{smax}$ 

worst case. The green border on the blue polytope represents the best choice for  $\lambda$  (Sect. 2.4.2). The red polygon in Figure 2.5a shows the region of values for  $\lambda$  and  $E_0$  where EB-TTFM( $\lambda$ ) behaves like E-TTFM.

Figure 2.5b gives a view of  $x, y$  axis which shows the feasible values of  $\lambda$  and  $E_0$  (blue and red area).

## 2.5 Lower bound on $E_{smax}$

In this section, we present a lower bound (Th. 5) on the maximum energy spent by an agent ( $E_{smax}$ ) for achieving data collection. The protocols considered here, like in the whole study, are those that can compare (but not use in any other way) the cover times and the residual energies of interacting agents. Let us denote this class of protocols by  $\mathbf{P}^{cv\cup e}$ . Following the intuition above for strategies of energy-balanced data collection protocols, we study the lower bound for different sub-classes of protocols in  $\mathbf{P}^{cv\cup e}$ , distinct in their strategies of turning agents into *terminated* mode.

Hence, we decompose  $\mathbf{P}^{cv\cup e}$  into subsets  $\mathbf{P}_i$ , defined below. For that, we order the agents according to their cover times and denote by  $F_i$  the set of mobile agents with the the  $i^{th}$  smallest cover time. Thus the set of fastest agents is  $F_1$ ,  $cv^i$  is the value of the  $i^{th}$  smallest cover time and  $ns$  is the number of different cover times. Then, we define  $\mathbf{P}_i$  as the class of protocols for which, there is at least one *non-terminated* agent in  $\cup_{j=1}^i F_j$  at the end of any execution (or turning into *terminated* mode during the last interaction of the execution). Moreover, in  $\mathbf{P}_i$ , when  $i > 1$ , there exists at least one execution in which all agents in  $\cup_{j=1}^{i-1} F_j$  are *terminated* before the end of execution, i.e.,

$$\begin{aligned} \mathbf{P}_i &= \{ \mathcal{P} \in \mathbf{P}^{cv\cup e} \mid \\ &\quad \forall e \in \mathbf{e}(\mathcal{P}), \exists f \in \cup_{j=1}^i F_j, Mode(f, e) \neq terminated \wedge \\ &\quad \text{if } (i > 1) : \\ &\quad \exists e \in \mathbf{e}(\mathcal{P}), \forall f \in \cup_{j=1}^{i-1} F_j, Mode(f, e) = terminated \}, \end{aligned}$$

where  $Mode(f, e) \in \{awake, sleep, terminated\}$  indicates the energy consumption mode of agent  $f$  just before the last interaction in execution  $e$  and  $\mathbf{e}(\mathcal{P})$  is the set of all executions of protocol  $\mathcal{P}$ .

By definition, the  $\mathbf{P}_i$ 's are disjoint and  $\cup_{i=1}^{n_s} \mathbf{P}_i = \mathbf{P}^{cv \cup e}$ . Moreover, E-TTFM belongs to  $\mathbf{P}_1$ . Note that EB-TTFM( $\lambda$ ) can belong to different subsets depending on the value of  $\frac{E_0}{\lambda}$ . Under the conditions of Lem. 2, EB-TTFM( $\lambda$ ) is in  $\mathbf{P}_1$ .

We obtain the lower bound on energy, using in particular the lower bound on time (the length of an execution) of [26]. This bound holds for any data collection protocol using only cover time comparisons. Let  $\mathbf{P}^{cv}$  be this class of protocols. The proof of the lower bound on time for  $\mathbf{P}^{cv}$  in [26] holds also for  $\mathbf{P}^{cv \cup e}$ , because only constraints related to the data collection problem are invoked. For example, it is necessary that every mobile agent either meets BST by itself, or meets at least one other agent that meets BST, before convergence. Then comparisons of the residual energies of two interacting agents cannot produce a shorter execution. This is expressed in the following observation.

Observation: The lower bound on time for any data collection in  $\mathbf{P}^{cv}$  [26] is also correct for any protocol in  $\mathbf{P}^{cv \cup e}$ .

Recall that this bound is tight and equal to the upper bound on time of the protocol TTFM, also presented in [26]. Hence, we denote this bound here by  $|e_{max}(\text{TTFM})|$ .

We still assume that  $cv_{min} \gg O(n^2)$  (see justifications in Sect. 2.1.3).

**Lemma 4. [First Lower Bound on  $E_{s_{max}}$  for Protocols in  $\mathbf{P}_i$ ]**

If  $cv_{min} \gg O(n^2)$ , for any protocol  $\mathcal{P}$  in  $\mathbf{P}_i$ , the energy spent by an agent in the worst case is at least  $E_{s_{max}}(\text{E-TTFM}) / \lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil$ .

*Proof.* First, consider the following schedule which repeats the segment  $X_1 X_2$ , where  $X_1 X_2$  is of length  $cv_{min}$ .

$X_1 \equiv$  [ repeat as much as possible all the possible interactions between agents in  $\cup_{j=1}^i F_j$  ],

$X_2 \equiv$  [ all the other necessary interactions to satisfy  $cv_s$  ].

Consider now the execution of  $\mathcal{P} \in \mathbf{P}_i$  for this schedule, denoted by  $e$ . Since no agent consumes more than  $E_{s_{max}}(\mathcal{P})$  amount of energy in any execution of  $\mathcal{P}$ , when an agent consumes  $E_{s_{max}}(\mathcal{P})$  in  $e$ , either it turns into *terminated* mode or  $e$  reaches the final configuration. And as  $\mathcal{P} \in \mathbf{P}_i$ , there is at least one agent in  $\cup_{j=1}^i F_j$  which is *non-terminated* at the end of  $e$ . Moreover, notice that  $|X_2| \leq \frac{n(n-1)}{2} = O(n^2)$  and as  $cv_{min} \gg O(n^2)$ , the length of  $X_2$  is negligible. Thus, following the above schedule, at time  $t = E_{s_{max}}(\mathcal{P}) \times \lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil / E_{wkp}$ , all agents in  $\cup_{j=1}^i F_j$  consume  $E_{s_{max}}(\mathcal{P})$  amount of energy. However, the length of  $e$  must be smaller than or equal to  $t$ . Because otherwise, either there would be a *non-terminated* agent consuming more than  $E_{s_{max}}$ , or all agents would be *terminated*, which contradicts the fact that  $\mathcal{P} \in \mathbf{P}_i$ . Thus, we have  $t \geq |e|$ , i.e.,

$$\frac{E_{s_{max}}(\mathcal{P})}{E_{wkp}} \times \lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil \geq |e|.$$

By the observation above, execution  $e$  satisfies the lower bound on time of [26]. Thus, we obtain  $|e| \geq |e_{max}(\text{TTFM})|$

$= |e_{max}(\text{E-TTFM})|$ , as E-TTFM has the same time performance as TTFM (Sect. 2.3). Moreover, by Prop. 1, when  $cv_{min} \gg O(n^2)$ ,  $|e_{max}(\text{E-TTFM})|E_{wkp} = Es_{max}(\text{E-TTFM})$ . Then, we have

$$\begin{aligned}
 Es_{max}(\mathcal{P}) &\geq \frac{|e| \times E_{wkp}}{\lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil} \geq \frac{|e_{max}(\text{E-TTFM})|E_{wkp}}{\lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil} \\
 &= Es_{max}(\text{E-TTFM}) / \lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil.
 \end{aligned}$$

□

**Lemma 5. [Second lower Bound on  $Es_{max}$  for Protocols in  $\mathbf{P}_i$ ]**

If  $cv_{min} \gg O(n^2)$ , for any protocol  $\mathcal{P}$  in  $\mathbf{P}_i$ , the energy spent by an agent in the worst case is at least  $cv^i E_{wkp}$ .

*Proof.* When  $i > 1$ , as  $\mathcal{P} \in \mathbf{P}_i$ , there exists at least one execution  $e$  in which all agents in  $\cup_{j=1}^{i-1} F_j$  are *terminated* before the end of the execution. Let  $t$  indicate the time when all agents in  $\cup_{j=1}^{i-1} F_j$  are *terminated* in  $e$  and let  $f \notin \cup_{j=1}^{i-1} F_j$  be a *non-terminated* agent (with cover time  $cv_f \geq cv^i$ ), holding one of the values not yet collected by BST at time  $t + 1$ . We prove the existence of another execution  $e'$  such that  $Es_{max}(e') \geq cv^i E_{wkp}$ . The execution  $e'$  begins with the same prefix of  $t$  events as  $e$  and continues by meetings between agent  $f$  and the *terminated* agents until  $f$  meets BST at time  $t + cv_f$ . In our energy consumption scheme, when a sleeping agent meets a *terminated* agent, it consumes  $E_{wkp}$  and does not change its mode (See Sect. 2.2). Therefore and because  $cv_f > cv_{min} \gg O(n^2)$ , during the interval  $[t, t + cv_f]$ , agent  $f$  consumes  $cv_f E_{wkp}$  (the necessary interactions involving other agents than  $f$  during  $cv_f$  are negligible, as in the proof of Lem. 4). So, when  $i > 1$ , we have  $Es_{max}(\mathcal{P}) \geq Es_{max}(e') \geq cv_f E_{wkp} \geq cv^i E_{wkp}$ .

Next, we consider the case where  $i = 1$ . We prove that  $Es_{max}(\mathcal{P}) \geq cv_{min} E_{wkp}$  by contradiction. Suppose that there exists a protocol  $\mathcal{P} \in \mathbf{P}_1$  such that  $Es_{max}(\mathcal{P}) < cv_{min} E_{wkp}$ . To contradict this, we construct an execution  $e$  of  $\mathcal{P}$  with  $Es_{max}(e) \geq cv_{min} E_{wkp}$ . Consider again the schedule  $X_1 X_2$  in the proof of Lem. 4. It is possible that at time  $t = Es_{max}(\mathcal{P}) / E_{wkp} < cv_{min}$ , an agent  $j \in F_1$  consumes  $Es_{max}(\mathcal{P})$  and turns into *terminated*. For the same reason as above, we can construct  $e$  in which a *non-terminated* agent  $f$  meets the *terminated* agent  $j$  repetitively until it meets others at time  $t + cv_f$ . Thus, when  $i = 1$ , we have  $Es_{max}(\mathcal{P}) \geq cv_{min} E_{wkp}$ . □

**Theorem 5. [Lower Bound on  $Es_{max}$  for Protocols in  $\mathbf{P}^{cv \cup e}$ ]** If  $cv_{min} \gg O(n^2)$ , for any protocol in  $\mathbf{P}^{cv \cup e}$ , the energy spent by an agent in the worst case is at least  $\max\{Es_{max}(\text{E-TTFM}) / \lceil \frac{|F|}{2} \rceil, cv_{min} E_{wkp}\}$ .

*Proof.* The result comes directly from Lem. 4 and 5, since  $\cup_i \mathbf{P}_i = \mathbf{P}^{cv \cup e}$  and

$$\begin{aligned}
 &\min_i \max\{Es_{max}(\text{E-TTFM}) / \lceil \frac{|\cup_{j=1}^i F_j|}{2} \rceil, cv^i E_{wkp}\} \\
 &= \max\{Es_{max}(\text{E-TTFM}) / \lceil \frac{|F|}{2} \rceil, cv_{min} E_{wkp}\}.
 \end{aligned}$$

□

Next, we identify the cases where this bound is reached by the presented protocols (Cor. 1), and where it differs only by a constant multiplicative factor from the energy complexity of EB-TTFM( $\lambda$ ), for the case of non-bounded memory and with a good choice of  $\lambda$  (Cor. 2).

**Corollary 1.** *When  $|F| \leq 2$  and  $cv_{min} \gg O(n^2)$ , E-TTFM is energy-optimal in  $\mathbf{P}^{cv\cup e}$ , with respect to the energy spent by an agent in the worst case.*

*Proof.* From Th. 5, if  $|F| \leq 2$ ,  $\forall \mathcal{P} \in \mathbf{P}^{cv\cup e}$ ,  $Es_{max}(\mathcal{P}) \geq Es_{max}(\text{E-TTFM})$ . As  $\text{E-TTFM} \in \mathbf{P}^{cv\cup e}$ , E-TTFM is energy optimal in  $\mathbf{P}^{cv\cup e}$ .  $\square$

**Corollary 2.** *When  $|F| > 2$  and  $cv_{min} \gg O(n^2)$ , for any protocol in  $\mathbf{P}^{cv\cup e}$ , the energy spent by an agent in the worst case is at least  $cv_{min}E_{wkp}$ . For the case where  $M \geq n - 1$  and  $\lambda$  satisfies the conditions of Th. 4, EB-TTFM( $\lambda$ ) reaches this bound asymptotically.*

*Proof.* This result comes directly from Th. 5 and Th. 4.  $\square$

## 2.6 Conclusion

In this chapter, a formal energy model for population protocols has been presented and studied. Designing a good model is difficult. If the model is too abstract (or general), it won't easily apply to reality. If it is too low level and considers too many parameters, it won't allow formal treatments. We believe that the model we introduced is a good compromise between the two approaches. On the one hand, it is based on practical information on how real resource-limited mobile sensor networks are functioning, and on the other hand, as shown in this chapter, it allows to obtain analytically precise formulas, which are obviously not directly intuitive.

Moreover, the proposed energy model is adaptable to many types of mobile sensor networks and their applications, and to those that are not considered in this work (e.g., to the case where the transmitted data does not always fit into one packet and in general, to the case of less limited agents). As already noticed, the assumption on the cover times does not particularize the model, since almost all real mobile sensor networks either operate in a bounded area (town, factory, security zone, etc.) or satisfy the home coming tendency (networks related to human or animal mobility). Nevertheless, other kinds of "periodic" conditions on the interactions (e.g., probabilistic interactions) can be considered to study energy consumption using the proposed framework. Finally, other important communication problems (as broadcast, all-to-all communication, routing, etc.) have many points in common with data collection (excepted that unique identifiers may be needed, like in community protocols [67]). Hence, the techniques developed here can be useful in the future studies of these problems.

The second contribution of the chapter consists in establishing analytical formulas for lower bounds and for the energy consumption of two proposed data collection algorithms. We want to emphasize that having analytical formulas allows obtaining information that could not be obtained by the sole intuition, and that would be difficult to obtain by simulations. On one hand, the study of the curves and plots

allows to get easily the conditions on the parameters (e.g., think of the value of  $\frac{3-\sqrt{5}}{2}$  for  $\theta$  in Sect. 2.4.2). On the other hand, the analytical approach allows obtaining general upper and (always difficult to establish) lower bounds.

To conclude, we summarize the main technical results of this chapter. In Sect. 2.4.1, we obtain formulas for the worst-case performance of energy in EB-TTFM( $\lambda$ ). Then, for the case of non-bounded memory, we compute the best value for  $\lambda$  with respect to  $E_0$ ,  $|F|$ ,  $E_{wkp}$ ,  $E_{slp}$  and  $cv_{min}$  (Sect. 2.4.2). Further analysis with the best  $\lambda$  shows that no matter how much initial energy an agent has, once it consumes more than some calculated amount  $E_{fix}$ , the best strategy is to switch to *terminated* mode, when interacting with a more powerful agent. We interpret the formulas graphically, and show that EB-TTFM( $\lambda$ ) is more energy efficient in the case where the number of fastest agents are more than 10 and the ratio between  $E_{slp}$  and  $E_{wkp}$  is less than  $\frac{3-\sqrt{5}}{2}$  (Sect. 2.4.2). At the end (Sect. 2.5), we give a uniform lower bound for energy consumption for all possible data collection protocols. We identify the cases where this bound is reached by the presented protocols (Cor. 1), and where it differs only by a constant multiplicative factor from the energy upper bound of EB-TTFM( $\lambda$ ) (Cor. 2).

# Power-Aware Population Protocols under Non-uniformly Random Scheduler

---

## 3.1 Introduction

In population protocols, the fact that agent moves are unpredictable is usually modeled by assuming the uniformly random scheduler ([11, 15, 16, 22]). That is, the interactions between any two agents are drawn uniformly at random. However, for some practical sensor networks, this assumption may be unrealistic. Consider, for instance, agents moving at different speeds. In this case, an agent interacts more frequently with a faster agent than with a slower one. In other networks, certain agents may be frequently prevented from communicating with some others, because they move in different limited areas, or disfunction from time to time, etc. In all these examples, the interactions are clearly not uniformly random. There are thus strong arguments for enhancing the basic model.

This chapter initiates the study of non-uniform schedulers in the context of population protocols. Considering the scheduler as a generator of sequences of pairwise interactions, non-uniform means that the next interacting pair  $(i, j)$  is chosen with a non-uniform probability  $P_{i,j}$ , depending on  $i$  and  $j$ .

As an additional justification for studying a non-uniform scheduler, notice that many experimental and analytical studies of different (finite boundary) mobile sensor networks show and exploit (respectively) the assumption that the *inter-contact* time of two agents (the time period between two successive interactions of the same two mobile agents) is distributed exponentially (cf. [34, 60, 117, 150]). Similarly, under a non-uniformly random scheduler, it appears that the inter-contact time  $T_{i,j}$ , of any two agents  $i$  and  $j$ , follows a geometric distribution ( $P[T_{i,j} = t] = (1 - P_{i,j})^{t-1}P_{i,j}$ ), which is the discrete analogue of the exponential case (observed in practical mobile networks).

The counterpart of considering a non-uniform scheduler is a more complex analysis. Though, it remains feasible in certain cases, as it is shown in this chapter. To illustrate this point, we consider the previously studied fundamental task, *data collection*. In the context of population protocols (assuming non-random schedulers), several data collection protocols have been proposed and their complexity in time has been studied [26] under cover times fairness. Notice that the analysis there was only for the worst case. However, we are not aware of any previous results concerning the average complexity of these protocols. The current chapter presents protocols that basically use the simple ideas of the TTF (Transfer To the Faster)

protocol of [26] (See Sect. 1.2.2). The new protocols are adapted to a non-uniform scheduler and improve energy consumption, as explained further.

First, consider the original version of TTF. The data transfer between the agents in TTF depends on the comparison of cover times of two interacting agents. Here we follow this idea. However, as the scheduler is probabilistic, we adapt the corresponding definition of the cover time to be the *expected* (instead of the maximum) number of interactions for an agent to interact with every other agent (see Sect. 3.2). Moreover, in this preliminary study, we make the assumption that every agent has enough memory to store  $n$  values. With these conditions, agent  $i$  could transfer any number of values to  $j$  (there is never more than  $n$ ). This assumption is common in the literature [11, 68].

The complexity analysis starts with the proofs of two lower bounds on the expected convergence time of any protocol solving data collection (Sect. 3.3). Then, an analysis of execution times in expectation and with high probability (w.h.p.), for the new version of TTF, is given (Sect. 3.4). The complexity in expectation indicates how the protocol is good in average, while the complexity w.h.p. tells how it is good almost all the time. We obtain explicit bounds, thus justifying the relevance of the enhanced model in protocol analysis and its operability.

We further investigate the non-uniform model by addressing also energy complexity. The goal is to improve energy consumption of TTF, while keeping good time complexity. For that, we propose a new parametrized protocol, called *lazy* TTF (Sect. 3.5). As opposed to TTF, it does not execute necessarily the transition of TTF resulting from an interaction. Instead, during an interaction  $(i, j)$ , TTF is executed with probability  $p_i$  (depending on agent  $i$ , playing the role of *initiator* in the interaction). Analysis and the corresponding numerical study show that a good choice of the parameters  $p_i$  results in lower energy consumption. To find such parameters, we formulate and solve a polynomial-time optimization program. The resulting optimized lazy TTF is compared to TTF in respect with time and energy complexity (Sect. 3.6). For this analysis, we adopt the energy scheme proposed for population protocols in Sect. 2.2.

This work is presented in ALGOSENSORS 2017 [27].

### 3.1.1 Related work

The uniformly random scheduler has been introduced and studied in the context of population protocols in the seminal paper [14]. Later, leader election protocols (cf. [10, 51]) and exact majority protocols (cf. [11]) have been proposed in this uniform model. Their performances are evaluated by the parallel expected convergence time and by the number of states available at each agent (space complexity). In several papers (cf. [6]), trade-offs between time and space complexities of protocols solving these problems are studied. Any leader election or majority protocol converges in  $\Omega(n/\text{polylog } n)$  expected time using  $O(\log \log n)$  states. A recent work [9] shows that, by employing “phase clocks”, both problems can be solved in  $O(\log^2 n)$  expected time, using  $O(\log n)$  states. Other complex problems, such as counting [22], community detection [28] and proportion computation [92], have been also studied under similar uniform scheduler model.



Besides the uniformly random scheduling independent of the agents states, there are works assuming a scheduling depending on the states of agents, like the *transition function* scheduler in [41] or the scheduling of reactions in CRN (Chemical Reaction Network model) according to the model of stochastic chemical kinetics (cf. [46]).

We should also mention the randomized gossip algorithm in [31] designed for the problem of averaging, in an arbitrarily connected network. Each node runs an independent Poisson clock (asynchronous time model), and at each clock tick, the node randomly selects a neighbor, with the probability given by the algorithm. Then, it averages its value with the chosen neighbor. Observe that this algorithm can be seen as a population protocol under non-uniformly random scheduler, in which two meeting agents average their values. For more details on gossip algorithms, refer to the surveys [115] and [49].

### 3.2 Model and Definitions

**Non-uniformly random scheduler.** Such a scheduler, denoted by  $S(P)$ , is defined by a matrix of probabilities  $P \in \mathbb{R}^{n \times n}$ . During an execution,  $S(P)$  chooses the next pair of agents  $(i, j)$  to interact (taking  $i$  as initiator and  $j$  as responder) with the probability  $P_{i,j}$ . Notice that, in the case of the matrix with entries  $P_{i,j} = 1/n(n-1)$  for  $i \neq j$ , and  $P_{i,i} = 0$ , the scheduler chooses each pair of agents uniformly at random for each next interaction (i.e., the scheduler is uniformly random).

The matrix  $P$  satisfies  $\sum_{i=1}^n \sum_{j=1}^n P_{i,j} = 1$  and  $\forall i \in \{1, \dots, n\}, P_{i,i} = 0$ , since interactions are pairwise. Moreover, for any edge  $(i, j)$  in the interaction graph  $G$ ,  $P_{i,j} > 0$ . As the graph considered here is complete, *every* pair of agents is chosen infinitely often with probability 1.

For a given  $P$ , one can compute the *expected* (finite) time for a given agent  $i$  to meet all the others. We call it *cover time of agent  $i$*  and denote it by  $cv_i$ . By resolving the coupon collector's problem with a non-uniform distribution [59], we obtain the cover time of each agent:  $cv_i = \int_0^\infty (1 - \prod_{j \neq i} (1 - e^{-(P_{i,j} + P_{j,i})t})) dt$ . Similarly to [26], for two agents  $i$  and  $j$ , if  $cv_i < cv_j$ , we say that  $i$  is *faster* than  $j$ , and  $j$  is *slower* than  $i$ . If  $cv_i = cv_j$ ,  $i$  and  $j$  are said to be in the same *category* of cover times. We denote by  $m$  the number of different categories of cover times.

**Time Complexity Measures.** The *convergence time* of a data collection protocol  $\mathcal{P}$  can be evaluated in two ways: first, in terms of expected time until termination, denoted by  $T_E(\mathcal{P})$ , and second, in terms of time until termination w.h.p.<sup>1</sup>, denoted by  $T_{whp}(\mathcal{P})$ . The termination of a data collection protocol has been defined in Sect. ??.

**Remark 1.** *The notion of parallel time, which is common when considering the uniformly random scheduler (cf. [16, 17]), is not used in this study. When using this measure of time, it is assumed that each agent participates in an expected number  $\Theta(1)$  of interactions per time unit. With the uniformly random scheduler, this time*

<sup>1</sup>An event  $\Xi$  is said to occur w.h.p., if  $P(\Xi) \geq 1 - \frac{1}{nc}$ , where  $c \geq 1$ .

measure is asymptotically equal to the number of interactions divided by  $n$ . However, with non-uniformly random scheduler, this is no more true.

### 3.3 Lower Bounds on the Expected Convergence Time

We now give two nontrivial lower bounds on the expected convergence time of data collection protocols. The first one (Th. 6) only depends on the number of agents. The second one (Th. 7) depends on the specific values of the probability matrix  $P$  used by the scheduler. The bounds are incomparable in general. To obtain the bounds, we observe that, for performing data collection, each agent has to interact at least once (otherwise, its value simply would not be delivered), and we compute the expected time ensuring that. The proof of Th. 6 uses an analogy with a generalization of the classical coupon collector's problem, which we introduce next.

Let  $k$  be a positive integer. Given a probability distribution  $(p_1, \dots, p_k)$  on  $[k] = \{1, \dots, k\}$ , the corresponding  $k$ -coupon collector's problem is defined by its *coupon sequence*  $(X_1, X_2, \dots)$  of independent and identically distributed (i.i.d.) random variables with  $P(X_t = i) = p_i$  for all  $i \in [k]$  and all  $t \geq 0$ . The  $k$ -coupon collector's problem's *expected time* is the expectation of the earliest time  $T$  such that  $\{X_1, \dots, X_T\} = [k]$ , i.e., all coupons were collected at least once.

More generally, given a set  $\mathcal{A}$  of subsets of  $[k]$  such that  $\bigcup_{A \in \mathcal{A}} A = [k]$ , and a probability distribution  $(p_A)$  on  $\mathcal{A}$ , the corresponding  $\mathcal{A}$ -group  $k$ -coupon collector's problem is defined by its *coupon group sequence*  $(X_1, X_2, \dots)$  of i.i.d. random variables with  $P(X_t = A) = p_A$  for all  $A \in \mathcal{A}$  and all  $t \geq 0$ . Its *expected time* is the expectation of the earliest time  $T$  such that  $\bigcup_{t=1}^T X_t = [k]$ , i.e., all coupons were collected in at least one coupon group.

Given an integer  $1 \leq g \leq k$ , the  $g$ -group  $k$ -coupon collector's problem is the  $\mathcal{A}$ -group  $k$ -coupon collector's problem where  $\mathcal{A} = \{A \subseteq [k] \mid |A| = g\}$ . This generalization of the classical coupon collector's problem has been studied, among others, by Stadje [124], Adler and Ross [5], and Ferrante and Saltalamacchia [56].

The following lemma characterizes the probability distributions that lead to a minimal expected time for the group coupon collector's problem. To the best of our knowledge, this is a new result which generalizes the characterization in the classical coupon collector's problem [59, 95], for which it is known that the uniform distribution leads to the minimal expected time.

**Lemma 6.** *The expected time of any  $\mathcal{A}$ -group  $k$ -coupon collector's problem is greater than or equal to the  $\mathcal{B}$ -group  $k$ -coupon collector's problem with uniform probabilities where  $\mathcal{B} \subseteq \mathcal{A}$  is of minimal cardinality such that  $\bigcup \mathcal{B} = [k]$ .*

*In particular, the expected time of any  $g$ -group  $k$ -coupon collector's problem is  $\Omega(k \log k)$  for every constant  $g \geq 1$ .*

*Proof.* We say that a set  $\mathcal{B} \subseteq \mathcal{A}$  of coupon groups is *covering* if  $\bigcup \mathcal{B} = [k]$ .

For every  $\mathcal{B} \subseteq \mathcal{A}$ , denote by  $T_{\mathcal{B}}$  the time until all coupon groups in  $\mathcal{B}$  appear at least once in the coupon group sequence. Denoting by  $T$  the time until all coupons appear at least once in the coupon group sequence, we have

$$T = \min \{T_{\mathcal{B}} \mid \mathcal{B} \text{ is covering}\} . \quad (3.1)$$

For every covering  $\mathcal{B}$ , let  $F_{\mathcal{B}}$  be the event that  $\mathcal{B}$  is the first covering coupon group set to completely appear in the coupon group sequence. The law of total expectation gives

$$\mathbb{E}(T) = \sum_{\mathcal{B} \text{ covering}} \mathbb{E}(T \mid F_{\mathcal{B}}) \cdot \mathbb{P}(F_{\mathcal{B}}) \geq \min_{\mathcal{B} \text{ covering}} \mathbb{E}(T \mid F_{\mathcal{B}}) . \quad (3.2)$$

By definition of  $F_{\mathcal{B}}$ , we have  $\mathbb{E}(T \mid F_{\mathcal{B}}) = \mathbb{E}(T_{\mathcal{B}} \mid F_{\mathcal{B}})$ . The latter is greater than or equal to the expected time of a  $|\mathcal{B}|$ -collector's problem, which can be seen by shifting probabilities of non- $\mathcal{B}$  coupon groups into  $\mathcal{B}$ . By [95, Theorem 1], this time is then at most that of the  $\mathcal{B}$ -coupon collector's problem with uniform probabilities, i.e.,  $|\mathcal{B}|H(|\mathcal{B}|)$  where  $H(m) = \sum_{\ell=1}^m 1/\ell$  denotes the  $m^{\text{th}}$  harmonic number. This proves the first part of the lemma.

To show the second part, we note that  $\lceil k/g \rceil$  coupon groups of size  $g$  are needed to cover the set  $[k]$ , i.e.,  $|\mathcal{B}| \geq \lceil k/g \rceil$ , which means

$$\mathbb{E}(T) \geq \left\lceil \frac{k}{g} \right\rceil \cdot H\left(\left\lceil \frac{k}{g} \right\rceil\right) \sim \frac{k}{g} \cdot \log \frac{k}{g} = \Omega(k \log k) \quad (3.3)$$

as  $k \rightarrow \infty$  if  $g$  is a constant. □

**Theorem 6.** *The expected convergence time of any protocol solving data collection with non-uniformly random scheduler is  $\Omega(n \log n)$ .*

*Proof.* For data collection, each agent has to transfer its value at least once, and the base station has to receive values at least once. Therefore, in any execution, each agent has to interact at least once. The expected time of every agent interacting at least once is that of a 2-group  $n$ -coupon collector's problem, i.e., is  $\Omega(n \log n)$  by Lemma 6. □

**Theorem 7.** *The expected convergence time of any protocol solving data collection with random scheduler  $S(P)$ , is  $\Omega(\max_i \frac{1}{\sum_{j=1}^n (P_{i,j} + P_{j,i})})$ .*

*Proof.* For any agent  $i$ , it is required at least  $\sum_{j=1}^n 1/(P_{i,j} + P_{j,i})$  time in expectation to establish one interaction. Thus, to complete one data collection, for which it is required that each agent interacts at least once, it takes at least  $\max_i \frac{1}{\sum_{j=1}^n (P_{i,j} + P_{j,i})}$  expected time. □

The next corollary considers a very simple protocol solving the data collection problem. In this protocol, agents transfer their values only when they interact with the base station. We consider it as a reference, to compare with other proposed protocols. The corollary follows from Th. 7.

**Corollary 3.** *With random scheduler  $S(P)$ , the expected convergence time of the protocol solving data collection and where each agent transfers its value only to the base station is  $\Omega(\max_i 1/(P_{i,\text{BST}} + P_{\text{BST},i}))$ .*

### 3.4 Protocol “Transfer To the Faster” (TTF)

Corollary 3 formalizes the straightforward observation that, if the only transfers performed by the agents are towards the base station, the convergence time depends on the slowest agent  $i$ . It can be very large, e.g. if  $P_{i,\text{BST}} + P_{\text{BST},i} \ll 1/n^2$ . Therefore, to obtain better time performances, we propose to study another data collection protocol based on the idea of the TTF protocol of [26]. In the sequel, the studied protocol is called TTF too, since its strategy is the same and there is no risk of ambiguity. The only difference is the *definition* of the cover time parameter (Sect. 3.2) used by this strategy (as explained in Sect. 3.1).

The strategy of TTF is simple. When agent  $i$  meets a faster agent  $j$ ,  $i$  transfers to  $j$  all the values it has in its memory (recall that transfer means to copy to the memory of the other and erase from its own). The intuition behind is that the faster agent  $j$  is more likely to meet the base station before  $i$ . Of course, whenever any agent  $i$  meets the base station, it transfers all the values it (still) has in its memory at that time to the base station. As a matter of fact, no transition depends on the actual value held by the agents. It depends only on the comparison between cover times, which are constants. Thus, the input values can be seen as tokens and the states of every agent can be represented by the number of tokens it currently holds. Recall, that in this study, it is assumed that each agent has enough memory for storing the tokens (i.e., an  $O(n)$  memory), and each pair of agents interacts infinitely often (i.e., the interaction graph is complete).

The sequel concerns analytical results on the time performance of TTF. Firstly, we associate to each configuration a vector of non-negative integers representing the number of tokens held by each agent. Then, it is shown that the evolution of such vectors during executions can be expressed by a *stochastic* linear system. Next,  $T_{\text{whp}}(\text{TTF})$  is expressed in terms of distances between the configuration vectors (Th. 8) and, by applying stochastic matrix theory ([77, 113, 130]) an upper bound on  $T_{\text{whp}}(\text{TTF})$  is obtained (Th. 9). Finally, using this result, we obtain also an upper bound on the convergence time in expectation,  $T_{\text{E}}(\text{TTF})$  (Th. 10).

Formally, we represent a configuration by a non-negative integer vector  $x \in \mathbb{N}^n$  that satisfies  $\sum_{i=1}^n x_i = n - 1$ . By abusing the terminology, we sometimes call such a vector a configuration. We denote the configuration vectors’ space by  $\mathbb{V}$ . By convention, the first element of  $x$  is the number of tokens held by the base station. Since, at the beginning of an execution, every mobile agent owns exactly one token and no token is held by the base station, the initial configuration is  $x_{\text{init}} = \mathbf{1} - \mathbf{e}_1$ , where  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)^T$  is the  $n \times 1$  unit vector with the  $i^{\text{th}}$  component equal to 1. The terminal configuration is  $x_{\text{end}} = (n - 1)\mathbf{e}_1$ .

Let  $x(t) \in \mathbb{V}$  be the discrete random integer vector that represents the configuration just after the  $t^{\text{th}}$  interaction in executions of TTF. We can see that  $\mathbb{P}(x(0) = x_{\text{init}}) = 1$ , and since the base station never transfers tokens to others,  $\mathbb{P}(x(t + 1) = x_{\text{end}}) \geq \mathbb{P}(x(t) = x_{\text{end}})$ . Moreover, since at any moment there is a positive probability for delivering any of the tokens to the base station,  $\lim_{t \rightarrow \infty} \mathbb{P}(x(t) = x_{\text{end}}) = 1$ .

Furthermore, the time complexities of TTF can be formalized using  $x(t)$  by

$$T_E(\text{TTF}) = \sum_{t=1}^{\infty} t \cdot (\mathbb{P}(x(t) = x_{\text{end}} \wedge x(t-1) \neq x_{\text{end}}))$$

and

$$T_{\text{whp}}(\text{TTF}) = \inf \left\{ t \mid \mathbb{P}(x(t) = x_{\text{end}}) \geq 1 - \frac{1}{n} \right\}.$$

To evaluate these time complexities, we study the evolution of  $x(t)$  during executions of TTF. Given time  $t$ , consider a transition rule applicable from a configuration represented by a vector  $v^t$  and resulting in a configuration with vector  $v^{t+1}$ . Suppose that at time  $t$ , the interaction  $(i, j)$  is chosen by the scheduler. If neither  $i$  nor  $j$  are the base station and if  $i$  is faster than  $j$  ( $cv_i < cv_j$ ), agent  $j$  transfers all its tokens to  $i$ . Thus,  $v_i^{t+1} = v_i^t + v_j^t$  and  $v_j^{t+1} = 0$ . The relation between  $v^t$  and  $v^{t+1}$ , in this case, can be expressed by the linear equation  $v^{t+1} = W(t+1)v^t$ , where  $W(t+1) = I + \mathbf{e}_i \mathbf{e}_j^T - \mathbf{e}_j \mathbf{e}_j^T \in \{0, 1\}^{n \times n}$ . If  $cv_i = cv_j$ , no token is transferred and  $v^{t+1} = v^t$ . We still have  $v^{t+1} = W(t+1)v^t$ , but with  $W(t+1) = I$ . On the other hand, if  $j$  is the base station,  $W(t+1) = I + \mathbf{e}_i \mathbf{e}_j^T - \mathbf{e}_j \mathbf{e}_j^T$ , as agent  $i$  transfers all of its tokens to the base station.

As the pair of agents is chosen independently with respect to  $P$ ,  $W(t+1)$  can be seen as a *random* matrix such that with probability  $P_{i,j} + P_{j,i}$ :

$$W(t+1) = \begin{cases} I + \mathbf{e}_i \mathbf{e}_j^T - \mathbf{e}_j \mathbf{e}_j^T & \text{if } cv_i < cv_j \text{ or } i = 1 \text{ or } j = 1 \\ I & \text{if } cv_i = cv_j \end{cases} \quad (3.4)$$

By comparing the resulting probability distributions, we readily verify that the relation between  $x(t)$  and  $x(t+1)$ , i.e.,  $x(t+1) = W(t+1)x(t)$ , is a stochastic linear system with the matrices specified in (3.4).

**Distance.** Consider a function  $d_\gamma(x) : \mathbb{V} \rightarrow \mathbb{R}$ . It associates any  $x$  in  $\mathbb{V}$  to a real number representing a “weighted” Euclidian norm distance between the configuration vector  $x$  and the vector representing a terminal configuration. That is,  $d_\gamma(x) = \|(x - x_{\text{end}}) \circ \gamma\|_2$ , where  $\gamma \in \mathbb{R}^n$  is a real vector,  $\circ$  the entry-wise product, and  $\|\cdot\|_2$  the Euclidean norm. The vector  $\gamma$  can be viewed as a weight vector. We choose  $\gamma$  in such a way that, if there is a transfer of tokens in interaction  $t+1$ , then  $d_\gamma(v^{t+1})$  is smaller than  $d_\gamma(v^t)$ . Intuitively this means that, when a transfer is performed, the resulting configuration is closer to termination.

**Lemma 7.** *Let  $i$  and  $j$  be two agents with  $cv_i < cv_j$ . Consider an interaction between  $i$  and  $j$  in a configuration represented by  $v^t$  and resulting in  $v^{t+1}$ . If  $\gamma_j/\gamma_i \geq \sqrt{2n-3}$ , then  $d_\gamma(v^{t+1}) \leq d_\gamma(v^t)$ .*

*Proof.* During an interaction  $(i, j)$  where  $i$  is faster than  $j$ ,  $j$  transfers its tokens to  $i$ . Suppose that agent  $i$  holds  $a$  tokens in  $v^t$ . To ensure  $d_\gamma(v^{t+1}) \leq d_\gamma(v^t)$ , it suffices to have  $((a+1)^2 - a^2)\gamma_i^2 \leq \gamma_j^2$ , which is equivalent to  $\gamma_j/\gamma_i \geq \sqrt{2a+1}$ . If agent  $j$  has tokens, agent  $i$  cannot have more than  $n-2$  tokens in  $v^t$ , thus  $a \leq n-2$ . Therefore, if  $\gamma_j/\gamma_i \geq \sqrt{2n-3}$ , the lemma is satisfied.  $\square$

Recall that  $m \leq n$  denotes the number of cover time categories (Sect. 3.2).

**Theorem 8.** *The convergence time with high probability of TTF,  $T_{\text{whp}}(\text{TTF})$ , is equal to  $\inf \left\{ t \mid \mathbb{P} \left( \frac{d_\gamma(x(t))}{d_\gamma(x_{\text{init}})} < (2n)^{-\frac{(m-1)}{2}} \right) \geq 1 - 1/n \right\}$  if  $\gamma_{\text{BST}} = 0$  and  $\gamma_j/\gamma_i \geq \sqrt{2n}$  whenever  $\text{cv}_i < \text{cv}_j$ .*

*Proof.* Given a configuration vector  $v^t$  at time  $t$ , if  $v^t \neq x_{\text{end}}$ , we have  $d_\gamma(v^t) \geq \gamma_{\min}$  where  $\gamma_{\min} = \min\{\gamma_i \mid i \neq \text{BST}\}$ . Further, using the relation  $\gamma_j/\gamma_i \geq \sqrt{2n}$ , we have  $d_\gamma(x_{\text{init}}) = \|\gamma\|_2 \leq \sqrt{\sum_{i=0}^{m-2} (2n)^i + (n-m+1)(2n)^{m-1}} \leq \sqrt{(2n)^{(m-1)/2}}$ . Therefore, if  $v^t \neq x_{\text{end}}$ , we have  $\frac{d_\gamma(v^t)}{d_\gamma(x_{\text{init}})} \geq \frac{1}{\|\gamma\|_2} \geq (2n)^{-(m-1)/2}$ .

Thus, if  $v^t$  satisfies  $\frac{d_\gamma(v^t)}{d_\gamma(x_{\text{init}})} < (2n)^{-(m-1)/2}$ , it is necessarily the terminal configuration. Since  $T_{\text{whp}}(\text{TTF}) = \inf\{t \mid \mathbb{P}(x(t) = x_{\text{end}}) \geq 1 - 1/n\}$ , we obtain the result.  $\square$

We are now ready to state and prove the main upper bound on the convergence time of TTF,  $T_{\text{whp}}(\text{TTF})$  (Th. 9). For that, we apply the stochastic matrix theory to the stochastic linear system defined above for  $x(t)$ .

Without loss of generality, we assume that  $\text{cv}_2 \leq \text{cv}_3 \leq \dots \leq \text{cv}_n$ . We choose  $\gamma \in \mathbb{R}^n$  by setting  $\gamma_1 = 0$ ,  $\gamma_2 = 1$ , and  $\gamma_{i+1} = \gamma_i$ , if  $\text{cv}_{i+1} = \text{cv}_i$ , and  $\gamma_{i+1} = \gamma_i \sqrt{2n}$ , if  $\text{cv}_{i+1} > \text{cv}_i$ . In particular,  $\gamma_n = (2n)^{(m-1)/2}$ .

**Theorem 9.** *With a non-uniformly random scheduler  $S(P)$ , the convergence time of TTF is at most  $\frac{m \log 2n}{\log \lambda_2(W)^{-1}}$  with high probability, where  $\gamma$  is defined above.  $\Gamma_{i,j} = \gamma_i/\gamma_j$ ,  $\tilde{W} = \sum_{i < j \wedge \text{cv}_i < \text{cv}_j} (P_{i,j} + P_{j,i})W_{ij}^{\Gamma^2} + \sum_{i < j \wedge \text{cv}_i = \text{cv}_j} (P_{i,j} + P_{j,i})I$ ,  $W_{ij}^{\Gamma^2} = I + \Gamma_{i,j}(e_i e_j^T + e_j e_i^T) + (\Gamma_{i,j}^2 - 1)e_j e_j^T$ , and  $\lambda_2(A)$  denotes the modulus of the second largest eigenvalue of matrix  $A$ .*

*Proof.* Firstly, we study the evolution of the vectors  $y(t) = (x(t) - x_{\text{end}}) \circ \gamma$ , which appear in the formulation of  $T_{\text{whp}}(\text{TTF})$  (Th. 8). As  $x_{\text{end}} = (n-1)e_1$  and  $\gamma_1 = 0$ ,  $y(t)$  reduces to  $x(t) \circ \gamma$ . Since  $x(t+1) = W(t)x(t)$ , we obtain

$$y(t+1) = x(t+1) \circ \gamma = (W(t)x(t)) \circ \gamma = W^\Gamma(t)y(t) \quad (3.5)$$

where  $W^\Gamma(t) = W(t) \circ \Gamma$  is the entry-wise product and matrix  $\Gamma \in \mathbb{R}^{n \times n}$  has entries  $\Gamma_{i,j} = \gamma_i/\gamma_j$  when  $j \neq 1$  and  $\Gamma_{i,1} = 0$  for all  $i \in \{1, \dots, n\}$ . Hence we get:

$$\mathbb{E}(y(t+1)^T y(t+1) \mid y(t)) = y(t)^T \mathbb{E}(W^\Gamma(t)^T \cdot W^\Gamma(t))y(t) \quad (3.6)$$

From (3.4), we know that with probability  $P_{i,j} + P_{j,i}$ , matrix  $W^\Gamma(t)^T \cdot W^\Gamma(t)$  is equal to

$$\begin{cases} I + \Gamma_{i,j}(e_i e_j^T + e_j e_i^T) + (\Gamma_{i,j}^2 - 1)e_j e_j^T & \text{if } \text{cv}_i < \text{cv}_j \text{ or } i = 1 \text{ or } j = 1 \\ I & \text{if } \text{cv}_i = \text{cv}_j \end{cases} \quad (3.7)$$

Then, setting  $W_{ij}^{\Gamma^2} = I + \Gamma_{i,j}(e_i e_j^T + e_j e_i^T) + (\Gamma_{i,j}^2 - 1)e_j e_j^T$ , we have:

$$\begin{aligned} \tilde{W} &= \mathbb{E}(W^\Gamma(t)^T \cdot W^\Gamma(t)) \\ &= \sum_{i < j \wedge \text{cv}_i < \text{cv}_j} (P_{i,j} + P_{j,i})W_{ij}^{\Gamma^2} + \sum_{i < j \wedge \text{cv}_i = \text{cv}_j} (P_{i,j} + P_{j,i})I \end{aligned} \quad (3.8)$$

In particular:

$$\tilde{W}_{1,1} = 1 \quad \text{and} \quad \tilde{W}_{i,1} = \tilde{W}_{1,i} = 0 \quad \text{for all } i \in \{1, \dots, n\} \quad (3.9)$$

Since  $W^\Gamma(t)^T \cdot W^\Gamma(t)$  is symmetric and positive semi-definite, so is its expectation  $\tilde{W}$ . Now, we turn to study the properties of the eigenvalues in  $\tilde{W}$ . By (3.9), matrix  $\tilde{W}$  is of the form

$$\tilde{W} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & W' & \\ 0 & & & \end{pmatrix} \quad (3.10)$$

for some  $W' \in \mathbb{R}^{(n-1) \times (n-1)}$ . Denoting the  $k^{\text{th}}$  largest eigenvalue of matrix  $A$  by  $\lambda_k(A)$ , we have  $\lambda_1(W') \leq \|W'\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n W'_{i,j}$  where  $\|\cdot\|_\infty$  denotes the operator norm with respect to the supremum norm on  $\mathbb{R}^{n-1}$ , i.e., the largest 1-norm of rows of the matrix. According to (3.8), the  $i^{\text{th}}$  row sum of  $\tilde{W}$  is

$$\sum_{j=1}^n \tilde{W}_{i,j} = \sum_{w \neq i} \sum_{v \neq w} P_{w,v} + \sum_{j: cv_j = cv_i} (P_{i,j} + P_{j,i}) \quad (3.11)$$

$$+ \sum_{j: cv_j < cv_i} \Gamma_{j,i}^2 (P_{i,j} + P_{j,i}) + \sum_{j: cv_j > cv_i} \Gamma_{i,j} (P_{i,j} + P_{j,i}) \quad (3.12)$$

for all  $i \in \{2, \dots, n\}$ . As  $\Gamma_{i,j} = \frac{\gamma_i}{\gamma_j} < 1$  whenever  $cv_i < cv_j$ , we obtain  $\sum_j W'_{i,j} < 1$  for all  $i$ . By the block decomposition (3.10), we thus have  $\lambda_1(\tilde{W}) = 1$  and

$$\lambda_2(\tilde{W}) = \lambda_1(W') \leq \|W'\|_\infty = \max_i \sum_j W'_{i,j} < 1. \quad (3.13)$$

Now, using the Rayleigh quotient, we have:

$$\mathbb{E} (y(t+1)^T y(t+1) \mid y(t)) \leq \lambda_2(\tilde{W}) \cdot \|y(t)\|_2^2 \quad (3.14)$$

Repeatedly using (3.14), we obtain the bound

$$\mathbb{E} \|y(t)\|_2^2 = \mathbb{E} (y(t)^T y(t)) \leq \lambda_2(\tilde{W})^t \|y(0)\|_2^2. \quad (3.15)$$

Applying Markov’s inequality, we obtain

$$\begin{aligned} \mathbb{P} \left( \frac{d_\gamma(x(t))}{d_\gamma(x_{\text{init}})} \geq (2n)^{\frac{-(m-1)}{2}} \right) &= \mathbb{P} \left( \frac{\|y(t)\|_2^2}{\|y(0)\|_2^2} \geq (2n)^{-(m-1)} \right) \\ &\leq (2n)^{m-1} \frac{\|y(t)\|_2^2}{\|y(0)\|_2^2} \end{aligned} \quad (3.16)$$

$$\leq (2n)^{m-1} \lambda_2(\tilde{W})^t \quad (3.17)$$

Thus, if  $(2n)^{m-1} \lambda_2(\tilde{W})^t \leq 1/n$ , i.e.,  $t \geq \frac{m \log 2n}{\log \lambda_2(\tilde{W})^{-1}}$ , then  $\mathbb{P} \left( \frac{d_\gamma(x(t))}{d_\gamma(x_{\text{init}})} \geq (2n)^{\frac{-(m-1)}{2}} \right) \leq 1/n$ . So, we obtain  $T_{\text{whp}}(\text{TTF}) \leq \frac{m \log 2n}{\log \lambda_2(\tilde{W})^{-1}}$ .  $\square$

Now, we study the performance of TTF with respect to the convergence time in expectation, i.e.  $T_E(\text{TTF})$ .

**Theorem 10.** *The expected convergence time of the TTF protocol is  $O\left(\frac{m \log n}{\log \lambda_2(\tilde{W})^{-1}}\right)$  where  $\tilde{W}$  is the matrix appearing in Theorem 9.*

*Proof.* Let  $T$  be the convergence time and  $T_\theta = \inf \{t \mid \mathbb{P}(x(t) = x_{\text{end}}) > 1 - \theta\} = \inf \{t \mid \mathbb{P}(T \geq t) \leq \theta\}$ . Analogously to (3.17) in Th. 9, we know that

$$T_\theta \leq \frac{\log 2n^{(m-1)\theta^{-1}}}{\log \lambda(\tilde{W})^{-1}} = \frac{(m-1) \log 2n}{\log \lambda(\tilde{W})^{-1}} + \frac{\log(1/\theta)}{\log \lambda(\tilde{W})^{-1}} = A + B \log(1/\theta) . \quad (3.18)$$

Since  $T$  is a non-negative random number, we have

$$\begin{aligned} T_E(\text{TTF}) &= \sum_{t=1}^{\infty} \mathbb{P}(T \geq t) \\ &= \mathbb{P}(T \geq 1) + \mathbb{P}(T \geq 2) + \dots + \mathbb{P}(T \geq T_\theta) + \sum_{t=1+T_\theta}^{\infty} \mathbb{P}(T \geq t) \\ &\leq T_\theta + \sum_{t=1+T_\theta}^{T_{\theta/2}} \mathbb{P}(T \geq t) + \sum_{t=1+T_{\theta/2}}^{T_{\theta/4}} \mathbb{P}(T \geq t) + \dots \\ &\leq T_\theta + T_{\theta/2} \cdot \theta + T_{\theta/4} \cdot \theta/2 + \dots = T_\theta + \sum_{i=1}^{\infty} T_{\theta/2^i} \cdot \frac{\theta}{2^{i-1}} \\ &\leq T_\theta + \sum_{i=1}^{\infty} (A + B \log \frac{2^i}{\theta}) \cdot \frac{\theta}{2^{i-1}} \\ &= T_\theta + A\theta \cdot \sum_{i=1}^{\infty} \frac{1}{2^{i-1}} + B\theta \cdot (\log 2 \cdot \sum_{i=1}^{\infty} \frac{i}{2^{i-1}} + \log \theta^{-1} \sum_{i=1}^{\infty} \frac{1}{2^{i-1}}) \\ &= T_\theta + (A\theta + B\theta \log \theta^{-1}) \cdot \sum_{i=1}^{\infty} \frac{1}{2^{i-1}} + 2B\theta \log 2 \cdot \sum_{i=1}^{\infty} \frac{i}{2^i} \\ &\leq (1 + 2\theta) \cdot A + B \log(1/\theta) + 4B\theta \log 2 \end{aligned}$$

Choosing  $\theta = 1/n$  leads to

$$T_E(\text{TTF}) \leq \frac{(m-1)[(1+2/n) \log 2n] + [(1+2/n) \log n + \log 16/n]}{\log \lambda_2(\tilde{W})^{-1}} .$$

□

### 3.5 Lazy TTF

The strategy of TTF may result in a long execution when an input value is transferred many times before being finally delivered to the base station. These transfers are certainly energy consuming. Then a natural issue is to transform TTF in order



to save energy, while keeping the time complexity as low as possible. The idea is to prevent certain data transfers, for example, when it is more likely to meet soon a faster agent and thus possibly make fewer transfers in overall. We propose a simple protocol based on TTF, called *lazy TTF*. In contrast with TTF, lazy TTF does not necessarily execute the transition resulting from an interaction. It chooses randomly to execute it or not. Formally, during an interaction  $(i, j)$ , with agent  $i$  acting as initiator, TTF is executed with probability  $p_i$ , where  $p \in \mathbb{R}^n$  is a vector of probabilities.

Notice that the choice of executing TTF depends uniquely on the initiator  $i$ . In practical terms, an initiator represents an agent that, by sensing the environment, has detected another agent  $j$ . At this moment  $i$  takes the random decision (with probability  $p_i$ ) whether a TTF transition should be executed and the interaction itself should take place, or not. In the latter case, not only the energy for the eventual data transfer is saved, but also the energy for establishing the interaction.

Observe that when  $p$  is the vector of all ones, lazy TTF behaves as TTF and its energy consumption is the same as for TTF. However, when  $p$  is the vector of all zeros, lazy TTF does not solve the problem of data collection as no value is ever transferred to the base station, but no energy is consumed for transferring of data or establishing interactions.

Depending on  $p$ , time complexities of lazy TTF can be worse than of TTF, given the same scheduler. At the same time, longer executions of lazy TTF may be more energy efficient. Thus, there is a trade-off between time and energy performance depending on the values of  $p$ . We now investigate the choice of  $p$  for obtaining good time-energy trade-off. Firstly, we give upper bounds on the time complexities of lazy TTF. Then, we introduce an optimization problem that takes  $p$  as a variable. Finally, numerical results in Sect. 3.6 demonstrate energy efficiency of lazy TTF, given the optimal  $p$ .

### 3.5.1 Convergence time of lazy TTF

To obtain an upper bound on the convergence time of lazy TTF, we show a particular equivalence of lazy TTF under scheduler  $S(P)$  with TTF under scheduler  $S(P \circ (p \cdot \mathbf{1}^T))$ , where  $\mathbf{1}$  is the vector of all ones and  $\circ$  presents the entry-wise product. This equivalence is on the level of distribution of configurations of the two protocols. Precisely, as we show below, the random vector  $x(t)$  for these two protocols is exactly the same, allowing to use Th. 9 to obtain a time complexity upper bound for lazy TTF.

Let us express  $x(t)$  for lazy TTF in a similar way as we did before for TTF in Sect. 3.4. First,  $P(x(0) = x_{\text{init}}) = 1$  is the same as for TTF. Then,  $x(t+1) = W(t+1)x(t)$  and  $W(t+1)$  can be seen as a *random* matrix such that, with probability  $P_{i,j} \times p_i + P_{j,i} \times p_j$ ,  $W(t+1)$  is as in Eq. 3.4. Notice that  $x(t)$  in case of TTF under  $S(P \circ (p \cdot \mathbf{1}^T))$  is expressed exactly in the same way (Sect. 3.4). Thus, by applying Th. 9 for TTF under  $S(P \circ (p \cdot \mathbf{1}^T))$ , we obtain the upper bound on  $T_{whp}(\text{lazy TTF}(p))$ .

**Theorem 11.** *With a non-uniformly random scheduler  $S(P)$ , the convergence time*

with high probability of lazy TTF is at most  $\frac{m \log 2n}{\log \lambda_2(\tilde{W})^{-1}}$ ,

$$\begin{aligned} \text{where } \tilde{W} = & \sum_{cv_i < cv_j} (P_{i,j}p_i + P_{j,i}p_j)W_{ij}^{\Gamma^2} + \sum_{cv_i < cv_j} (P_{i,j}(1-p_i) + P_{j,i}(1-p_j))I \\ & + \sum_{cv_i = cv_j} (P_{i,j} + P_{j,i})I, \text{ and } W_{ij}^{\Gamma^2} = I + \Gamma_{i,j}(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T) + (\Gamma_{i,j}^2 - 1)\mathbf{e}_j\mathbf{e}_j^T. \end{aligned} \quad (3.19)$$

Then, the upper bound on  $T_E(\text{lazy TTF}(p))$  can be obtained in the same way as in Th. 10.

To summarize, note that, as executions of lazy TTF are equivalent to those of TTF under  $S(P \circ (p \cdot \mathbf{1}^T))$  in the sense explained above, one can imagine that lazy TTF transforms the matrix of interaction probabilities "on the fly" (during executions). It can be also seen as if it transforms the interaction graph itself. Indeed, certain vectors  $p$  may make some pairs of agents to interact with extremely small probability (or not interact at all), thus effectively remove these pairs from the graph. This is illustrated by the numerical results given in Section 3.6.2. Next, we are looking for vectors  $p$ , optimizing an upper bound on the time performance of lazy TTF( $p$ ) to ensure a good time energy trade-off. Equivalently, we are looking for schedulers (matrices  $P$ ) for which the original TTF is efficient in this sense.

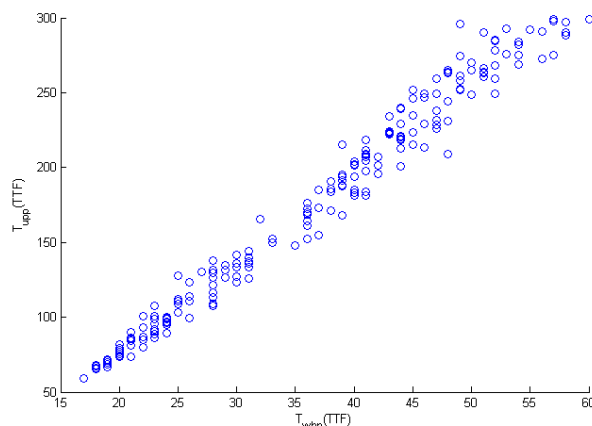
The goal is to find a vector  $p$  minimizing the upper bound on  $T_{whp}(\text{lazy TTF}(p))$  (Th. 11). To that end, an optimization program  $OP_1$ , taking  $p$  as a variable, is proposed as follows:

$$\begin{aligned} OP_1 : & \min_{p \in \mathbb{R}^n} \lambda_2(\tilde{W}) \\ & \text{s.t. } Eq. 3.19 \\ & 0 \leq p_i \leq 1 \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

By Th. 11, minimizing the upper bound of  $T_{whp}(\text{lazy TTF}(p))$  is equivalent to minimizing the second largest eigenvalue of  $\tilde{W}$ . According to Appendix A.2.2,  $OP_1$  can be reformulated as a semi-definite program  $OP_2$  (Appendix A.2), which is convex and can be solved in polynomial time.

$$\begin{aligned} OP_2 : & \min_{p \in \mathbb{R}^n, s} s \\ & \text{s.t. } sI - \tilde{W} \succeq 0 \\ & Eq. 3.19 \\ & 0 \leq p_i \leq 1 \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

Let  $\hat{p}$  be the optimal solution of  $OP_2$ . We can see that if  $\hat{p}$  is all ones vector, lazy TTF( $\hat{p}$ ) performs as TTF. Otherwise, lazy TTF( $\hat{p}$ ) outperforms TTF in terms of the upper bounds on time. This optimized upper bound ensures that lazy TTF( $\hat{p}$ ) converges in a reasonable time. In the next section, by the numerical results obtained for different small examples, we demonstrate the efficiency of lazy TTF( $\hat{p}$ ), in terms of energy consumption.

Figure 3.1: Relation between  $T_{\text{whp}}(\text{TTF})$  and  $T_{\text{upp}}(\text{TTF})$ .

## 3.6 Numerical Results

### 3.6.1 The relation between $T_{\text{whp}}(\text{TTF})$ and its upper bound

The goal of this section is to justify the relevance of the method for obtaining the optimal probability vector  $p$  for lazy TTF. To justify this, we show by simulation that the time upper bound value for TTF is well correlated with the exact value of its time complexity (calculated by Markov chains, for small systems). This implies the same correlation for lazy TTF, because the bounds in Th. 9 and Th. 11 are obviously well correlated too (one is obtained from the other; see Sect. 3.5). That is why the optimal probability vector  $p$  for the upper bound of lazy TTF is close to the optimal vector for the real (tight) convergence time.

From Th. 9, we have an upper bound on time w.h.p. for TTF, denoted here by  $T_{\text{upp}}(\text{TTF})$ . In this section, we show the relation between  $T_{\text{upp}}(\text{TTF})$  and  $T_{\text{whp}}(\text{TTF})$ . In our experiment, two systems of size 4 and 5 are considered and 100 schedulers are generated randomly for each system. Since the system is of small size, for each scheduler  $s$ , the exact value of  $T_{\text{whp}}^s(\text{TTF})$  can be obtained by constructing the corresponding Markov Chain. The upper bound,  $T_{\text{upp}}^s(\text{TTF})$ , can be calculated by Th. 9. Then, for every generated  $s$ , we plot  $T_{\text{whp}}^s(\text{TTF})$  and  $T_{\text{upp}}^s(\text{TTF})$  on the figure with  $x$ -axis for  $T_{\text{whp}}(\text{TTF})$  and  $y$ -axis for  $T_{\text{upp}}(\text{TTF})$ .

From Fig. 3.1, we can see that  $T_{\text{upp}}(\text{TTF})$  has a nearly linear relation with  $T_{\text{whp}}(\text{TTF})$ . It means that  $T_{\text{upp}}(\text{TTF})$  in Th. 9 captures well the relation of the scheduler's behavior to the time performance of TTF in most of the cases. Moreover, it demonstrates that, for lazy TTF, minimizing  $T_{\text{whp}}(\text{lazy TTF}(p))$  in Sect. 3.5 is reasonable for improving the energy performance.

### 3.6.2 TTF vs. Lazy TTF( $\hat{p}$ ) in terms of time complexity bounds

The comparison is made considering the upper bounds given in Th. 9 and Th. 11. The justification (by numerical experiments) that such a comparison makes sense

appears in Sect. 3.6.1. The main goal of this section is to illustrate the effect of the application of the optimal vector  $\hat{p}$  on TTF and its scheduler, in the sense explained in Sect. 3.5.

The numerical experiments, in this section, are performed as follows. First, we generate randomly the probability matrix  $P$  to simulate a non-uniformly random scheduler  $S(P)$ . Here, we present results for 3 representative schedulers. Second, by solving  $OP_2$  (Sect. 3.5), we get the best  $\hat{p}$  for lazy TTF( $p$ ) and we compare the upper bounds on convergence time w.h.p. of TTF and of lazy TTF( $\hat{p}$ ) (Th. 9 and Th. 11, see Tab. 3.1). At last, we interpret the values of  $\hat{p}$  for a better understanding of lazy TTF (Fig. 3.3).

The systems under consideration are composed of four agents, the base station (BST), the fastest agent  $f$ , the slowest agent  $nf_2$ , and an intermediate non-fastest agent ( $nf_1$ ). Thus,  $cv_f < cv_{nf_1} < cv_{nf_2}$ .

The matrix  $P$ , for each considered  $S(P)$  of the three, is encoded by edge labels in a complete interaction graph corresponding to the considered population. The three schedulers are depicted in Fig. 3.2. In the graph (a), take for instance the label 0.021 between  $f$  and BST. It means that the probability that the next interaction concerns the fastest agent (as initiator) and BST, is 0.021.

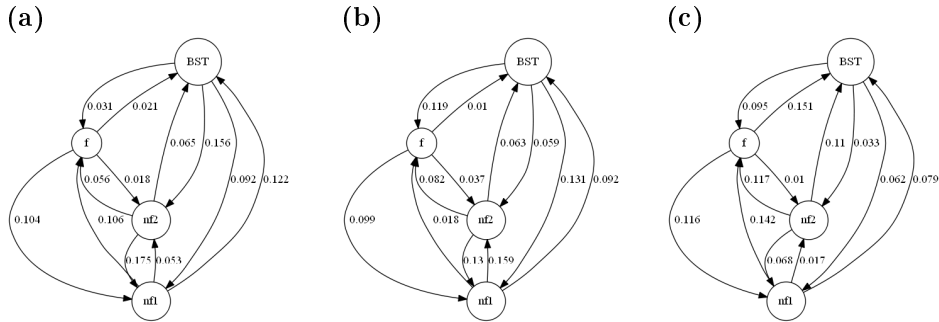


Figure 3.2: Graph illustrations for three random schedulers.

Table 3.1 below presents the numerical results for each scheduler given in Fig. 3.2. The second column gives the vector  $\hat{p}$  in the order  $[BST, f, nf_1, nf_2]$ . The third and fourth columns present the upper bounds on the time w.h.p. for TTF and lazy TTF( $\hat{p}$ ), computed using Th. 9 and Th. 11 respectively and rounded down to integers.

S	$\hat{p}$	TTF	lazy
<b>a</b>	[1, 1, 0, 0]	80	67
<b>b</b>	[1, 0.35, 1, 0]	23	22
<b>c</b>	[1, 1, 1, 1]	13	13

Table 3.1: Time complexity of TTF vs. lazy TTF( $\hat{p}$ ).

Lines **a** and **b** in Table 3.1 show that the tentative to save energy by inhibiting some interactions, e.g.,  $\hat{p}_{nf_2}(\mathbf{a}) = 0$  and  $\hat{p}_{nf_2}(\mathbf{b}) = 0$ , is not really significant for the

execution time bounds (80 vs. 67, 23 vs. 22). In line **c** (in case of scheduler (c), Fig. 3.2) lazy TTF behaves as TTF.

As explained in Sect. 3.5, lazy TTF( $\hat{p}$ ) under  $S(P)$  is equivalent to TTF under  $S(P')$ , where  $P'_{i,j} = P_{i,j} \times \hat{p}_i, \forall (i, j) \in \mathbf{E}$ . Fig. 3.3 presents the interaction graphs encoding  $P'$  for schedulers **(a)** and **(b)**.

Observe that according to graph **(a)** in Fig. 3.2, for both  $nf_1$  and  $nf_2$ , the probability to meet BST is about the same ( $0.092+0.122 = 0.214$  vs.  $0.065+0.156 = 0.221$ ). Moreover, the probability for a token to be transferred to BST from  $nf_1$  or  $nf_2$  through  $f$  is very small. Thus, when meeting  $nf_1$ , a better heuristic for  $nf_2$  is to wait for meeting BST, rather than to transfer tokens to  $nf_1$ . The value obtained for  $\hat{p}$  confirms this heuristic, since  $\hat{p}_{nf_1} = \hat{p}_{nf_2} = 0$  (line **a**, Table 3.1). According to graph **(b)**, Fig. 3.2,  $nf_1$  and  $nf_2$  have a better probability to meet BST than  $f$ . Thus, when  $nf_1$  or  $nf_2$  meets  $f$ , transferring tokens to  $f$  does not seem to be a good choice. This intuition is confirmed by the computed value for  $\hat{p}$ , since  $f$  executes TTF with a small probability 0.35 (line **b**, Table 3.1).

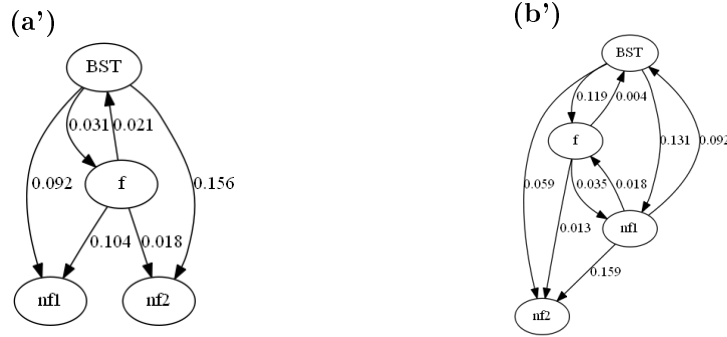


Figure 3.3: Resulting schedulers with  $\hat{p}$ .

### 3.6.3 Gaps on time and energy between TTF and Lazy TTF( $\hat{p}$ )

For the energy consumption analysis, we consider the energy model proposed in Sect. 2.2 for population protocols. In this model, an agent senses its vicinity by proximity sensor, consuming a negligible amount of energy [112]. Once the interaction is established, each participant consumes a fixed amount of energy  $\mathcal{E}_{wkp}$  (mainly for switching on its radio, which is known to be very energy consuming; cf.[111]). Now, recall that, with lazy TTF, the choice of executing TTF depends on the probability  $p_i$  of the initiator  $i$ . If TTF should not be executed, the initiator does not proceed to establish the interaction (i.e.,  $\mathcal{E}_{wkp}$  is not spent), as explained in Sect. 3.5.

We study the expectation of the total energy consumption of a protocol  $\mathcal{P}$ , denoted  $\mathcal{E}(\mathcal{P})$ . According to the energy scheme explained above,  $\mathcal{E}(\mathcal{P})$  is evaluated by the expected total energy spent for establishing all the interactions till convergence. It is proportional to the time expectation  $T_E(\mathcal{P})$ . In particular,  $\mathcal{E}(\text{TTF}) = 2T_E(\text{TTF}) \cdot \mathcal{E}_{wkp}$  and  $\mathcal{E}(\text{lazy TTF}(p)) = 2T_E(\text{lazy TTF}(p)) \times \sum_i \sum_j (P_{i,j} p_i + P_{j,i} p_j) \times \mathcal{E}_{wkp}$ .

For the systems of small size with a scheduler  $s$ , the exact values of  $T_E^s(\text{TTF})$  and  $T_E^s(\text{lazy TTF}(\hat{p}^s))$  can be calculated by constructing the corresponding Markov

Chain. In the experiments, systems of size 4,5,6,7 and 8 are considered and for each size  $n$ , 10000 different schedulers are generated randomly. Denote by  $\mathcal{S}(n)$  the set of these schedulers. For each scheduler  $s \in \mathcal{S}(n)$ ,  $T_E^s(\text{TTF})$ ,  $\hat{p}^s$ ,  $T_E^s(\text{lazy TTF}(\hat{p}^s))$ ,  $\mathcal{E}^s(\text{TTF})$  and  $\mathcal{E}^s(\text{lazy TTF}(\hat{p}^s))$  are evaluated. Then, the gaps on time and on energy between lazy TTF( $\hat{p}^s$ ) and TTF<sup>s</sup> are denoted by  $Gap(T_E, n)$  and  $Gap(\mathcal{E}, n)$ , respectively, and are computed as follows.

$$Gap(T_E, n) = \left( \sum_{s \in \mathcal{S}(n)} \frac{T_E^s(\text{lazy TTF}(\hat{p}^s)) - T_E^s(\text{TTF})}{T_E^s(\text{TTF})} \right) / 10000$$

and

$$Gap(\mathcal{E}, n) = \left( \sum_{s \in \mathcal{S}(n)} \frac{\mathcal{E}^s(\text{lazy TTF}(\hat{p}^s)) - \mathcal{E}^s(\text{TTF})}{\mathcal{E}^s(\text{TTF})} \right) / 10000.$$

Size n	$Gap(T_E, n)$	$Gap(\mathcal{E}, n)$
4	11.60%	-15.32%
5	17.10%	-23.60%
6	22.04%	-30.79%
7	26.31%	-36.99%
8	27.41%	-39.07%

Table 3.2: Gaps on time and energy.

Results appear in Table. 3.2. In column 3, it can be seen that lazy TTF consumes less energy than TTF for all systems. Lazy TTF saves at least 15% of energy. The counterpart is (a slight) increase in the execution time, as shown in column 2.

# An Optimization Model for Data Collection in Wireless Body Area Networks

---

## 4.1 Wireless Body Area Networks

**Power consumption characteristics of WBAN.** WBAN differ from the other WSN considering the specific characteristics of the wireless environment on the human body. First, during the communication, bio-sensors produce heat, which is absorbed by the surrounding tissue and thus increases the temperature of the body. Thus, the transmit power of each bio-sensor is strictly limited. Secondly, for the convenience of wearing and implantation, bio-sensors are tiny and lightweight. Therefore, they have limited energy resources, memory space and computational power. Furthermore, it may be hard to recharge or change the batteries of bio-sensors, specially of those which are implanted into human's body. Thus, a long lifetime is necessarily required for bio-sensors and for WBAN. For example, a glucose monitor would require a lifetime of more than 5 years.

Therefore, unlike the traditional WSN, where the maximum throughput of data and minimal routing overhead are the first requirements on the designs, the energy problem is the first concern and challenge in WBAN. The goal of all designs in every layer of WBAN, is to make the network energy-efficient. For instance, in [55], the authors propose an energy efficient medium access control (MAC) protocol referred to as BodyMAC which uses flexible bandwidth allocation to improve node energy efficiency. Besides, it includes a new efficient sleep mode so as to reduce the idle listening duration.

Nowadays, WBAN has been standardized by IEEE 802.15.6 in physical and MAC layers which provide low complexity, low cost and ultra-low power communication in or around the human body. Readers can refer to surveys [85, 94, 131] for an overview of the development of WBAN in recent years.

This chapter will focus on designing energy efficient data collection protocol for WBAN, using optimization methods. First, a minmax multi-commodity net-flow formulation is presented to schedule the routing of data to the base station, which is a mixed integer linear program (Sect. 4.2). This optimization model minimizes the worst power consumption of each bio-sensor plus the total heating costs produced by the nodes subject to flow conservation and maximum power available constraints for each node. Then a variable neighborhood search (VNS) meta-heuristic procedure is proposed to compute tight near optimal solutions (Sect. 4.3) of this model. Finally,

preliminary numerical results for the VNS approach are provided when compared to the optimal solution of the problem (Sect. 4.4).

This work has been published in ICORES 2014 [4].

## 4.2 Problem Formulation

We model a fixed WBAN by the means of a graph  $G = (V, E)$ , where  $V$  denotes the set of sensor (bio-sensor) nodes and  $E$  is a set of directed arcs. The assumption of directed arcs is valid for WBANs since before any message is transmitted, the route between the source and the destination can be established using Ad-hoc On-demand Distance-Vector routing (AODV) protocols. Without loss of generality, we assume that every node has a fixed initial power capacity  $Cap \in \mathbb{R}^+$ . The set of nodes  $V$  is composed of a subset of source nodes  $V_s$  which sense and collect the data to be transmitted, a set of intermediate transmitters  $V_I$  and a set of sink nodes  $V_t$  where all data is received. For each node  $j \in V$  we define the sets  $\delta^-(j) = \{i \in V : (i, j) \in E\}$  and  $\delta^+(j) = \{i \in V : (j, i) \in E\}$ . We denote by  $C$  the set of commodities to be transmitted, where each commodity  $c \in C$  consists of routing  $D_c$  packets from a source node  $i \in V_s$  to a destination node  $j \in V_t$ . Let  $e_{i,j}$  be the unitary energy needed for transmission of packets on arc  $(i, j) \in E$  and define the total energy consumption of node  $j \in V$  as  $\sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c$  where  $D_c f_{i,j}^c$  is the number of packets of commodity  $c$  transmitted on arc  $(i, j)$ . Note that this amount of energy is computed under the assumption that the transmission energy requirement is negligible compared to the energy required for receiving packets at each node. This is a valid assumption since an extremely low transmit power per node is required in short range ultra-wide band in WBANs, and thus the effort is considerably higher when the nodes are receiving packets [118]. Moreover, this allows significant energy saving when using network coding techniques with the objective of providing reliability under lower-energy constraints [21]. We consider the following multi-commodity netflow formulation denoted hereafter by  $P_0$  as

$$\mathbf{P}_0 : \min_{f,x} \left\{ \sum_{i \in V} a_i x_i + \max_{j \in V} \sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c \right\} \quad (4.1)$$

$$\text{s.t.} \quad \sum_{j \in \delta^+(i)} f_{i,j}^c - \sum_{j \in \delta^-(i)} f_{j,i}^c = b_i^c, \forall i \in V, \forall c \in C \quad (4.2)$$

$$\sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c \leq Cap \times x_j, \forall j \in V \quad (4.3)$$

$$-x_i \leq b_i^c \leq x_i, \forall i \in V, \forall c \in C \quad (4.4)$$

$$f_{i,j}^c \in [0, 1], \forall (i, j) \in E, c \in C \quad (4.5)$$

$$x_i \in \{0, 1\}, \forall i \in V \quad (4.6)$$

where the flow variables  $f_{i,j}^c$  represent the fraction of commodity  $c \in C$  to be transmitted on an arc  $(i, j) \in E$ . The binary variables  $x_i, i \in V$  are used to decide whether node  $i \in V$  will be active when transmitting packets through the network. The objective function in (4.1) is to minimize the total heating costs  $a_i, i \in V$  produced by bio-sensors, which are placed in the body of a patient plus the worst case



power consumption of each active node in the network. The later is a crucial aspect in a WSN since by definition, its lifetime is equal to the minimum lifetime of all nodes in the network [121]. In other words, the network lifetime ends as soon as any node runs out of battery. Let  $b_i^c$  be equal to 1 if node  $i \in V_s$ , or be equal to  $-1$  if node  $i \in V_t$ , and zero otherwise. Constraint (4.2) are flow conservation constraints for each node  $i \in V$  and for each commodity  $c \in C$  while constraint (4.3) imposes the condition that each node has a maximum available power to receive packets in the network. Note that this constraint is forced to be equal to zero when its respective node is set to an inactive state condition. Constraint (4.4) imposes the condition that all sources and sink nodes must always be active, otherwise the network can not sense or relay the collected data toward the base station. Finally, constraints (4.5)-(4.6) are the domain constraints. Note that model  $P_0$  can be easily converted into a mixed integer linear programming (MILP) problem, denoted by  $P_1$ , by introducing an upper bounding variable  $z$  instead of using the max term in its objective function.

We remark that model  $P_1$  provides an optimal routing strategy. However, it does not consider other technical aspects such as broadcasting control flows and organization of the network. The routing strategy is mandatory in WBANs as it allows significant power savings when transmitting sensed data through the network.

$$\begin{aligned}
 \mathbf{P}_1 : \min_{f,x} \quad & z + \sum_{i \in V} a_i x_i \\
 \text{s.t.} \quad & z \geq \sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c, \forall j \in V \\
 & \sum_{j \in \delta^+(i)} f_{i,j}^c - \sum_{j \in \delta^-(i)} f_{j,i}^c = b_i^c, \forall i \in V, \forall c \in C \\
 & \sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c \leq Cap \times x_j, \forall j \in V \\
 & -x_i \leq b_i^c \leq x_i, \forall i \in V, \forall c \in C \\
 & f_{i,j}^c \in [0, 1], \forall (i, j) \in E, c \in C \\
 & x_i \in \{0, 1\}, \forall i \in V
 \end{aligned}$$

In the next section, we introduce a variable neighborhood search meta-heuristic approach to compute near optimal solutions for  $P_1$ .

### 4.3 Near Optimal Solution: Meta-heuristics approach

Meta-heuristics are simple algorithmic procedures commonly used to find near optimal (suboptimal) solutions for combinatorial optimization problems. In practice, they have been proven to be highly effective when solving several hard problems. Especially when the dimension of the problem increases rapidly, which is often the case in real world applications, and when no solver is available to solve these problems to optimality. The most frequently utilized meta-heuristics approaches are genetic algorithms [73], tabu search [62], ant colony system [50], particle swarm op-

timization [82], variable neighborhood search [70], simulated annealing [1], among others. For a more detailed on meta-heuristics, refer to the book [63].

### 4.3.1 Variable neighborhood search

*Variable Neighborhood Search* (VNS) was first proposed by Hansen and Mladenovic and used the idea of neighborhood change during the descent toward local optima while avoiding the valleys that contain them [70]. There are three essential consecutive phases in VNS procedure: *shake*, *local search* and *move*. These three phases are repeated until a certain criteria set by the user is reached. Note that, before the start of VNS, a pre-selected neighborhood structure should be defined according to the optimization problem, i.e., for each solution  $x$ , we define  $N_k(x)$  the set of solutions in the  $k^{th}$  neighborhood of  $x$ .

We give a short explanation here for each phase of VNS. In the shake phase, the algorithm jumps randomly from the current solution  $x$  to another solution  $x'$  in  $N_k(x)$ . Then, a local search starting with  $x'$  (e.g. Greedy algorithm) is launched, and a local optimum solution  $x''$  is returned at the end of the local search phase. During the move phase, the algorithm decides whether  $x''$  is kept as optimal solution and the value of  $k$  (neighborhood structure parameter) for the next interaction.

### 4.3.2 Communication topology

In WBAN, several types of communication topology are proposed and considered in different situations, e.g. star, tree [114], mesh [21, 33]. The most common topology is a star, where the nodes are connected to a central coordinator in star manner [131]. However, the star configuration follows a single hop strategy which is not always the best choice. In [114], the authors discuss about energy efficient topology design for WBANs. They consider a tree network topology and discuss on the energy savings when using single hop and multi hop strategies. They conclude that the distance between nodes plays an important role and that both single hop or multi hop strategies achieve energy savings under different conditions. In [33], authors show that for nodes located far away from the base station, single hop is inefficient with respect to energy consumption and network reliability. Therefore, relay nodes are designed and implemented for delivering data, which constructs a mesh topology in WBAN.

Here, we consider the mesh topology, which is a valid assumption in WBAN as it provides more reliable communications when using cooperative and diversity coding transmission schemes with enhanced throughput [21]. As depicted in Fig. 4.1, bio-sensors are placed in each stage. In particular, stage 1 consists of the source nodes and stage 2 to the final stage consist of intermediate nodes. Finally one node is acting as a sink (base station) to receive all sensed and collected information sent by the source nodes through the network. We denote by  $N_{st}$  the number of nodes placed at stage  $st \in \{1, \dots, ST\}$  where  $ST$  refers to the final stage in Fig. 4.1.

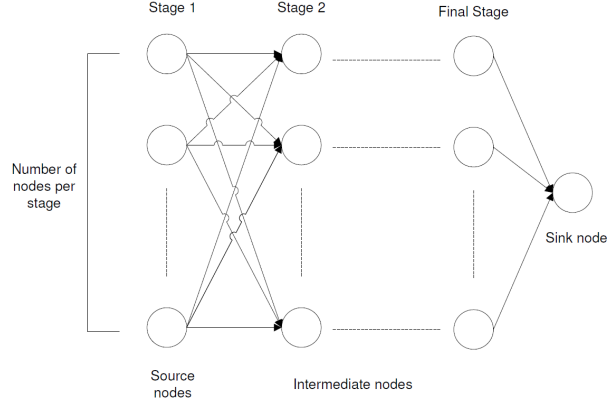


Figure 4.1: WBAN with mesh topology

### 4.3.3 Design of VNS for WBAN problem

In principle, a genetic algorithm or a tabu search approach would serve to compute feasible solutions for our proposed multicommodity flow formulation in a straightforwardly manner. Here, we choose VNS mainly due to its simplicity and low memory requirements. In particular, we adopt a reduced VNS strategy which drops the local search phase of the basic VNS algorithm as it is the most time consuming step [70]. In order to compute feasible solutions for problem  $P_1$  using a VNS approach, we observe that for any fixed assignment of vector  $x$  in  $P_1$ , the problem reduces to solve the following linear programming problem

$$\begin{aligned}
 \bar{P}_1(\mathbf{x}) : \min_f \quad & z \\
 s.t. \quad & z \geq \sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c, \forall j \in V \\
 & \sum_{j \in \delta^+(i)} f_{i,j}^c - \sum_{j \in \delta^-(i)} f_{j,i}^c = b_i^c, \forall i \in V, \forall c \in C \\
 & \sum_{c \in C} \sum_{i \in \delta^-(j)} e_{i,j} D_c f_{i,j}^c \leq Cap \times x_j, \forall j \in V \\
 & f_{i,j}^c \in [0, 1], \forall (i, j) \in E, c \in C
 \end{aligned}$$

There are  $2^{|N|-|N_s|-|N_t|}$  feasible assignments for vector  $x$  in  $P_1$ . It is obvious that some of them are not feasible as they might turn problem  $\bar{P}_1(x)$  infeasible. We propose a VNS approach to compute feasible solutions for  $P_1$  by randomly generating these binary vectors.

We define the neighborhood structure  $N_k(x)$  for  $P_1$  as the set of neighbor solutions  $x'$  in  $P_1$  at a distance "k" from  $x$ , where the distance "k" corresponds to the Hamming distance between the binary vectors  $x$  and  $x'$ .

The VNS approach is presented in Protocol 4. Steps 0-1 give an initial solution to  $P_1$  and Step 2 is the main VNS procedure to obtain a near optimal solution.

---

**Protocol 4** VNS algorithm

---

```

1: Input: a problem instance of  $P_1$ 
2: Output: a feasible solution  $(\tilde{x}, \tilde{f}, \tilde{v})$  for  $P_1$ 
3: Step 0:
4:  $Time \leftarrow 0; k \leftarrow 1;$ 
5:  $count \leftarrow 0; x_i \leftarrow 0, \forall i \in V \setminus V_s \cup V_t;$ 
6:  $x_i \leftarrow 1, \forall i \in V_s \cup V_t;$ 
7: Step 1:
8: for  $st = 2$  to  $St$  do
9:    $r \leftarrow \min(a_i, i \in V_{st})$ 
10:   $x_r \leftarrow 1$ 
11:   $(\bar{x}, \bar{f}, \bar{v}) \leftarrow$  Solve the linear problem  $\bar{P}_1(x)$ .
12:   $\bar{v} \leftarrow \bar{v} + \sum_{i \in V} a_i \bar{x}_i$ 
13:   $(\tilde{x}, \tilde{f}, \tilde{v}) \leftarrow (\bar{x}, \bar{f}, \bar{v})$ 
14: Step 2:
15: while  $Time \leq maxTime$  do
16:   for  $j = 1$  to  $k$  do
17:    choose randomly  $i' \in V_I$ 
18:    if  $x_{i'} = 0$  then
19:       $x_{i'} \leftarrow 1$ 
20:    else
21:       $x_{i'} \leftarrow 0$ 
22:       $(\bar{x}, \bar{f}, \bar{v}) \leftarrow$  Solve the linear problem  $\bar{P}_1(x)$ .
23:       $\bar{v} \leftarrow \bar{v} + \sum_{i \in V} a_i \bar{x}_i$ 
24:      if  $\bar{v} < \tilde{v}$  then
25:         $k \leftarrow 1; (\tilde{x}, \tilde{f}, \tilde{v}) \leftarrow (\bar{x}, \bar{f}, \bar{v})$ 
26:         $Time \leftarrow 0; count \leftarrow 0$ 
27:      else
28:        Keep previous solution
29:         $count \leftarrow count + 1$ 
30:        if  $(k \leq |V_I|)$  and  $(count > \eta)$  then
31:           $k \leftarrow k + 1; count \leftarrow 0$ 

```

---

**Initialization:** We denote by  $(\bar{x}, \bar{f}, \bar{v})$  the solution returned by solving the linear program  $\bar{P}_1(x)$  where  $\bar{v}$  represents the objective function value of  $\bar{P}_1(x)$ . We initialize all the required variables in Step 0. Then in Step 1, we obtain an initial feasible assignment for vector  $x$ , by simply setting one node  $r$  at each stage  $st$  to be active, where  $r = \min\{a_i, i \in N_{st}\}$ . This allows solving  $\bar{P}_1$  and obtaining an initial feasible or infeasible solution  $(\tilde{x}, \tilde{f}, \tilde{v})$  for  $P_1$  that we keep.

**VNS details:** During the execution of the while loop in VNS algorithm, if for any  $x$ , the model  $\bar{P}_1(x)$  is infeasible, then the solution  $x$  is discarded and not considered as a valid solution. In the *shake* phase (line 16-21), VNS is performed by randomly assigning binary values in  $k \leq |N_I|$  positions of vector  $x$  where these positions belong to the set  $N_I$ . From line 24 to line 31, it is the *move* phase of our VNS algorithm. Initially,  $k \leftarrow 1$  while it is increased in one unit, when there is no improvement after a certain number ( $\eta$ ) of solutions have been evaluated. On the other hand, if a new current solution is better than the best found so far, then  $k \leftarrow 1$ , the new solution is recorded and the process goes on. The whole process is repeated until the cpu time variable "Time" is less than or equal to the maximum available "maxTime". Note that we set "Time  $\leftarrow$  0" when a new better solution is found. This gives the possibility to search other "maxTime" units of time with the hope of finding better solutions.

## 4.4 Numerical Results

We first present preliminary numerical results for the proposed VNS approach using only one sample for the input data of the instances. Subsequently, as the proposed model has two conflicting objectives, we provide preliminary numerical comparisons for  $P_1$  while adopting a weighted sum criteria for the objective function of  $P_1$  while adopting a weighted sum criteria for the objective function of  $P_1$  in order to analyze the behavior of the model. Finally, we compute average numerical results.

### 4.4.1 Numerical results for the VNS algorithm

**Parameter settings.** In order to present preliminary numerical results for problem  $P_1$  using the proposed VNS, the input data is randomly generated as follows. The entries in matrix  $(e_{ij})$  are uniformly drawn from  $[0, 1]$  while the heating costs  $a_i, i \in V$  and packets  $D_c, c \in C$  are uniformly distributed in  $[0, 10]$ . The maximum energy available for each node is set equal to  $Cap = 0.4 \times \sum_{i \in V} e_{i,1} \times \sum_{c \in C} D_c / |C|$ . The value of  $\eta$  in the VNS algorithm is calibrated to 20. We set the maximum number of commodities be equal to  $|C| = |V_s|$ , i.e., we assume that each source node can only sense one type of commodity. This is a valid assumption as bio-sensors are usually designed for sensing specialized information in a WBAN. Finally, we set the parameter  $maxTime = 100$ .

**Implementation.** A matlab program is implemented using CPLEX 12 to solve problem  $P_1$ , its linear programming relaxation, and each  $\bar{P}_1(x)$  within each iteration

$N$	# stages	nodes/stage	$P_1$	cpu(s)	$LP_1$	cpu(s)	VNS	cpu(s)	Gap $_{LP}$ %	Gap $_{VNS}^{ini}$ %	Gap $_{VNS}$ %
13	3	4	31.5214	1.8120	24.6043	0.7340	31.5214	0.3910	21.9442	13.9145	0
46	3	15	89.0458	2.5000	81.0329	1.9060	89.0458	67.4370	8.9986	26.4712	0
61	3	20	141.9789	5.7810	125.7967	4.3280	142.3221	1044.1710	11.3976	17.6738	0.2418
17	4	4	53.2150	0.8910	32.0876	0.7500	53.2150	0.0320	39.7019	2.1233	0
61	4	15	99.6237	10.9060	91.0988	3.2190	99.6858	222.8290	8.5571	22.8606	0.0623
21	5	4	51.7760	0.7660	46.6627	0.7340	51.7760	0.0010	9.8757	0	0
41	5	8	73.5483	16.3120	50.3013	1.2030	73.5483	135.1730	31.6078	17.6137	0
51	5	10	63.5951	12.7030	50.8429	1.7970	65.2285	208.0150	20.0522	38.6343	2.5685
33	8	4	56.3391	3.7350	27.9300	0.8280	56.7766	2.1240	50.4251	1.4147	0.7767
49	8	6	60.9010	24.8910	38.4090	1.3440	62.5080	0.1570	36.9320	2.6387	2.6387
41	10	4	55.9875	12.3280	26.8474	0.9220	55.9875	416.3600	52.0474	9.8009	0
61	10	6	83.3481	792.2180	44.5579	1.7500	84.4382	25.8290	46.5400	9.8488	1.3078
49	12	4	76.0640	18.1090	31.3246	1.6250	76.0640	0.2354	58.8180	0	0
73	12	6	83.1798	17204.7970	40.4708	2.2970	84.9211	0.6880	51.3455	4.4978	2.0934
61	15	4	70.8630	31.1720	33.4660	1.0630	70.8630	13.9380	52.7736	0	0
91	15	6	75.5382	2096.2040	42.1599	2.5940	77.6941	0.3265	44.1873	4.6672	2.8540

Table 4.1: Numerical results for the VNS algorithm

of the VNS algorithm. The numerical experiments have been carried out on a Pentium IV, 1 GHz with 2 GoBytes of RAM under windows XP.

In Table. 4.1, column 1 shows the number of nodes considered in each instance. Columns 2 and 3 show the number of stages and number of nodes per stage for each of the instances, respectively. Columns 4 and 5 provide the optimal solution of  $P_1$  and the cpu time in seconds needed by CPLEX, respectively. Columns 6 and 7 provide the optimal solutions for the LP relaxation of  $P_1$  and the cpu time respectively. Similarly, columns 8 and 9 show the best solution found with our VNS algorithm and its cpu time in seconds, respectively. Finally, columns 10, 11 and 12 give the gaps for the LP relaxation, the gaps for the initial solution of VNS and the gaps for the final solution of VNS, respectively. The gaps are computed as  $Gap_{LP} = (\frac{P_1 - LP_1}{P_1}) \times 100$  for the LP case,  $Gap_{VNS}^{ini} = (\frac{VNS_{ini} - P_1}{P_1}) \times 100$  for the initial solution obtained with the VNS algorithm and  $Gap_{VNS} = (\frac{VNS - P_1}{P_1}) \times 100$  for the best solution found with VNS, respectively. The numerical results presented in Table. 4.1 are computed using only one sample for the input data of each instance.

From Table. 4.1, we mainly observe that the gaps obtained with the VNS algorithm are near optimal for all the instances we test, e.g., not larger than 3% from the optimal solution of the problem. Regarding the cpu times, we observe that the VNS approach requires more time when the number of stages is less than the number of nodes per stage. Furthermore, this cpu time is even larger than the cpu time required by CPLEX. This is mainly caused by the fact that the VNS algorithm needs to solve many linear programs in this case. Ultimately, we observe that the gaps obtained when using the initial solutions found with VNS are not very tight which shows somehow the effectiveness of the VNS approach. On the other hand, when the number of stages is larger than the number of nodes per stage, we observe that the VNS algorithm is significantly faster than CPLEX. Moreover, in this case we see that the initial solution found with the proposed algorithm is very tight and in some cases optimal, e.g. this is the case for instances with 49 and 61 nodes. In particular, we see that the cpu time required by CPLEX becomes prohibitive for some of these instances. Finally, we observe that the gaps obtained with the LP relaxation of  $P_1$  are far from the optimal solution of the problem.

$\alpha$	$P_1$	cpu(s)	$LP_1$	cpu(s)	Gap $_{LP}$ %
Inst.1: $N = 13$ , #stages = 3 and nodes/stage = 4					
0	30.8022	0.5780	24.6294	0.4220	20.0400
0.25	24.0651	0.3900	20.1701	0.3600	16.1853
0.5	18.1543	0.4530	14.7102	0.3440	18.9714
0.75	10.5910	0.3900	9.1875	0.3600	13.2519
1	3.5855	0.3750	3.5855	0.5470	0
Inst.2: $N = 41$ , #stages = 5 and nodes/stage = 8					
0	37.9941	1.1410	35.4446	0.8910	6.7103
0.25	34.7115	2.3280	27.6292	0.9530	20.4034
0.5	28.3399	10.2190	18.9793	0.9530	33.0298
0.75	17.7986	21.2970	10.2668	0.9370	42.3172
1	1.5070	1.1570	1.5070	0.8900	0
Inst.3: $N = 49$ , #stages = 8 and nodes/stage = 6					
0	49.8155	1.5630	37.1658	0.9370	25.3931
0.25	41.4989	3.8130	29.4254	0.9690	29.0936
0.5	32.2754	17.5310	21.2694	0.9690	34.1004
0.75	20.4248	15.7180	13.1134	0.9540	35.7968
1	4.9574	1.1560	4.9574	1.1400	0

Table 4.2: Numerical results for weighted objectives

#### 4.4.2 Weighted objective function

In order to explore the behavior of model  $P_1$  when the two conflicting objectives have different degree of importance, we adopt a weighted objective function criteria and write the objective function in  $P_1$  as

$$\min_{f,x} \alpha z + (1 - \alpha) \sum_{i \in V} a_i x_i, \quad (4.7)$$

where  $\alpha \in [0, 1]$ . From a practical point view, the weighted objective function in (4.7) provides an alternative way to handle the trade-off between the total heat generated on patients with more delicate skin versus power energy savings in order to maximize the network lifetime. This would allow to avoid possible hazardous damages on the patients.

In Table. 4.2, we present preliminary numerical results for different values of parameter  $\alpha$  and for three instances having different number of nodes, stages and nodes per stage. More precisely, in column 1 we give the value of  $\alpha$ . In columns 2,3 and 4,5, we present the optimal function value of  $P_1$  (resp.  $LP_1$ ) and their cpu time in seconds needed by CPLEX, respectively. Finally, column 6 shows the gaps for the LP relaxation which are computed exactly as in Table. 4.1. Without loss of generality, the input data is randomly generated exactly as for Table. 4.1 as well.

From Table. 4.2, we mainly observe that the gaps of the LP relaxation goes to zero when the value of  $\alpha \leftarrow 1$ . This means that solving the LP relaxation of  $P_1$ , in this case, suffices to obtain the optimal solution of the problem. On the opposite, when  $0 \leq \alpha < 1$ , the gaps of its relaxation deteriorates considerably which turns the problem more difficult to solve.

#### 4.4.3 Average numerical results

In order to provide more insight regarding our VNS algorithm, we further present average numerical results for the instances presented in Table. 4.1. These results

$N$	# stages	nodes/stage	$P_1$	cpu(s)	$LP_1$	cpu(s)	VNS	cpu(s)	Gap $_{LP}$ %	Gap $_{VNS}^m$ %	Gap $_{VNS}$ %
13	3	4	39.8464	0.4312	32.0752	0.4062	39.8464	0.9370	19.6396	-	0
46	3	15	95.0189	3.4718	80.4871	1.7472	95.7131	110.3312	15.2916	-	0.5546
61	3	20	152.7756	7.9654	129.0012	3.8741	153.9987	477.3697	16.2044	-	1.1142
17	4	4	42.2965	0.4716	32.9802	0.3938	42.2965	9.4488	22.1685	-	0
61	4	15	96.7648	17.4686	80.2268	2.8404	98.8593	252.7788	17.0192	-	2.2219
21	5	4	45.2427	0.6064	32.5676	0.4186	45.2427	26.4560	26.9386	-	0
41	5	8	72.1552	10.3562	52.7956	0.9094	74.8638	36.1282	27.0548	-	3.1426
51	5	10	85.3136	38.3716	65.2315	1.5314	87.1930	42.5282	23.5664	-	2.9688
33	8	4	58.1523	2.5686	31.7155	0.5250	58.7456	10.6092	46.3102	4.0809	1.3087
49	8	6	69.7592	82.4720	46.7561	1.0534	71.0608	17.0594	33.0253	9.5012	1.8658
41	10	4	63.9525	10.1404	37.6166	0.6188	65.2207	7.5086	40.8058	3.7043	2.0128
61	10	6	68.0209	100.3878	40.9167	1.3532	70.3692	47.8566	40.1894	8.8179	3.2827
49	12	4	66.3268	30.0688	29.9523	0.7094	67.0870	0.1336	54.8053	4.1534	1.7767
73	12	6	75.0030	3456.5126	44.3070	1.8002	76.2619	16.2030	40.9237	5.2012	1.9912
61	15	4	72.5088	1441.3000	35.3752	0.9186	72.7524	12.2218	50.7315	1.0480	0.2773
91	15	6	80.5427	2740.1874	44.6374	2.6750	81.0830	20.6280	44.8002	3.2151	0.8021

-: No feasible initial solution found.

Table 4.3: Average numerical results for the VNS algorithm

are presented in Table. 4.3 and the column information is exactly the same as for Table. 4.1. We generate 50 samples for the input data of the instances in row 1 – 13. While for the instances in rows 14 – 16, we use only 10 samples to compute the averages as their cpu times become highly prohibitive. In particular, we arbitrarily set the maximum time for CPLEX to solve these instances be at most 3600 seconds.

From Table. 4.3, we observe similar trends as in Table. 4.1 concerning the gaps obtained with VNS. They are no larger than 3.5% for all the instances tested, compared to the optimal solution of the problem. We also see that the cpu times required by the VNS approach are larger than those required by CPLEX, e.g., the instances in rows 1-9. While for the instances in rows 10-16, CPLEX requires more cpu time. In particular, the instances in rows 14-16 require a huge amount of cpu time using CPLEX while the VNS algorithm finds very tight near optimal solutions with gaps no larger than 2% in less than 25 seconds approximately. Another observation is that the initial solutions obtained with VNS approach are not feasible for instances in rows 1-8, i.e., at least in one of the 50 samples, the initial solution was infeasible. Conversely, finding initial solutions for instances in rows 9-16 is easier. In general, the gaps of initial solutions are not larger than 10%. Finally, the gaps obtained with the LP relaxation of  $P_1$  are not tight, compared to the optimal solution. We note that the LP gaps deteriorate significantly when the number of stages is larger than the number of nodes per stage which is the cases for instances in rows 9-16.

## 4.5 Conclusion

In this chapter, we proposed a minmax multi-commodity netflow formulation to optimally route data packets in a health-care wireless body area network. The aim of the model is to minimize the worst power consumption of each bio-sensor node over the body of a patient plus the total heating costs subject to flow conservation and maximum capacity energy constraints. The model is formulated as a mixed integer linear program. Thus, we proposed a variable neighborhood search procedure to obtain near optimal solutions. Preliminary numerical results indicate that the VNS



approach obtains near optimal solutions with integrality gaps no larger than 3.5%.



# Stochastic Semidefinite Optimization Using Sampling Methods

---

## 5.1 Introduction

It is well known that optimization models are used for decision making as it finds the best solution from all the feasible solutions. In the traditional models, all the parameters are assumed to be known, which conflicts with many real world problems. For instance, in portfolio problems, the return of assets are uncertain [103]. Further, real world problems almost invariably include some unknown parameters, e.g., random demands in power, gas networks, random obstacles in robotics and random precipitation in power production [86, 144]. As a result, the problem is often modeled with random constraints, as shown below:

$$\begin{aligned}
 \min_{x \in \mathbb{R}^n} \quad & f(x) \\
 \text{s.t.} \quad & G(x, \xi) \geq \mathbf{0} \\
 & x \in X,
 \end{aligned} \tag{5.1}$$

where  $\xi \in \mathbb{R}^d$  is a random vector with distribution  $F$  and support  $\Xi$ <sup>1</sup>,  $X$  is a deterministic feasible region,  $\mathbf{0} \in \mathbb{R}^m$  is a vector of zeros and  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $G: \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^m$  are given mapping functions.

In chapter 3, we enhanced population protocols with uniformly random scheduler, where agent  $i$  meets agent  $j$  with (discrete) probability  $P_{i,j}$  during an interaction. However, it happens that the measure of  $P_{i,j}$  in the system may be not precise, i.e., the value of  $P_{i,j}$  is uncertain. Under the assumption of uncertainty, the optimization program  $OP_2$  for the parametrized lazy TTF protocol (3.20), which takes  $P_{i,j}$  as parameter, has random constraints exactly like (5.1).

Since the methods developed for deterministic optimization are not applicable to the model with random constraints, *stochastic programming* is proposed to handle this uncertainty, which combines the concepts of optimization theory with the theory of probability and statistics (See [116] for lectures on stochastic programming).

In this chapter, we focus on *chance constrained programs* [40], one branch of stochastic programming, which is widely applied in financial industry for risk management [89], in engineering for safety requirement [93], in power system management [86, 144].

---

<sup>1</sup>The support of a random vector is the set of values that it can take.

### 5.1.1 Chance constrained programs

To deal with the random constraints (5.1), if we suppose that  $\Xi$  is a finite set, one substitution for (5.1) is  $G(x, \xi) \geq \mathbf{0}$ ,  $\forall \xi \in \Xi$ , which ensures an absolutely robust solution, since every feasible solution of the resulting deterministic program satisfies every possible values of  $\xi \in \Xi$  (worst-case analysis). However, the resulting program is obviously expensive to be solved, whereas feasible solutions may not ever exist. On the other hand, another substitution  $G(x, E[\xi]) \geq \mathbf{0}$ , which takes only the expectation of each parameter into account (average-case analysis), converts the origin program to a deterministic one easy to solve. But its solution may have no robustness against the perturbations of parameters. As a result, another type of constraints, called *chance constraints*, are designed for a balance between the robustness of the solution and the solvability of the program, which was first proposed in [40] to deal with an industrial problem. And the corresponding program is called *chance constrained program* (CCP), defined as follows:

$$\begin{aligned} \text{CCP : } \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \Pr\{G(x, \xi) \geq \mathbf{0}\} \geq 1 - \epsilon \\ & x \in X, \end{aligned} \tag{5.2}$$

where  $\epsilon \in (0, 1]$  is a risk parameter. Usually,  $\epsilon$  is a small value ensuring the constraints to be satisfied with high probability, i.e., the solution has an “acceptable” performance under most realizations of the uncertain parameters.

In real world, many problems can be modeled with constraints like (5.2). For example, in power management system, since wind and solar energy contain many uncertainties to be predicted, depending on the weather and the location, a power plant should be designed to meet energy demand at least to a certain confidence level ( $1 - \epsilon$ ). In financial investment, since market conditions are unpredictable, an investment should be guaranteed to a certain amount of returns at a high confidence level. Moreover, in the design of safety in the system, which requires fault tolerance and reliability, chance constrained program is also useful. Readers can refer to [47, 71, 107] for a background of CCP and some convexity theorems.

### 5.1.2 Methods to solve chance constrained program

In order to circumvent CCP, we usually consider tractable approximation. For instance, convex approximation [98, 99] is a way which analytically generates deterministic convex problems which can be solved efficiently. However, it requires to know the structure of the distribution and structural assumptions on the constraints. Another way is *simulation-based* approach based on Monte-Carlo sampling, for example the well-known *scenario approach* [36, 37, 100].

**Scenario approach.** Scenario approach is easy to be understood, where random parameters are sampled under their distributions, which turns the origin model

(CCP) to a deterministic one, as shown below:

$$\begin{aligned} \text{CCP-SA : } \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & G(x, \xi^i) \geq \mathbf{0}, \forall i \in \{1, \dots, N\} \\ & x \in X, \end{aligned}$$

where  $N$  is the number of samplings and the samplings (scenarios)  $\xi^1, \dots, \xi^N$  are assumed to be independently chosen from the distribution  $F$ . In [37], the authors proved that CCP-SA yields a feasible solution to CCP with probability at least  $1 - \beta$ , given that

$$N \geq \lceil \frac{2}{\epsilon} \log(\frac{1}{\beta}) + 2n + \frac{2n}{\epsilon} \log(\frac{2}{\epsilon}) \rceil, \quad (5.3)$$

supposing  $X$  is a convex set and  $G(x, \xi)$  is a convex function.

**Scenario approach with constraint removal.** However, the solution of CCP-SA is possibly too conservative compared to the optimal solution of CCP. In [103], the authors showed this conservativeness by testing a portfolio selection problem. Thus, to enhance the performance of CCP-SA, in [38], the authors developed a sampling-and-discarding approach which removes some sampling constraints from CCP-SA, and thus results in a less conservative solution. They gave theoretical proofs, showing that when the number of discarded constraints  $k$  satisfies

$$\binom{k}{k+n-1} \sum_{j=0}^{k+n-1} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j} \leq \beta, \quad (5.4)$$

the solution of the reduced model (with optimal constraint removal) remains feasible for CCP and intact with probability at least  $1 - \beta$ .

A greedy algorithm was proposed in [104], to select the constraints to be removed by solving multiple problems of the form CCP-SA. In details, at each iteration  $i$ , an initial program of form CCP-SA with  $N - i + 1$  constraints is solved to determine the set of  $n_i$  candidate constraints to be removed. Next, each candidate constraint is removed at a time and we solve the corresponding CCP-SA with  $N - i$  constraints. After that, the constraint whose removal yields the greatest diminution (in our case) in the objective value, is chosen to be discarded. The greedy algorithm requires solving  $1 + \sum_{i=1}^k n_i$  programs of form CCP-SA. Recent work of [61] presented a precise procedure of this algorithm on control design.

Although the calculation is more expensive here compared with CCP-SA, it allows a less conservative solution.

**Sample average approximation.** Besides scenario approaches, sample average approximation [88] is another simulation-based approach widely studied and applied, in which some of the sampled constraints are allowed to be violated. The model is

shown below.

$$\begin{aligned} \text{CCP-SAA} : \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \frac{1}{N} \sum_{i=1}^N \mathbb{I}(G(x, \xi^i) \geq \mathbf{0}) \geq 1 - \alpha \\ & x \in X, \end{aligned}$$

where  $\alpha \in [0, \epsilon)$  and  $\mathbb{I}(y) : \{\text{True}, \text{False}\} \rightarrow \{0, 1\}$  is the indicator function which equals to one when  $y$  is true, and zero otherwise.

We can see that when  $\alpha = 0$ , CCP-SAA is equivalent to CCP-SA. In [88], the authors proved that supposing  $X$  is a finite set of size less than  $U^n$ , every feasible solution to CCP-SAA will be feasible to CCP with probability at least  $1 - \beta$ , given that  $N \geq \frac{1}{2(\epsilon - \alpha)^2} \log(\frac{1}{\beta}) + \frac{n}{2(\epsilon - \alpha)^2} \log(U)$ . Thus,  $N$  has a quadratic dependence on  $\epsilon - \alpha$ , indicating that  $N$  grows quite large for  $\alpha$  near  $\epsilon$ . The advantage of sample average approximation is that we do not need to solve CCP-SAA to optimality to obtain a solution to CCP and the solution is less conservative than the one of CCP-SA even though more samplings are required.

In [88], the authors conduct experiments on the probabilistic set cover problem and probabilistic transportation problem. Both are modeled in forms of mixed-integer programs. Let  $\Pr\{G(x, \xi) \not\geq \mathbf{0}\}$  be the risk of solution  $x$ . The results show that using  $\alpha > 0$  and a larger sample size in CCP-SAA yields solutions with smaller risks than using  $\alpha = 0$  and a small sample size. More important, optimal solutions obtained in CCP-SAA are less conservative than the optimal solutions of CCP-SA, i.e., smaller final values are obtained in our case. Readers can refer to [105] for applications of CCP-SAA.

### 5.1.3 The problem

In this chapter, we study chance constrained semidefinite program (CCSDP), which will be defined explicitly later in Sect. 5.2. This model has been used to study minimum-volume invariant ellipsoid problem in control theory [43], design problems in RC circuit [151]. Moreover, the example given previously in the introduction considering lazy TTF protocol (the model  $OP_2$ ), if we look into its formulation taking matrix  $P$  as random parameters, it can also be modeled as a CCSDP. Thus, we are interested in this chapter to understand how to solve CCSDP efficiently. For the related work and applications of CCSDP, readers can refer to [20, 145, 151].

### 5.1.4 Overview

This chapter is organized as follows. In Sect. 5.2, the mathematical formulation of CCSDP is presented. In Sect. 5.3, scenario approaches applied to CCSDP are presented and our novel simulation-based method is introduced. To illustrate its strength, in Sect. 5.4, a well known CCSDP problem in control theory is studied and numerical experiments are conducted to compare our results with the state-of-the-art. In the end, a conclusion is given.

This work has been published in ICORES 2015 [142, 143].

## 5.2 Chance Constrained Semidefinite Programs

To study chance constrained semidefinite programs, we start by a more generalized model, called *conic program with chance constraints* (CCC) <sup>2</sup>:

$$(CCC): \min\{f(x) : Pr\{G(x, \xi) \in K\} \geq 1 - \epsilon, x \in X\},$$

where  $x \in \mathbb{R}^n$  is a vector of decision variables,  $X$  is a deterministic feasible region,  $\xi \in \mathbb{R}^d$  is a random vector supported by a distribution  $\Xi \subseteq \mathbb{R}^d$ ,  $K \subset \mathbb{R}^l$  is a closed convex cone <sup>3</sup>,  $G : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^l$  is a random vector-valued function and  $\epsilon$  is a risk parameter given by a decision maker.

Setting  $K$  to a positive semidefinite cone which is a closed convex cone (See Prop. 2 in Appendix), we have chance constrained semidefinite program (CCSDP). Formally,

$$(CCSDP) \min\{f(x)_{x \in X} : Pr\{F(x, \xi) \succeq 0\} \geq 1 - \epsilon\},$$

where  $F(x, \xi) \succeq 0$  is a linear matrix inequality (See Appendix A.2).

## 5.3 Simulation-based Approximation

To solve CCSDP, the classical simulation-based methods presented in Sect. 5.1.2 can be applied. Firstly, we show explicitly how to apply scenario approach and sample average approximation to CCSDP (Sect. 5.3.1 and 5.3.2). Then, a novel method which combines these two is introduced in Sect. 5.3.3.

### 5.3.1 Scenario approach

According to Sect. 5.1.2, the approximation of classical scenario approach is:

$$(CCSDP-SA) \min\{f(x)_{x \in X} : F(x, \xi^i) \succeq 0, \forall i = 1, \dots, N\}$$

where  $N$  is the number of samplings and  $\xi^i$  is a random sample.

### 5.3.2 Sample average approximation - Big-M sampling approach

To adopt sample average approximation (Sect. 5.1.2) to CCSDP, we introduce the "big-M" function with integer variables  $y$  to be the indicator function. Then, we have the following tractable approximation:

$$(CCSDP-BM) \begin{aligned} & \min_{x, y} f(x) \\ & s.t. \quad F(x, \xi^i) + y_i MI \succeq 0, \forall i \in 1, \dots, N \\ & \quad \sum_{i=1}^N y_i \leq \epsilon \times N \\ & \quad x \in X, y \in \{0, 1\}^N, \end{aligned}$$

---

<sup>2</sup>A conic optimization problem consists of minimizing a convex function over the intersection of an affine subspace and a convex cone.

<sup>3</sup>A convex cone is a subset of a vector space over an ordered field that is closed under linear combinations with positive coefficients

where  $I$  is an identity matrix and  $M$  is a large constant.

We see that if  $y_i = 1$ , the constraint  $F(x, \xi^i) + y_i MI \succeq 0$  is satisfied for any candidate solution  $x$ , including those  $x \in \{x | F(x, \xi^i) \not\succeq 0, x \in X\}$  discarded by scenario approach (CCSDP-SA). Therefore, CCSDP-BM is less conservative than CCSDP-SA. However, since CCSDP-BM introduces the binary variables, which makes the model NP-hard. It requires a great effort on computation.

### 5.3.3 Combination of Big-M and constraints discarding

To have a balance between the conservativeness of the solution and the computation efforts, a new method combining the sampling average approximation and scenario approach is proposed.

At the first step, we solve a relaxed CCSDP-BM model, where  $y$  is no more an integer vector but a real vector whose values are between zero and one. Then, the resulting relaxed solution  $y$  helps us to select the set of removal constraints from CCSDP-SA. The idea of removing constraints was first proposed in [38] (See Sect. 5.1.2). We suppose that the optimal value of  $y_i \in [0, 1]$  obtained by solving the relaxed CCSDP-BM, indicates the probability of discarding the corresponding constraint. Because when  $y_i$  is close to one, the solution  $x$  obtained in CCSDP-BM violates the constraint  $F(x, \xi^i) \succeq 0$  in CCSDP-SA with high probability. So the constraints  $F(x, \xi^i) \succeq 0$  in CCSDP-SA whose  $y_i$ s are large, can then be discarded, which returns back a reduced CCSDP-SA model. One can use the criterion given in (5.4) to decide the number of constraints to be discarded. At last, we solve this new reduced model.

In a word, we make use of a relaxed sample average approximation for CCSDP to generate a reduced sampling model CCSDP-SA. Using this method, only two convex SDP programs are needed to be solved. We obtain a less conservative solution than the one obtained in CCSDP-SA (Sect. 5.3.1), while it costs less on computation compared with the method in Sect. 5.3.2.

## 5.4 Numerical Experiments

To show the strength of our method proposed to solve CCSDP, we conduct numerical experiments. The problem that we choose to test our method is a minimum-volume invariant ellipsoid problem in control theory [43], whose classical mathematical model is a semidefinite program. Firstly, we present the problem and its mathematical model with random parameters (Sect. 5.4.1). Then, in Sect. 5.4.2, we explicitly show the models obtained by our method and by two other scenario approaches (Sect. 5.1.2): the classical one and the one with greedy constraint removal. In Sect. 5.4.3, the instances and the implementation in our experiments are given. Finally, numerical results are given (Sect. 5.4.4).



### 5.4.1 Control system problem

First of all, we state out the considered problem in control theory and its mathematical model. Consider the following discrete-time controlled dynamical system:

$$\begin{aligned} x(t+1) &= Ax(t) + bu(t), \quad t = 0, 1, \dots \\ x(0) &= \bar{x}, \end{aligned}$$

where  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$  are system specifications,  $t$  is the index of discrete time,  $\bar{x} \in \mathbb{R}^n$  is the initial state and  $u(t) \in [-1, 1]$  is the control at time  $t$ . For such control system, the interest is to characterize the evolution of  $x(t)$  so that its stability on the initial state  $\bar{x}$  can be determined. We say that a control system is *stable*, if there exists a *controlled invariant set*  $\mathcal{S}$  that any state starting from  $\mathcal{S}$  remains inside  $\mathcal{S}$ . Thus, the controlled invariant set defines the safe region for the initial state.

There are two important families of controlled invariant set for control systems: *ellipsoids* and *polyhedral sets*<sup>4</sup>. An ellipsoid is expressed by:

$$E(Z) = \{x \in \mathbb{R}^n : x^T Z x \leq 1\},$$

where  $Z$  is a symmetric positive definite matrix.<sup>5</sup>

For the control system defined above, it has been shown that an invariant ellipsoid  $E(Z)$  exists if and only if  $\|A\| < 1$  and there exists a  $\lambda \geq 0$  such that

$$\begin{bmatrix} 1 - b^T Z b - \lambda & -b^T Z A \\ -A^T Z b & \lambda Z - A^T Z A \end{bmatrix} \succeq 0. \quad [97]$$

Then, a natural optimization problem for this control system is proposed, aiming to find such an invariant ellipsoid with the minimum-volume, i.e., to have  $\arg \min_Z \text{Vol}(E(Z))$  where  $\text{Vol}(E(Z))$  denotes the volume of ellipsoid  $E(Z)$ . Since the above condition is a linear matrix inequality, the corresponding optimization model belongs to a semidefinite program (Appendix A.2). Notice that,  $\text{Vol}(E(Z))$  is proportional to  $(\det Z)^{-1/n}$ , where  $\det Z$  denotes the determinant of matrix  $Z$ .

Till now, we consider a deterministic control system. But in this chapter, we suppose that the system specification  $b$  is corrupted with noises, i.e.,  $b$  is a vector of random parameters. Under this assumption, the original optimization problem is then formulated as a set of CCSDPs [43], which is  $\{\text{CCMVIE}(\lambda), \lambda \in D\}$ :

$$\begin{aligned} \text{CCMVIE}(\lambda) : \quad & \max_{w \in \mathbb{R}, Z \in \mathbb{R}^{n \times n}} w \\ & s.t. \quad w \leq (\det Z)^{1/n} \\ & \Pr \left\{ \begin{bmatrix} 1 - b^T Z b - \lambda & -b^T Z A \\ -A^T Z b & \lambda Z - A^T Z A \end{bmatrix} \succeq 0 \right\} \geq 1 - \epsilon \\ & Z \succeq 0, \end{aligned}$$

where  $D = \{0.00, 0.01, \dots, 0.99, 1.00\}$  is a finite set,  $b$  and  $A$  are system specifications and  $\epsilon \in (0, 1]$  is the risk parameter.

<sup>4</sup>A polyhedral set is  $\{x \in \mathbb{R}^n : Ax \leq b\}$  for a matrix  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ .

<sup>5</sup>A symmetric real matrix  $M$  is said to be positive definite if the scalar  $z^T M z$  is positive for every non-zero column vector  $z$  of  $n$  real numbers.

### 5.4.2 Experiment approaches

In our experiments, three models are tested to solve CCMVIE( $\lambda$ ) respectively. The first one comes from the classical scenario approach explained in Sect. 5.3.1, the second one refers to the scenario approach with greedy constraint removal (Sect. 5.1.2) and the last one is the application of our proposed method (Sect. 5.3.3). In the sequel, we present these three models.

**I: Scenario approach.** According to Sect. 5.3.1, we obtain the following model  $\{\text{CCSC}(\lambda), \lambda \in D\}$  with

$$\begin{aligned} \text{CCSC}(\lambda) \quad & \max \quad w \\ & \text{s.t.} \quad w \leq (\det Z)^{1/n} \\ & \quad \begin{bmatrix} 1 - b_i^T Z b_i - \lambda & -b_i^T Z A \\ -A^T Z b_i & \lambda Z - A^T Z A \end{bmatrix} \succeq 0, \forall i = \{1, \dots, N\} \\ & \quad Z \succeq 0, \end{aligned}$$

where  $N$  is the number of samplings.

**II: Scenario approach with greedy constraints removal.** The greedy algorithm for constraint removal is presented in Sect. 5.1.2. The final solution is obtained by solving the following sample counterpart (SP) of CCMVIE( $\lambda$ ).

$$\begin{aligned} \text{CCSP}(\lambda) : \quad & \max \quad w \\ & \text{s.t.} \quad w \leq (\det Z)^{1/n} \\ & \quad \begin{bmatrix} 1 - b_i^T Z b_i - \lambda & -b_i^T Z A \\ -A^T Z b_i & \lambda Z - A^T Z A \end{bmatrix} \succeq 0, \forall i = \{1, \dots, N\} - \mathcal{A} \\ & \quad Z \succeq 0, \end{aligned}$$

where  $\mathcal{A}$  is the set of the indexes of the  $k$  removed constraints obtained by the greedy procedure.

**III: Our method.** According to Sect. 5.3.3, our method contains two parts. First, we solve the following relaxed "big-M" model CCRBM( $\lambda$ ) and obtain the solution of  $y \in \mathbb{R}^N$ :

$$\begin{aligned} \text{CCRBM}(\lambda) \quad & \max \quad w \\ & \text{s.t.} \quad w \leq (\det Z)^{1/n} \\ & \quad \begin{bmatrix} 1 - b_i^T Z b_i - \lambda & -b_i^T Z A \\ -A^T Z b_i & \lambda Z - A^T Z A \end{bmatrix} + y_i M I \succeq 0, \forall i = 1, \dots, N \\ & \quad \sum_{i=1}^N y_i \leq \epsilon \times N \\ & \quad Z \succeq 0 \\ & \quad 0 \leq y_i \leq 1, \forall i = 1, \dots, N. \end{aligned}$$

Procedure:

---

1. For each  $\lambda \in D$ :
    - (a) Solve CCRBM( $\lambda$ ) and obtain the relaxation values of  $y$ ,
    - (b) Determine the set  $\mathcal{A}$  of removed constraints according to  $y$ ,
    - (c) Solve CCSP( $\lambda$ ), and let  $v(\lambda)$  be the objective value and  $Z(\lambda)$  be the corresponding solution.
  2. Return  $Z(\lambda^*)$  as the optimal solution, where  $\lambda^* = \operatorname{argmax}_{\lambda \in D} v(\lambda)$ .
- 

Figure 5.1: The procedure of our simulation-based method

We sort the elements of  $y$  in descending order and take the first  $k$  indexes into set  $\mathcal{A} = \{i_1, \dots, i_k\}$ . Then, we solve CCSP( $\lambda$ ). The whole procedure of our method for obtaining the finale matrix  $Z$  is shown in Fig. 5.1.

### 5.4.3 Design of the experiments

**Data.** We use the same instances as in [43]. We have two groups of data.

$$\text{Data 1 : } A = \begin{bmatrix} -0.8147 & -0.4163 \\ 0.8167 & -0.1853 \end{bmatrix}, \bar{b} = \begin{bmatrix} 1 \\ 0.7071 \end{bmatrix},$$

$$\epsilon = 0.05, \rho = 0.01, \beta = 0.05$$

$$\text{Data 2 : } A = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0.0028 & 0.0142 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.0825 & -0.4126 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{b} = \begin{bmatrix} 0 \\ 0.0076 \\ 0 \\ -0.1676 \\ 0 \end{bmatrix}, \epsilon = 0.03, \rho = 0.001, \beta = 0.05,$$

where  $\beta$  is a parameter needed to decide the sample size  $N$  and number of removal constraints  $k$ . We assume that  $b_i = \bar{b}_i + \rho \xi_i, \forall i = 1, \dots, n$  where  $\bar{b}_i \in \mathbb{R}$  is the nominal value,  $\rho \geq 0$  is a fixed parameter to control the level of perturbation and  $\xi_i$  is a standard Gaussian random variable.

#### Selecting the Sample Size and the Number of Constraints to be Removed.

As we have shown previously in Sect. 5.1.2, the minimum number of samplings to ensure the feasibility of solution can be calculated by (5.3). For each sample size

$N$ , according to (5.4), the number of constraints to be removed is calculated as following :

$$k = \lfloor \epsilon N - d + 1 - \sqrt{2\epsilon N \ln \frac{(\epsilon N)^{d-1}}{\beta}} \rfloor, \quad (5.5)$$

where  $d$  is the dimension of the matrix  $Z$ .

For data 1, we consider four sample sizes:  $N = 400, 600, 800$  and  $1000$ . For each sample size, in addition to the  $k$  obtained by (5.5), we vary the ratio of  $k/N$  from 0.03 to 0.05 to study the influence of  $k$  on the result.

For data 2, we consider three sample sizes:  $N = 1000, 1200$  and  $1400$ . Similar to data 1, in addition to the  $k$  obtained by (5.5), we vary the ratio of  $k/N$  from 0.02 to 0.03.

### Implementation.

All experiments are run under MATLAB R2012b on a Windows 7 operating system with i7 CPU 2GHz and 4GB of RAM. The computations are performed using CVX 2.1 [64]<sup>6</sup> with semidefinite program solver *SeDuMi* [125].<sup>7</sup>

#### 5.4.4 Numerical results

Table 5.1 and Table 5.2 provide the computational results of Data 1 and Data 2 respectively.  $N$  presents the sampling number.  $k$  is the number of removal constraints and  $k/N$  is the corresponding ratio. We use the average linear size measure for ellipsoid, which is defined as  $ALS(E(Z)) = (\text{Vol}(E(Z))^{1/n})$  [43]. Readers can refer to page 265 in [97] for the motivation of using this measure. The smaller the volume of an ellipsoid is, the smaller its average linear size is. The columns *SC*, *Greedy*, *BMSP* give the average linear size of ellipsoid obtained by scenario approach (I of Sect. 5.4.2), scenario approach with greedy constraint removal (II of Sect. 5.4.2) and our method (III of Sect. 5.4.2), respectively.  $1 - \text{Vio}$  shows the satisfaction rate of each solution estimated under 100000 simulated random samples. *Gap* presents the gap between the solution of the current method and the solution of the scenario approach.

Table 5.3 shows the CPU time expressed in seconds. The columns *SC*, *Greedy*, *BMSP* show the average CPU time of all tests in Table 5.1 and Table 5.2 when applying scenario approach, scenario approach with greedy constraint removal and our method respectively.

We observe that the real violation is significantly below 5% and 3% respectively in Tables 5.1 and 5.2. It is easy to see that as  $k$  increases, we obtain a better solution both with scenario approach with greedy constraint removal and with our method; and the violation of the solution is larger. The reason is that more constraints we remove, the larger feasible set of  $\text{CCSP}(\lambda)$  we obtain, which involves more violated

<sup>6</sup>cvx is a modeling system for disciplined convex programming.

<sup>7</sup>SeDuMi is an add-on for MATLAB, which solves optimization problems with linear, quadratic and semidefinite constraints.

Table 5.1: Results for Data 1 with  $\epsilon = 0.05, \rho = 0.01, \beta = 0.05$ 

N	k	k/N	SC	1-Vio	Greedy	1-Vio	Gap(%)	BMSP	1-Vio	Gap(%)
400	-		4.1348	0.9988	-	-	-	-	-	-
	3	0.008			4.1328	0.9988	0.5	4.1234	0.9948	2.7
	12	0.030			4.1309	0.9992	0.9	4.1090	0.9902	6.3
	16	0.040			4.1190	0.9928	3.8	4.1065	0.9842	6.9
	20	0.050			4.1148	0.9818	4.8	4.0988	0.9767	8.8
600	-		4.1438	0.9988	-	-	-	-	-	-
	9	0.015			4.1098	0.9884	8.2	4.1095	0.9892	8.3
	18	0.030			4.1060	0.9829	9.1	4.1025	0.9811	10.0
	24	0.040			4.1050	0.9835	9.3	4.0976	0.9744	11.2
	30	0.050			4.1043	0.9799	9.5	4.0962	0.9720	11.5
800	-		4.1482	0.9998	-	-	-	-	-	-
	15	0.019			4.1151	0.9891	7.9	4.1138	0.9923	8.2
	24	0.030			4.1106	0.9917	9.0	4.1066	0.9859	10.0
	32	0.040			4.1083	0.9883	9.6	4.1028	0.9781	10.9
	40	0.050			4.1047	0.9846	10.4	4.0990	0.9776	11.8
1000	-		4.1455	0.9994	-	-	-	-	-	-
	22	0.022			4.1228	0.9968	5.4	4.1124	0.9889	7.9
	30	0.030			4.1221	0.9938	5.6	4.1066	0.9865	9.4
	40	0.040			4.1144	0.9916	7.5	4.1027	0.9791	10.3
	50	0.050			4.1050	0.9861	9.7	4.0974	0.9734	11.6

Table 5.2: Results for Data 2 with  $\epsilon = 0.03, \rho = 0.001, \beta = 0.05$ 

N	k	k/N	SC	1-Vio	Greedy	1-Vio	Gap(%)	BMSP	1-Vio	Gap(%)
1000	-		0.0689	0.9995	-	-	-	-	-	-
	14	0.014			0.0634	0.9980	7.9	0.0631	0.9966	8.5
	20	0.020			0.0615	0.9958	10.7	0.0613	0.9920	11.1
	30	0.030			0.0603	0.9908	12.5	0.0603	0.9915	12.4
1200	-		0.0677	0.9994	-	-	-	-	-	-
	6	0.013			0.0631	0.9933	6.7	0.0629	0.9970	7.1
	24	0.020			0.0611	0.9925	9.7	0.0612	0.9917	9.6
	36	0.030			0.0592	0.9877	12.5	0.0596	0.9890	12.0
1400	-		0.0664	0.9992	-	-	-	-	-	-
	17	0.012			0.0617	0.9958	7.1	0.0615	0.9943	7.3
	28	0.020			0.0603	0.9943	9.2	0.0605	0.9933	8.9
	42	0.030			0.0592	0.9868	10.8	0.0596	0.9927	10.3

Table 5.3: Average CPU time of calculation

	Data 1			Data 2		
	SC	Greedy	BMSP	SC	Greedy	BMSP
CPU time	13.6	201.5	23.3	251.7	4955.2	521.4

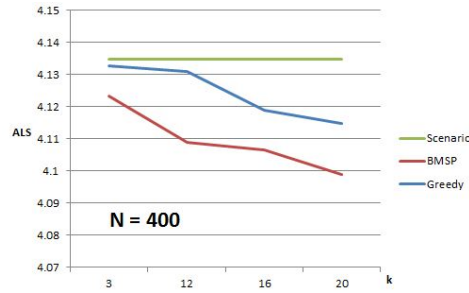


Figure 5.2: Comparison of average linear size for Data 1

elements of  $CCSC(\lambda)$ .

In Table 5.1, for each sampling number  $N$ , our method ( $BMSP$ ) obtains better solution than scenario approach with greedy constraint removal ( $Greedy$ ) with smaller final value (average linear size of ellipsoid) and a larger violation which is below 5%. For  $Greedy$ , the gap is between 0.5%-10.4%, compared with scenario approach. While for our method, the gap is between 2.7%-11.8%. Figure 5.2 gives a precise look on the final value obtained by  $Greedy$  and  $BMSP$  for different values of  $k$  for 400 samples. In Fig.5.3, we compare the violation of  $Greedy$  and  $BMSP$ . We observe that the increasing rate of violation is nearly the same. Figure 5.4 shows the local view of ellipsoid for Data 1 obtained by scenario approach, greedy approach and our method with  $N = 400$  and  $k = 20$ . We can see that the ellipsoid obtained by our method has the smallest volume.

In Table 5.2, we obtain a *Gap* more obvious than the previous one on Data

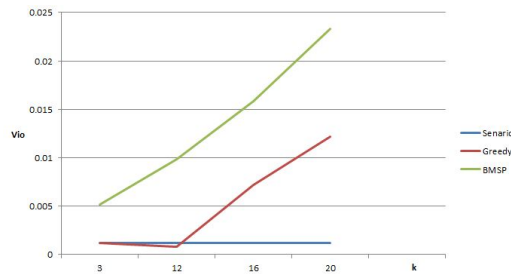


Figure 5.3: Comparison of violation ratio for Data 1

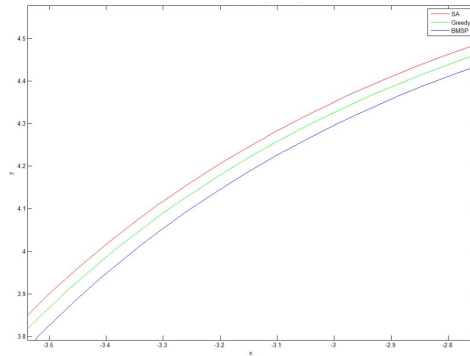


Figure 5.4: Local view of chance-constrained invariant ellipsoid of Data 1 with  $N = 400, k = 20$ .

1. For the case where  $k$  is chosen by (5.5), our method obtains a gap better than *Greedy* method with 0.2% to 0.6% improvement. While for other choices of  $k$ , their *gaps* are very close to each other.

The advantage of our method (*BMSP*) compared with scenario approach with greedy constraint removal (*Greedy*) is on the computing time. In the *Greedy* procedure, we need to solve  $k + 1$  times the semidefinite program  $\text{CCSP}(\lambda)$  in order to decide the removal constraints. While in our method, we only need to solve 2 semidefinite programs. Therefore, we observe from Table 5.3 that *BMSP* consumes much less CPU time than *Greedy* and almost twice CPU time than scenario approach. But as a counterpart of the CPU time, we obtain better solution than scenario approach.

## 5.5 Conclusion

In this chapter, we introduced a new simulation-based method to solve chance constrained semidefinite program. This method is a combination of sample average approximation and scenario approach. We apply this method to semidefinite programming problem in control theory. The numerical results show that our method provides better solutions within a reasonable CPU time.





# Conclusion

---

## 6.1 Summary

To conclude, we summarize the main contributions of this thesis. Firstly, the formal energy model proposed in Sect. 2.2.2 initiated the study on power-aware population protocols. This enhanced model allowed a purely analytical analysis of the energy complexity of a protocol, in the same spirit as for time and space complexity, without appealing to simulations. The worst case (Chapter 2) and the average case (Chapter 3) analyses were performed to obtain general bounds on time and energy. Moreover, it is adaptable to other types of mobile sensor networks, for example, where the mobility pattern of agents is no more “periodic”, but follows other mobility models (like random waypoint model, Brownian model or Rush hour (human) traffic model).

Secondly, we emphasized the importance of using analytical approaches on designing protocols for WSN. By establishing analytical formulas for the energy consumption of data collection protocols, we got information that could hardly be obtained by simulations. For example, we obtained general lower and upper bounds (Sect. 2.5 and 3.3), showing the explicit relations between the protocols’ performances and the characteristics of the network system (e.g., cover times, network size, energy consumption for a communication). Moreover, we obtained the explicit conditions on these characteristics (Sect. 2.4.2) after comparing the analytical bounds.

Thirdly, we showed a promising application of mathematical optimization in the design of (parametrized) distributed algorithms. When a great number of parameters are involved in the protocol (like the vector  $p$  in lazy-TTF), the resulting analytical bound for the performance turns to be high-dimensional and complex with complicated constraints imposed on these parameters (e.g., Eq. 3.19). Accordingly, having proper parameters with analytical approaches becomes nearly impossible. However, optimization techniques can be adapted for solving this issue. An optimization model can be built, taking parameters as variables and the analytical bound as the objective function, such as  $OP_1$  in Sect. 3.5.

At last, we demonstrated the usefulness of mathematical optimization when designing a (centralized) power-aware protocol for wireless body area networks (Chapter 4). In addition, we enhanced the study to more complicated scenarios, where random factors are involved (Chapter 5), showing that optimization is a general tool widely used in decision makings. Meanwhile, by proposing heuristics (Sect. 4.3 and 5.3.3), we advanced the techniques for solving two complex basic optimization models. These heuristics are adaptable for any realistic problem expressed in the same framework.

## 6.2 Perspectives

The different contributions of this thesis, listed in the paragraphs above, have several natural developments.

At first, our analytical model and approach can be used to study the energy issues for a specific kind of mobile WSNs, where sensors with unpredictable movements communicate only when they are in proximity. Nowadays, many realistic networks, with similar characteristics, become new topics of interest in scientific and industry communities. For example, in Vehicular Ad Hoc Networks (VANETs), vehicles share information and communicate between each other only when physically close, due to the mobility constraints. In Near Field Communication (NFC), two NFC devices, usually smartphones, establish communication within a range of 4 cm, for social networking, mobile payment, etc. Our formal energy model (based on the real-world networks) can be considered as a reference to the future enhancement of analytical approach for such networks. It is simple and easy to adapt to various applications (e.g., to the case where the transmitted data does not fit into one packet or to the case where the energy consumption for transferring data depends on the distance between agents). Moreover, besides the worst-case energy consumption (the maximum energy spent per node) studied in this thesis, other energy metrics (e.g., the total energy consumption, the maximum time until one sensor has no energy) are also analytically measurable, by using this model for power-aware designs. Finally, in addition to the data collection problem, other fundamental problems of distributed computing, such as routing, broadcast, leader election or consensus, can also be studied analytically in terms of energy consumption.

The idea of using optimization tools for analyzing power-aware population protocols can also be applied when designing other (parametrized) distributed algorithms. Moreover, this approach is not limited to energy metrics, but can also be used for time and space metrics, provided that the analytical formulas for the performances are given. Using such formulas, the optimization model can be easily constructed. However, this gives rise to the following technical question, which should be investigated further : Can the resulting optimization model (possibly complex) be solved efficiently and in a fully distributed manner?

Solving optimization problem in a distributed way, referred to as *distributed optimization*, has been proposed in the 90's and studied since. Distributed optimization appears in various fields (e.g., distributed learning, decision makings in sensor networks and vehicle coordination), and is adapted to the scenarios where agents in the network measure data, which cannot be treated or analyzed centrally. One example is a system where nodes can only coordinate their decisions with their neighbors. Each node has its own local cost function known only to itself, and they need to agree on a final solution that minimizes the sum of all costs. For instance, in sensors network, to track the position of an acoustic source, each node measures the received signal's strength (including noise), and thus can guess the position of this source. At the end, they cooperate and agree on a single position which minimizes the sum of deviations from their guesses.

In the distributed optimization field, researchers have focused mostly on the following convex optimization model. Formally, in a network of  $n$  agents, each agent  $i$

has a local variable  $x_i \in \mathbb{R}^m$  and a private convex function  $f_i(x) : \mathbb{R}^m \rightarrow \mathbb{R}$ . Agents cooperate to minimize the average function  $f(x) = 1/n \sum_{i=1}^n f_i(x)$ , such that the local variable  $x_i(t)$  converges to the optimal solution  $x^* = \arg \min_{x \in \mathbb{R}^m} f(x)$ . The sub-gradient method in optimization theory is adapted, such that each agent computes its sub-gradient step locally using only local information, e.g., the asynchronous iterative (consensus) algorithms ([129], [123]) and the incremental algorithms ([109], [96]). The alternating direction method of multipliers (ADMM) [32] was proposed to distributively solve a more general convex optimization model. These methods are developed and widely applied in message passing models or similar ones with shared memory. In most of the cases, the convergence rate of ADMM is very slow.

In population protocols, agents are anonymous and have no knowledge of the size of the system. In this case, the developed classical methods for distributed optimization can not be applied directly to obtain a converged optimal solution. Hence, novel methods are required to adapt distributed optimization to the case of population protocols.



# Bibliography

- [1] Emile Aarts and Jan Korst. Simulated annealing and boltzmann machines. 1988.
- [2] Nedal Ababneh, Nicholas Timmons, Jim Morrison, and David Tracey. Energy-balanced rate assignment and routing protocol for body area networks. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 466–471. IEEE, 2012.
- [3] Pooyan Abouzar, Kaveh Shafiee, David G Michelson, and Victor CM Leung. Action-based scheduling technique for 802.15. 4/zigbee wireless body area networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, pages 2188–2192. IEEE, 2011.
- [4] Pablo Adasme, Abdel Lisser, and Chuan Xu. A multicommodity formulation for routing in healthcare wireless body area networks. In *ICORES 2014- Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems*, pages 409–416, 2014.
- [5] Ilan Adler and Sheldon M. Ross. The coupon subset collection problem. *Journal of Applied Probability*, 38:737–746, 2001.
- [6] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L Rivest. Time-space trade-offs in population protocols. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2560–2579. SIAM, 2017.
- [7] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L. Rivest. Time-space trade-offs in population protocols. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 2560–2579, 2017.
- [8] Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. Accepted to SODA 2018, July 2017.
- [9] Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols, Technical Report 2017.
- [10] Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *International Colloquium on Automata, Languages, and Programming*, pages 479–491. Springer, 2015.
- [11] Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and exact majority in population protocols. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 47–56. ACM, 2015.

- [12] Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.
- [13] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568, 2009.
- [14] Dana Angluin, James Aspnes, Zoe Diamadi, Michael J. Fischer, and Rene Peralta. Computation in networks of passively mobile finite-state sensors. In *Proceedings of the Twenty-third Annual ACM Symposium on Principles of Distributed Computing*, pages 290–299, 2004.
- [15] Dana Angluin, James Aspnes, Zoe Diamadi, Michael J Fischer, and Rene Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.
- [16] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. In *International Symposium on Distributed Computing*, pages 61–75. Springer, 2006.
- [17] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, pages 87–102, 2008.
- [18] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.
- [19] Dana Angluin, James Aspnes, Michael J Fischer, and Hong Jiang. Self-stabilizing population protocols. In *Principles of Distributed Systems*, pages 103–117. Springer, 2005.
- [20] Ari Ariyawansa and Yuntao Zhu. Chance-constrained semidefinite programming. Technical report, DTIC Document, 2000.
- [21] Gabriel E Arrobo and Richard D Gitlin. New approaches to reliable wireless body area networks. In *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.
- [22] James Aspnes, Joffroy Beauquier, Janna Burman, and Devan Sohler. Time and space optimal counting in population protocols. In *OPODIS 2016*, pages 13:1–13:17, 2016.
- [23] Seema Bandyopadhyay and Edward J Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723. IEEE, 2003.

- 
- [24] Joffroy Beauquier, Peva Blanchard, and Janna Burman. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *International Conference On Principles Of Distributed Systems*, pages 38–52. Springer, 2013.
- [25] Joffroy Beauquier, Janna Burman, Simon Clavière, and Devan Sohier. Space-optimal counting in population protocols. In *Distributed Computing - 29th International Symposium, DISC 2015, Tokyo, Japan, October 7-9, 2015, Proceedings*, pages 631–646, 2015.
- [26] Joffroy Beauquier, Janna Burman, Julien Clement, and Shay Kutten. On utilizing speed in networks of mobile agents. In *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 305–314, 2010.
- [27] Joffroy Beauquier, Janna Burman, Shay Kutten, Thomas Nowak, and Chuan Xu. Data Collection in Population Protocols with Non-uniformly Random Scheduler. In *Algorithms for Sensor Systems - 13th ALGOSENSORS, Vienna, Austria*, September 2017.
- [28] Luca Becchetti, Andrea E. F. Clementi, Emanuele Natale, Francesco Pasquale, Prasad Raghavendra, and Luca Trevisan. Friend or foe? population protocols can perform community detection. *CoRR*, abs/1703.05045, 2017.
- [29] Amira Boulmaiz, Djemil Messadeg, Noureddine Doghmane, and Abdelmalik Taleb-Ahmed. Robust acoustic bird recognition for habitat monitoring with wireless sensor networks. *I. J. Speech Technology*, 19(3):631–645, 2016.
- [30] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [31] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.
- [32] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [33] Bart Braem, Benoit Latre, Ingrid Moerman, Chris Blondia, Elisabeth Reusens, Wout Joseph, Luc Martens, and Piet Demeester. The need for cooperation and relaying in short-range high path loss sensor networks. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, pages 566–571. IEEE, 2007.
- [34] Han Cai and Do Young Eun. Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 159–170, 2007.

- [35] Shukai Cai, Taisuke Izumi, and Koichi Wada. How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model. *Theory of Computing Systems*, 50(3):433–445, 2012.
- [36] Giuseppe Calafiore and Marco C Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.
- [37] Giuseppe C Calafiore and Marco C Campi. The scenario approach to robust control design. *Automatic Control, IEEE Transactions on*, 51(5):742–753, 2006.
- [38] Marco C Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.
- [39] Rachel Cardell-Oliver, Mark Kranz, Keith Smettem, and Kevin Mayer. A reactive soil moisture sensor network: Design and field evaluation. *International journal of distributed sensor networks*, 1(2):149–162, 2005.
- [40] Abraham Charnes, William W Cooper, and Gifford H Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.
- [41] Ioannis Chatzigiannakis, Shlomi Dolev, Sándor Fekete, Othon Michail, and Paul Spirakis. On the fairness of probabilistic schedulers for population protocols. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2010.
- [42] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11):976–978, 2005.
- [43] Sin-Shuen Cheung, Anthony Man-Cho So, and Kuncheng Wang. Linear matrix inequalities with stochastically dependent perturbations and applications to chance-constrained semidefinite optimization. *SIAM Journal on Optimization*, 22(4):1394–1430, 2012.
- [44] Yongwook Choi, Gopal Pandurangan, Maleq Khan, and V. S. Anil Kumar. Energy-optimal distributed algorithms for minimum spanning trees. *IEEE Journal on Selected Areas in Communications*, 27(7):1297–1304, 2009.
- [45] Vedat Coskun, Busra Ozdenizci, and Kerem Ok. The survey on near field communication. *Sensors*, 15(6):13348–13405, 2015.
- [46] Rachel Cummings, David Doty, and David Soloveichik. Probability 1 computation with chemical reaction networks. *Natural Computing*, 15(2):245–261, 2016.



- 
- [47] Darinka Dentcheva, András Prékopa, and Andrzej Ruszczynski. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1):55–77, 2000.
- [48] Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):7, 2011.
- [49] Alexandros G Dimakis, Soumya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.
- [50] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.
- [51] David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. In *International Symposium on Distributed Computing*, pages 602–616. Springer, 2015.
- [52] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.
- [53] Mehdi Effatparvar, Mehdi Dehghan, and Amir Masoud Rahmani. A comprehensive survey of energy-aware routing protocols in wireless body area sensor networks. *Journal of medical systems*, 40(9):201, 2016.
- [54] Jocelyne Elias and Ahmed Mehaoua. Energy-aware topology design for wireless body area networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 3409–3410. IEEE, 2012.
- [55] Gengfa Fang and Eryk Dutkiewicz. Bodymac: Energy efficient tdma-based mac protocol for wireless body area networks. In *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pages 1455–1459. IEEE, 2009.
- [56] Marco Ferrante and Monica Saltalamacchia. The coupon collector’s problem. *Materials matemàtics*, 2014(2):35 pp., 2014.
- [57] Michael Fischer and Hong Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *International Conference On Principles Of Distributed Systems*, pages 395–409. Springer, 2006.
- [58] Russ Fish, Montralio Flickinger, and Jay Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In *IEEE INFOCOM*, 2006.
- [59] Philippe Flajolet, Daniele Gardy, and Loÿs Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39(3):207–229, 1992.

- [60] Wei Gao and Guohong Cao. User-centric data dissemination in disruption tolerant networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 3119–3127, 2011.
- [61] Simone Garatti and Marco C Campi. Modulating robustness in control design: Principles and algorithms. *Control Systems, IEEE*, 33(2):36–51, 2013.
- [62] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of Combinatorial Optimization*, pages 3261–3362. Springer, 2013.
- [63] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.
- [64] Michael Grant, Stephen Boyd, and Yinyu Ye. *Cvx: Matlab software for disciplined convex programming*, 2008.
- [65] Lin Gu and John A Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems*, 29(2-3):157–182, 2005.
- [66] Lin Gu, John A Stankovic, et al. Radio-triggered wake-up capability for sensor networks. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 27–37, 2004.
- [67] Rachid. Guerraoui and Eric. Ruppert. Even small birds are unique: Population protocols with identifiers. In *Technical Report CSE-2007-04*. York University, 2007.
- [68] Rachid Guerraoui and Eric Ruppert. Names trump malice: Tiny mobile agents can tolerate byzantine failures. In *ICALP (2)*, pages 484–495, 2009.
- [69] Chunlong Guo, Lizhi Zhong, and Jan M Rabaey. Low power distributed mac for ad hoc sensor radio networks. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2944–2948. IEEE, 2001.
- [70] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.
- [71] René Henrion and Cyrille Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41(2):263–276, 2008.
- [72] Ted Herman, Sriram V. Pemmaraju, Laurence Pilard, and Morten Mjelde. Temporal partition in sensor networks. In *Stabilization, Safety, and Security of Distributed Systems, 9th International Symposium, SSS 2007, Paris, France, November 14-16, 2007, Proceedings*, pages 325–339, 2007.
- [73] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

- 
- [74] Seongik Hong, Injong Rhee, Seong Joon Kim, Kyunghan Lee, and Song Chong. Routing performance analysis of human-driven delay tolerant networks using the truncated levy walk model. In *MobilityModels*, pages 25–32, 2008.
- [75] Reed W Hoyt. Sparnet-spartan data network for real-time physiological status monitoring. Technical report, 2008.
- [76] ChingYao Huang, MeiLing Liu, and ShihHeng Cheng. Wrap: A weighted random value protocol for multiuser wireless body area network. In *Spread Spectrum Techniques and Applications (ISITA), 2010 IEEE 11th International Symposium on*, pages 116–119. IEEE, 2010.
- [77] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.
- [78] Sushant Jain, Rahul C Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3):327–339, 2006.
- [79] Emil Jovanov, Aleksandar Milenkovic, Chris Otto, and Piet C De Groen. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *Journal of NeuroEngineering and rehabilitation*, 2(1):6, 2005.
- [80] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *ACM Sigplan Notices*, 37(10):96–107, 2002.
- [81] Raja Jurdak, Antonio G Ruzzelli, and Gregory MP O’Hare. Radio sleep mode optimization in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 9(7):955–968, 2010.
- [82] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.
- [83] Junaid Ahmed Khan, Hassaan Khaliq Qureshi, and Adnan Iqbal. Energy management in wireless sensor networks: A survey. *Computers & Electrical Engineering*, 41:159–176, 2015.
- [84] Kyung Sup Kwak, MA Ameen, Daehan Kwak, Cheolhyo Lee, and Hyungsoo Lee. A study on proposed iee 802.15 wban mac protocols. In *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pages 834–840. IEEE, 2009.
- [85] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18, 2011.

- [86] Ya-zhou Lei, Wei-sheng Wang, Yong-hua Yin, and Hui-zhu DAI. Wind power penetration limit calculation based on chance constrained programming. *Proceedings of the CSEE*, 5(006), 2002.
- [87] Maarten Lont, Dusan Milosevic, and Arthur van Roermund. *Wake-up receiver based ultra-low-power WBAN*. Springer, 2014.
- [88] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.
- [89] Xin-shun MA, Fu-shuan WEN, and Jian-xin LIU. A chance constrained programming based approach for building optimal bidding strategies for generation companies with risk management. *Power System Technology*, 10(008), 2005.
- [90] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications, WSNA 2002, Atlanta, Georgia, USA, September 28, 2002*, pages 88–97.
- [91] Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless PDA users. volume 9, pages 40–55, 2005.
- [92] Yves Mocquard, Emmanuelle Anceaume, and Bruno Sericola. Optimal proportion computation with population protocols. In *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on*, pages 216–223. IEEE, 2016.
- [93] David R Morgan, J Wayland Eheart, and Albert J Valocchi. Aquifer remediation design under uncertainty using a new chance constrained programming technique. *Water resources research*, 29(3):551–561, 1993.
- [94] Samaneh Movassaghi, Mehran Abolhasan, Justin Lipman, David Smith, and Abbas Jamalipour. Wireless body area networks: A survey. *IEEE Communications Surveys & Tutorials*, 16(3):1658–1686, 2014.
- [95] Toshio Nakata. Coupon collector’s problem with unlike probabilities. Preprint, 2008.
- [96] Angelia Nedic and Dimitri P Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.
- [97] Arkadi Nemirovski. Lectures on modern convex optimization. In *Society for Industrial and Applied Mathematics (SIAM)*. Citeseer, 2001.
- [98] Arkadi Nemirovski. On safe tractable approximations of chance constraints. *European Journal of Operational Research*, 219(3):707–718, 2012.

- 
- [99] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.
- [100] Arkadi Nemirovski and Alexander Shapiro. Scenario approximations of chance constraints. In *Probabilistic and randomized methods for design under uncertainty*, pages 3–47. Springer, 2006.
- [101] Sotiris E. Nikolettseas, Theofanis P. Raptis, and Christoforos Raptopoulos. Interactive wireless charging for energy balance. In *36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016*, pages 262–270, 2016.
- [102] Chris Otto, Aleksandar Milenkovic, Corey Sanders, and Emil Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of mobile multimedia*, 1(4):307–326, 2006.
- [103] Bernardo K Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. Computational study of a chance constrained portfolio selection problem. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009.
- [104] Bernardo K Pagnoncelli, D Reich, and Marco C Campi. Risk-return trade-off with the scenario approach in practice: a case study in portfolio selection. *Journal of Optimization Theory and Applications*, 155(2):707–722, 2012.
- [105] BK Pagnoncelli, Shapiro Ahmed, and A Shapiro. Sample average approximation method for chance constrained programming: theory and applications. *Journal of optimization theory and applications*, 142(2):399–416, 2009.
- [106] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.
- [107] András Prékopa. Probabilistic programming. *Handbooks in operations research and management science*, 10:267–351, 2003.
- [108] Jan M Rabaey, Josie Ammer, Tufan Karalar, Suetfei Li, Brian Otis, Mike Sheets, and Tim Tuan. Picoradios for wireless sensor networks: the next challenge in ultra-low power design. In *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, volume 1, pages 200–201. IEEE, 2002.
- [109] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27. ACM, 2004.
- [110] Ramesh Rajagopalan and Pramod K Varshney. Data aggregation techniques in sensor networks: A survey. 2006.
- [111] Venkatesh Rajendran, Katia Obraczka, and J.J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

- [112] Mohammad Abdur Razzaque and Simon Dobson. Energy-efficient sensing in wireless sensor networks using compressed sensing. *Sensors*, 14(2):2822–2859, 2014.
- [113] Wei Ren, Randal W Beard, et al. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.
- [114] Elisabeth Reusens, Wout Joseph, Benoit Latré, Bart Braem, Günter Vermeeren, Emmeric Tanghe, Luc Martens, Ingrid Moerman, and Chris Blondia. Characterization of on-body communication channel and energy efficient topology design for wireless body area networks. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):933–945, 2009.
- [115] Devavrat Shah et al. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.
- [116] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.
- [117] Gaurav Sharma and Ravi R Mazumdar. Scaling laws for capacity and delay in wireless ad hoc networks with random mobility. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 3869–3873, 2004.
- [118] Xiaomeng Shi, Muriel Médard, and Daniel E Lucani. When both transmitting and receiving energies matter: An application of network coding in wireless body area networks. In *International Conference on Research in Networking*, pages 119–128. Springer, 2011.
- [119] Eugene Shih, Paramvir Bahl, and Michael J Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 160–171. ACM, 2002.
- [120] Victor Shnayder, Bor-rong Chen, Konrad Lorincz, Thaddeus RF Fulford-Jones, and Matt Welsh. Sensor networks for medical care. 2005.
- [121] Suresh Singh, Mike Woo, and Cauligi S Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 181–190, 1998.
- [122] Alan F Smeaton, Dermot Diamond, Philip Kelly, Kieran Moran, King-Tong Lau, Deirdre Morris, Niall Moyna, Noel E O’Connor, and Ke Zhang. Aggregating multiple body sensors for analysis in sports. 2008.
- [123] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772–790, Aug 2011.

- 
- [124] Wolfgang Stadje. The collector's problem with group drawings. *Advances in Applied Probability*, 22(4):866–882, 1990.
- [125] Jos F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.
- [126] Hüseyin Ozgür Tan and Ibrahim Körpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *ACM Sigmod Record*, 32(4):66–71, 2003.
- [127] Qinghui Tang, Naveen Tummala, Sandeep KS Gupta, and Loren Schwiebert. Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue. *IEEE Transactions on Biomedical Engineering*, 52(7):1285–1294, 2005.
- [128] Gerard Tel. *Introduction to Distributed Algorithms (2nd ed.)*. 2000.
- [129] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
- [130] John N Tsitsiklis, Dimitri P Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. In *1984 American Control Conference*, pages 484–489, 1984.
- [131] Sana Ullah, Henry Higgins, Bart Braem, Benoit Latre, Chris Blondia, Ingrid Moerman, Shahnaz Saleem, Ziaur Rahman, and Kyung Sup Kwak. A comprehensive survey of wireless body area networks. *Journal of medical systems*, 36(3):1065–1094, 2012.
- [132] Karen Van Dam, Steve Pitchers, and Mike Barnard. Body area networks: Towards a wearable future. In *Proc. WWRP kick off meeting, Munich, Germany*, pages 6–7, 2001.
- [133] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.
- [134] Aart Van Halteren, Richard Bults, Katarzyna Wac, Nicolai Dokovsky, George Koprnikov, Ing Widya, Dimitri Konstantas, Val Jones, and Rainer Herzog. Wireless body area networks for healthcare: the mobihealth project. *Stud Health Technol Inform*, 108:181–193, 2004.
- [135] Lodewijk FW Van Hoesel and Paul JM Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. 2004.
- [136] Lieven Vandenbergh and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

- [137] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *INFOCOM 2001. Twentieth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 3, pages 1388–1397. IEEE, 2001.
- [138] Thomas Watteyne, Isabelle Augé-Blum, Mischa Dohler, and Dominique Barthel. Anybody: a self-organization protocol for body area networks. In *Proceedings of the ICST 2nd international conference on Body area networks*, page 6. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [139] Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on*, pages 108–120. IEEE, 2005.
- [140] Yanwei Wu, Xiang-Yang Li, YunHao Liu, and Wei Lou. Energy-efficient wake-up scheduling for data collection and aggregation. *Parallel and Distributed Systems, IEEE Transactions on*, 21(2):275–287, 2010.
- [141] Chuan Xu, Janna Burman, and Joffroy Beauquier. Power-aware population protocols. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2067–2074, June 2017.
- [142] Chuan Xu, Jianqiang Cheng, and Abdel Lisser. A sampling method to chance-constrained semidefinite optimization. In *Proceedings of the International Conference on Operations Research and Enterprise Systems*, pages 75–81, 2015.
- [143] Chuan Xu, Jianqiang Cheng, and Abdel Lisser. Stochastic semidefinite optimization using sampling methods. In *International Conference on Operations Research and Enterprise Systems*, pages 93–103. Springer, 2015.
- [144] Ning Yang, CW Yu, Fushuan Wen, and CY Chung. An investigation of reactive power planning based on chance constrained programming. *International Journal of Electrical Power & Energy Systems*, 29(9):650–656, 2007.
- [145] David D Yao, Shuzhong Zhang, and Xun Yu Zhou. *LQ Control without Riccati Equations: Stochastic Systems*. FEW-Econometrie en beslistkunde, 1999.
- [146] Yanjun Yao, Qing Cao, and Athanasios V Vasilakos. Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 23(3):810–823, 2015.
- [147] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493–506, 2004.



- [148] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238. ACM, 2004.
- [149] Rong Zheng, Jennifer C Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 35–45. ACM, 2003.
- [150] Hongzi Zhu, Luoyi Fu, Guangtao Xue, Yanmin Zhu, Minglu Li, and Lionel M Ni. Recognizing exponential inter-contact time in vanets. In *INFOCOM, 2010 Proceedings*, pages 1–5, 2010.
- [151] Yuntao Zhu. *Semidefinite programming under uncertainty*. PhD thesis, Washington State University, 2006.



# Appendix



# Semidefinite Programming

---

## A.1 Positive semi-definite matrix

A symmetric real matrix  $A \in \mathbf{R}^{n \times n}$  is said to be positive semi-definite, denoted by  $A \succeq 0$ , if the scalar  $z^T A z$  is greater than or equal to zero for every non-zero vector  $z \in \mathbf{R}^n$ , where  $z^T$  is the transpose of  $z$ . The set of positive semi-definite matrices is denoted by  $S_+^n$ .

**Property 1.** For any matrix  $A$ , the matrix  $A^T A$  is positive semidefinite.

**Property 2.** The set of positive semidefinite symmetric matrices,  $S_+^n$ , is convex. That is, if  $A$  and  $B$  are positive semi-definite, then for any  $\alpha$  between 0 and 1,  $\alpha A + (1 - \alpha)B$  is also positive semidefinite, i.e., if  $A, B \in S_+^n, \forall \alpha \in [0, 1], \alpha A + (1 - \alpha)B \in S_+^n$ .

**Property 3.** All eigenvalues of a symmetric positive semi-definite matrix are non-negative.

**Lemma 8.** The expectation of random positive semi-definite matrix is positive semi-definite.

*Démonstration.* Let  $A \in \mathbf{R}^{n \times n}$  be a random semi-definite matrix. That is,  $A$  is symmetric and for all  $z \in \mathbf{R}^n$ , we have

$$z^T A z = \sum_{i,j} A_{i,j} z_i z_j \geq 0 . \quad (\text{A.1})$$

The expected matrix  $E A$  is symmetric since  $A_{i,j} = A_{j,i}$  implies  $E A_{i,j} = E A_{j,i}$ . Moreover, for all  $z \in \mathbf{R}^n$ , we have

$$z^T (E A) z = \sum_{i,j} (E A_{i,j}) z_i z_j = E \sum_{i,j} A_{i,j} z_i z_j \geq 0 \quad (\text{A.2})$$

by Equation (A.1). □

## A.2 Semi-definite programming

### A.2.1 Linear matrix inequality

A linear matrix inequality in the variable  $x \in \mathbf{R}^n$  has the form

$$A(x) = A_0 + x_1 A_1 + \dots + x_m A_m \succeq 0,$$

where  $A_0 \in \mathbb{R}^{n \times n}, \dots, A_m \in \mathbb{R}^{n \times n}$  are symmetric matrices.

The semidefinite programming [136] is to minimize a linear function of a vector of variables  $x \in \mathbb{R}^m$  subject to a matrix inequality :

$$\begin{aligned} \min_{x \in \mathbb{R}^m} \quad & c^T x \\ \text{s.t.} \quad & A_0 + \sum_{i=1}^m x_i A_i \succeq 0 \end{aligned}$$

where  $c \in \mathbb{R}^m$  and  $A_0, A_1, \dots, A_m \in \mathbb{R}^{n \times n}$  are symmetric matrices.

A semidefinite program (SDP), defined as above, is a convex optimization, since the objective function and constraint are both convex (Prop. 2). Many programs can be cast into SDP, such as linear programs, convex quadratically constrained quadratic programs and second-order cone programs. And more importantly, SDP can be solvable efficiently both in theory and in practice (applying interior point methods [12]). Thus, SDP has many applications including transitional convex constrained optimization, combinatorial optimization [12] and control theory [30].

### A.2.2 Maximum eigenvalue minimization

Let  $A(x) = A_0 + x_1 A_1 + \dots + x_m A_m$  be a symmetric matrix of size  $n$  depending affinely on  $x \in \mathbb{R}^m$ , where  $\forall i \in \{0, \dots, m\}, A_i = A_i^T \in \mathbb{R}^{n \times n}$ . The problem of minimizing the maximum eigenvalue of  $A(x)$ , i.e.  $\min_{x \in \mathbb{R}^m} \lambda_{max}(A(x))$ , is equivalent to the following SDP :

$$\begin{aligned} \min_{x \in \mathbb{R}^m, s \in \mathbb{R}} \quad & s \\ \text{s.t.} \quad & sI - A(x) \succeq 0 \end{aligned} \tag{A.3}$$

where  $I \in \mathbb{R}^{n \times n}$  is the identity matrix.

# Sythèse en français

---

Le sujet de la thèse consiste à développer les protocoles pour les réseaux de capteurs sans fil, prenant en compte leur consommation d'énergie.

Ce manuscrit contient d'abord l'étude d'une extension du modèle des protocoles de populations, qui représentent des réseaux de capteurs asynchrones, passivement mobiles, limités en ressources et anonymes. Il n'existe pas actuellement de modèle communément accepté pour l'étude analytique de la consommation d'énergie. Les travaux dans ce domaine reposent sur des simulations ou des expérimentations. Pour la première fois (à notre connaissance), un modèle formel de consommation d'énergie est proposé pour les protocoles de populations. Ce modèle est utilisable pour développer des protocoles originaux pour des problèmes classiques et fournit un cadre formel pour étudier la consommation d'énergie en tant que paramètre de complexité.

A titre d'application, nous étudions à la complexité en énergie (dans le pire des cas et en moyenne) pour le problème de collecte de données. Deux protocoles prenant en compte la consommation d'énergie sont proposés. Le premier (EB-TTFM) est déterministe et efficace sous certaines conditions. Nous avons réussi à obtenir une borne inférieure pour le problème de collecte de données qui justifie l'efficacité de EB-TTFM. Le second (lazy TTF) est randomisé. Nous avons considéré un cadre où les interactions entre capteurs se font selon une loi non-uniforme (naturelle) et étudié le temps et l'énergie dépensés dans le contexte de collecte de données. Des techniques de matrices stochastiques sont utilisées pour l'analyse de comportements non-uniformes de capteurs mobiles. Ensuite, nous avons traduit le problème de collecte de données en un problème d'optimisation de temps et de consommation d'énergie. Nous avons pu en déduire ce protocole probabiliste efficace. Des simulations ont été faites pour vérifier son efficacité.

Pour déterminer les valeurs optimales des paramètres, nous faisons appel aux techniques d'optimisation. Nous appliquons aussi ces techniques dans un cadre différent, celui des réseaux de capteurs corporels (WBAN). Une formulation de flux est proposée pour acheminer de manière optimale les paquets de données en minimisant la pire consommation d'énergie. Une procédure de recherche à voisinage variable est développée et les résultats numériques montrent son efficacité. Enfin, nous considérons le problème d'optimisation avec des paramètres aléatoires. Précisément, nous étudions un modèle semi-défini positif sous contrainte en probabilité. Un nouvel algorithme basé sur la simulation est proposé et testé sur un problème réel de théorie du contrôle. Nous montrons que notre méthode permet de trouver une solution moins conservatrice que d'autres approches en un temps de calcul raisonnable.





**Titre :** Conception et analyse de protocoles, pour les réseaux de capteurs sans fil, prenant en compte la consommation d'énergie

**Mots clefs :** réseaux de capteurs mobiles, réseaux de capteurs corporels, protocoles de population, consommation d'énergie, collect de données, programmation semi-définie sous contraintes en probabilités

**Résumé :** Ce manuscrit contient d'abord l'étude d'une extension du modèle des protocoles de populations, qui représentent des réseaux de capteurs asynchrones, passivement mobiles, limités en ressources et anonymes. Pour la première fois (à notre connaissance), un modèle formel de consommation d'énergie est proposé pour les protocoles de populations. A titre d'application, nous étudions à la complexité en énergie (dans le pire des cas et en moyenne) pour le problème de collecte de données. Deux protocoles prenant en compte la consommation d'énergie sont proposés. Le premier est déterministe et le second randomisé. Pour déterminer les valeurs optimales des paramètres, nous faisons appel aux techniques d'optimisation. Nous appliquons aussi ces techniques dans un cadre différent,

celui des réseaux de capteurs corporels (WBAN). Une formulation de flux est proposée pour acheminer de manière optimale les paquets de données en minimisant la pire consommation d'énergie. Une procédure de recherche à voisinage variable est développée et les résultats numériques montrent son efficacité. Enfin, nous considérons le problème d'optimisation avec des paramètres aléatoires. Précisément, nous étudions un modèle semi-défini positif sous contrainte en probabilité. Un nouvel algorithme basé sur la simulation est proposé et testé sur un problème réel de théorie du contrôle. Nous montrons que notre méthode permet de trouver une solution moins conservatrice que d'autres approches en un temps de calcul raisonnable.

**Title :** Power-Aware Protocols for Wireless Sensor Networks

**Keywords :** wireless mobile sensor networks, wireless body area networks, population protocols, energy consumption, data collection, chance constrained semi-definite programs

**Abstract :** In this thesis, we propose a formal energy model which allows an analytical study of energy consumption, for the first time in the context of population protocols. Population protocols model one special kind of sensor networks where anonymous and uniformly bounded memory sensors move unpredictably and communicate in pairs. To illustrate the power and the usefulness of the proposed energy model, we present formal analyses on time and energy, for the worst and the average cases, for accomplishing the fundamental task of data collection. Two power-aware population protocols, (deterministic) EB-TTFM and (randomized) lazy-TTF, are proposed and studied for two different fairness conditions, respectively. Moreover, to obtain the best parameters in lazy-TTF, we adopt optimization techniques and evaluate the

resulting performance by experiments. Then, we continue the study on optimization for the power-aware data collection problem in wireless body area networks. A minmax multi-commodity netflow formulation is proposed to optimally route data packets by minimizing the worst power consumption. Then, a variable neighborhood search approach is developed and the numerical results show its efficiency. At last, a stochastic optimization model, namely the chance constrained semidefinite programs, is considered for the realistic decision making problems with random parameters. A novel simulation-based algorithm is proposed with experiments on a real control theory problem. We show that our method allows a less conservative solution, than other approaches, within reasonable time.

