CONTROLLED SINK MOBILITY AND WIRELESS SENSOR NETWORK LIFETIME MAXIMIZATION

by

M. Emre Keskin

B.S., Industrial Engineering, Bilkent University, 2004

M.S., Industrial Engineering, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Industrial Engineering

Boğaziçi University

2014

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. İ. Kuban Altınel for his invaluable guidance not only for this thesis but also for life in general. I have benefited much from his wisdom and insights. I will always take him as a role model during my journey towards being an academician. He has brought many distinguished academicians up and I am proud of being one of his pupils. I am thankful to my unofficial cosuperviser Prof. Necati Aras especially for his criticisms towards my models. They would probably be unrealistic without his critics. His attentive instructions during critical times greatly improved the overall quality of the thesis. I owe a debt of gratitude to Prof. Cem Ersoy for his truly invaluable contributions and recommendations which made this study to be a good one for the computer engineering field as well. I am deeply grateful to Prof. Taner Bilgiç, Assoc. Prof. Temel Öncan and Assist. Prof. Yavuz Boğaç Türkoğulları for taking part in my thesis committee.

I wish to express special acknowledgements to my current and former colleagues for the atmosphere they have created. Special thanks go to M. Güray Güler, Alper Döyen, Emre Demirezen, Yasin Ulukuş, Nisa Önsel, and Oylum Şeker.

I owe many thanks to my family. My father Mehmet have always unconditionally supported me materially and morally. My mother Hatice have always been my bosom friend. My father-in-law Nihat and my mother-in-law Nebahat have always been ready for any kind of support. My daughter, my little source of joy Aslı Zeynep cheered me up when I needed and not needed. Most of all, my wife Betül carried heavy burden of this thesis with me. With faithful support and persistent confidence, she has taken the load off my shoulder.

I gratefully acknowledge the financial support of Boğaziçi University Research Fund through project no 5190D, and TÜBİTAK, under PhD Fellowship Program No. 2211.

ABSTRACT

CONTROLLED SINK MOBILITY AND WIRELESS SENSOR NETWORK LIFETIME MAXIMIZATION

The design of a *Wireless sensor network* (WSN) involves the optimal deployment and activity scheduling of the sensors as well as optimal deployment or routing of sinks and optimal routing of data flows. In this thesis, we first attempt to reflect the limited nature of the mobile sink by considering nonzero sink travel times and taking the data accumulated during the sink travel time into account. The total sink travel time is considered as a part of the network lifetime. We provide two mixed integer linear programming (MILP) models that are flexible enough to handle multiple sink tours as well as a hop limited data routing protocol in which data is routed from sensors towards the sink through the shortest paths including at most a predefined number of hops. We propose heuristic procedures for the solution of the MILP models and show the importance of considering nonzero sink travel times by numerical experiments. An extension to these MILP models that possess a framework with multiple mobile sinks is also developed and it is demonstrated that sink travel times can be neglected for multiple sinks. Later on, we develop several MILP models which integrate sensor placement, activity scheduling and data routing issues with the static sink placement or mobile sink routing design issues. The breadth of the integration changes from the integration of the sink routing problem with the data routing problem to the integration of the sensor placement, activity scheduling, sink routing problems with data routing problems. We study the effect of the integration of WSN design issues by comparing the objective value of the models on a large set of randomly generated problem instances. We also devise heuristics and a branch-and-price algorithm for the solution of the proposed MILP models and empirically test their accuracy and efficiency on a large set of test instances.

ÖZET

HAREKETLİ ANA ALICILAR VE KABLOSUZ DUYGAÇ AĞLARININ ÖMRÜNÜN ENBÜYÜKLENMESİ

Kablosuz duygaç ağları (KDA) tasarımı en iyi duygaç yerleştirme ve duygaç etkinlik çizelgelemesinin yanısıra en iyi ana alıcı yerleştirme ya da rotalama ve en iyi veri rotalama problemlerini de içerir. Bu tezde ilk olarak hareketli ana alıcının sınırlı yapısını yansıtmak için ana alıcının yolculuk süresini ve bu sürede biriken bilgiyi göze almaya çalışıyoruz. Toplam ana alıcı yolculuk süresini ağ ömrünün bir parçası olarak ele alıyoruz. Bu amaçla çoklu sayıda ana alıcı devrini ve bilginin duygaçlardan ana alıcıya sınırlı sekme sayısıyla en kısa yol üzerinden aktarıldığı veri rotalama iletişim protokolü uygulamasını olanaklı kılacak esneklikte iki karma tamsayılı doğrusal programlama (KTDP) gösterimi sunuyoruz. KTDP gösterimlerinin çözümü için sezgisel yöntemler öneriyor ve sıfırdan farklı ana alıcı yolculuk süresinin önemini bilgisayısal deneylerle gösteriyoruz. Ayrıca KTDP gösterimlerini çoklu sayıda hareketli ana alıcılı yeni bir KTDP gösterimi ile genişletiyor ve çoklu ana alıcı varlığında ana alıcı yolculuk sürelerinin gözardı edilebilir olduğunu gösteriyoruz. Daha sonra, duygaç yerleştirme, etkinlik çizelgeleme ve bilgi rotalamayı ana alıcı yerleştirme ya da rotalama problemleriyle derin bir biçimde bütünleyen KTDP gösterimleri geliştiriyoruz. Bütünlemenin kapsamı ana alıcı rotalama ve veri rotalama problemlerinin bütünlenmesinden başlamakta, duygaç yerleştirme, etkinlik çizelgeleme, ana alıcı ve veri rotalama problemlerinin bütünlenmesine uzanmaktadır. Bütünlemenin olumlu etkisini gösterimlerin en iyi amaç işlev değerlerini çok sayıda rassal problem örnekleri üzerinden kıyaslayarak gösteriyoruz. Aynı zamanda KTDP gösterimlerinin çözümü için sezgiseller ve bir dal-eder yöntemi geliştiriyor, bunların doğruluk ve etkinliklerini geniş bir deney problemi kümesini kullanarak sınıyoruz.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_j	Column of nonbasic variable x_j
a_t	Passage time from period $t-1$ to period t
a_{irt}	Auxiliary variable replacing $w_t q_{irt}$
b_{irt}	Auxiliary variable replacing $w_t p_{ir}$
B	Sensor placement budget
B_1	Lowest sensor placement budget
B_2	Second lowest sensor placement budget
B_3	Third lowest sensor placement budget
B_4	Fourth lowest sensor placement budget
B_5	Fifth lowest sensor placement budget
B_6	Highest sensor placement budget
B_l	Low sensor placement budget
B_m	Medium sensor placement budget
B_h	High sensor placement budget
c_j	Objective function coefficient of nonbasic variable x_j
c^r	Unit data reception cost
c_k^r	Unit data reception cost of sensor k
cp_{ir}	Shows whether or not the value of p_{ir} is changed
cq_{irt}	Shows whether or not the value of q_{irt} is changed
c^s	Unit sensing and coordination cost
c_{ij}^t	Unit data transmission cost of arc (i,j)
c_{ki}^t	Unit data transmission cost of sensor k while the sink is at point i
d_{ij}	Distance between sink locations i and j
d_k	Coverage requirement of point k
D	An upper bound on the length of a sink tour
E_i	Initial energy of sensor i
E_k	Initial energy of sensor k
E_r	Battery energy of type r sensor

f_{ir}	Cost of placing sensor (i, r)
f_{jik}	Auxiliary variable replacing $x_{ji}n_{kj}$
fp_{ir}	Denotes the location of the feasible sensor (i, r)
fq_{irt}	Denotes the activity of the feasible sensor (i, r) at period t
F_{ki}	Amount of data that sensor k receives while the sink is at point i
h	Amount of data packets generated by a sensor per unit time
h_r	Data generation rate of a type-r sensor
$I_{\mathcal{M}}$	Index set of all extreme points of $\mathcal{M}$
$I_{\mathcal{RM}}$	A subset of $I_{\mathcal{M}}$
$\mathcal{K}$	Set of points to be covered, sensor field
$\mathcal{K}_{ir}$	Set of points covered by sensor (i, r)
l	A positive integer less than or equal to the length of the current tour
l_p	Position of point $p \in \mathcal{N}^{(t)}$ on the tour
L	Best network lifetime obtained by the sequential assignment heuristic
L_2	Network lifetime found by subproblem 2
L_3	Network lifetime found by subproblem 3
LSRP	Model integrating the coverage, activity scheduling, multiple sink location and data routing problems for multiple stationary sinks
M	A very large number
$\mathcal{M}_s$	Set of sinks
$\mathcal{M}$	Set of all feasible solutions of the Lagrangean sub-problem
MCSLRP	Model integrating the coverage, multiple mobile sink and data routing problems
MLSRP	Model integrating the coverage, activity scheduling, multiple sink routing and data routing problems
MSLRP	Model routing multiple mobile sinks and routing the data simultaneously
n	Number of iterations of the weighted Dantzig-Wolfe decomposition algorithm

n_{ki}	Amount of time that passes until the sink is at visit point $i \in \mathcal{N}$ and sensor $k \in \mathcal{S}$ is unserved
N	Number of candidate sink locations
$\mathcal{N}$	Set of candidate sink locations
$\mathcal{N}_i$	Set of sink locations which are directly reachable from sensor i
$\mathcal{N}_k$	Set of sink visit points that can serve sensor $k \in \mathcal{S}$
$\mathcal{N}_{ir}$	Set of sink locations neighboring sensor (i, r)
$\mathcal{N}^{(t)}$	Set of sink visit points that are on the tour at iteration t
$nmax_{ki}$	Upper limit on n_{ki}
NI	Number of improvements of the best upper bound in the Dantzig Wolfe decomposition algorithm
NL	Set of nodes which are not explored yet
$O^{(\phi)}$	Optimal objective value of the restricted model for parameter value ϕ
$\bar{p}$	p value fixed in the branching process
p^*	Optimal p value
p_{ir}	Indicates whether or not sensor (i, r) is deployed
p_{ji}	Auxiliary variable replacing $x_{ji}t_j$
P	Number of sinks
$\mathcal{P}$	Set of p values for which β variables are generated for the restricted master problem
P_{ab}	Probability of correctly receiving a packet through arc (a, b)
$\bar{q}$	q value fixed in the branching process
q^*	Optimal q value
q_{irt}	Indicates whether or not sensor (i, r) is active in period t
$q_{n\ell ts}$	Shows whether or not sink s goes from i to j between periods $t-1$ and t
$\mathcal{Q}$	Set of q values for which β variables are generated for the restricted master problem
$\bar{rc}$	Maximum reduced cost
r_r^c	Communication range of type r sensor
r_r^s	Sensing range of type r sensor

R	Number of sink tours during the lifetime
$R(0,1)$	A real number randomly chosen from $[0,1]$
$\mathcal{R}$	Set of sensor types
sensor (i,r)	Type r sensor deployed at point i
$\mathcal{S}$	Set of (candidate) sensor locations
$\mathcal{S}_i$	Set of sensors neighboring sensor i
$\mathcal{S}_{ir}$	Set of sensor locations neighboring sensor (i,r)
$\mathcal{S}_{ki}$	Set of sensors for which sensor k lies on the shortest path to point i
$\mathcal{S}_\ell$	Set of sensors neighboring the sink location ℓ
$\mathcal{T}$	Set of time periods
T	Number of time periods
T_1	Number of elements in $\mathcal{X}_1$
T_2	Number of elements in $\mathcal{X}_2$
T^*	Lowest possible number of periods leading to the maximum network lifetime
t_i	Total sojourn time at $i \in \mathcal{N}$
$tmin_i$	Lower bound on the total sojourn time at $i \in \mathcal{N}$
$tmax_i$	Upper bound on the total sojourn time at $i \in \mathcal{N}$
$\mathcal{T}$	Set of time periods
u_0	Minimum Lagrangean dual value
u_i	The order of the visit point $i \in \mathcal{N}$ on the sink route
u_k	The number of sensors that can cover point k
$u_{ir\ell t}$	Given upper bound on the flow from a type r sensor located at point i to a sink located at point ℓ in period t
u_{irjst}	Given upper bound on the flow from a type r sensor located at point i to a sensor of type s located at point j
v	Nonzero finite speed of the single sink
v_s	Nonzero finite speed of the sink s
w_t	Length of period t
x_j	A nonbasic variable
x_{ij}	Denotes whether or not arc (i,j) is included in the sink tour

x_{ijt}	Amount of flow from sensor i, to sensor j at period t
x_{irjst}	Amount of flow from sensor (i,r) to sensor (j,s) in period t
$\mathcal{X}_1$	Index set of all β variables
$\mathcal{X}_1^c$	Index set of the generated β variables at the current node
$\mathcal{X}_2$	Index set of all α variables
$\mathcal{X}_2^c$	Index set of the generated α variables at the current node
$y_{ir\ell t}$	Amount of flow from sensor (i,r) to sink deployed at point ℓ in period t
$y_{i\ell t}$	Amount of flow from sensor i to sink at point ℓ at period t
Y	Upper bound on the flows at sensor-to-sink arcs
$\bar{z}$	z value fixed in the branching process
z^*	Optimal z value
z_{ki}	Indicates whether or not sensor located at point $k \in \mathcal{S}$ is served by the sink visit point $i \in \mathcal{N}$
z_ℓ	Indicates whether or not a sink is located at point ℓ
$z_{\ell t}$	Indicates whether or not a sink is located at point ℓ in period t
$z_{\ell ts}$	Indicates whether or not sink s is located at point ℓ at period t
$\mathcal{Z}_{BEST}$	Best sink locations found by the hybrid heuristic
z_{BEST}	Best feasible objective function value found in the branch-and-price algorithm
z_{CM}	Lifetime of the best solution of the conventional model
z_G	Maximum network lifetime obtained by Gurobi
z_{LSRP}	Objective function value of the best solution computed by Gurobi for LSRP
z_{MCSLRP}	Objective function value of the best solution computed by Gurobi for MCSLRP
z_{MLSRP}	Objective function value of the best solution computed by Gurobi for MLSRP
z_{MSLRP}	Objective function value of the best solution computed by Gurobi for MSLRP
$\mathcal{Z}_{MP}$	Minimum possible Lagrangean dual value, which is the optimal value of MP

z_{PIH}	Maximum network lifetime obtained by period iteration heuristic
$\mathcal{Z}_{RMP}$	Optimal objective function value of the restricted master problem
z_{HH}	Maximum network lifetime obtained by hybrid heuristic
z_{S_1}	Optimal objective function value of subproblem 1
z_{S_2}	Optimal objective function value of subproblem 2
z_{SAH}	Maximum network lifetime obtained by sequential assignment heuristic
z_{UHM}	Lifetime of the best solution of the unlimited hop model
$\mathcal{Z}$	Set of p values for which α variables are generated for the restricted master problem
$\hat{\alpha}$	Lagrangean multiplier corresponding to constraint (4.54)
α^{t_2}	Multiplier for variable z^{t_2}
α_{mk}	replaces $100 \times P_{ma}P_{ab}P_{bk}$
α_{irt}	replaces $M(\lambda_{irt} + \beta_{irt} + \gamma^1_{irt} - \gamma^2_{irt}) + \sum_{k \in \mathcal{K}_{ir}} \mu_{kt}$
$\hat{\beta}$	Lagrange multiplier corresponding to constraint (4.53)
$\boldsymbol{\beta}$	Lagrange multiplier corresponding to constraint (5.10)
β^{t_1}	Multiplier for variables (p^{t_1}, q^{t_1})
$\boldsymbol{\beta_{BEST}}$	Best β value
$\boldsymbol{\gamma}$	Lagrange multiplier corresponding to constraint (4.48)
$\boldsymbol{\gamma^1}$	Lagrange multiplier corresponding to constraint (5.12)
$\boldsymbol{\gamma^1_{BEST}}$	Best γ^1 value
$\boldsymbol{\gamma^2}$	Lagrange multiplier corresponding to constraint (5.13)
$\boldsymbol{\gamma^2_{BEST}}$	Best γ^2 value
$\boldsymbol{\eta}$	Lagrange multiplier corresponding to constraint (4.51)
δ	Lagrange multiplier corresponding to constraint (5.7)
δ_{BEST}	Best δ value
ϵ	A very small tolerance, precision
θ	Coefficients of the current multiplier
$\boldsymbol{\theta}$	Lagrange multiplier corresponding to constraint (4.47)

κ_1	Coefficient of distance independent constant part of the transmission energy
κ_2	Coefficient of distance dependent part of the transmission energy
$\boldsymbol{\lambda}$	Lagrange multiplier corresponding to constraint (5.9)
$\boldsymbol{\lambda_{BEST}}$	Best λ value
$\boldsymbol{\mu}$	Lagrange multiplier corresponding to constraint (5.6)
$\boldsymbol{\mu_{BEST}}$	Best μ value
ϕ	Number of periods
ϕ_{BEST}	Best objective function value computed using hybrid heuristic
ϕ_z	Maximum network lifetime obtained by the Lagrangean heuristic in the allowed computation time for the given sink locations z
$\phi_{ITERBEST}$	Maximum network lifetime achieved in the current period iteration loop
$\boldsymbol{\phi}$	Lagrange multiplier corresponding to constraint (4.49)
$\boldsymbol{\sigma}$	Lagrange multiplier corresponding to constraint (4.52)

LIST OF ACRONYMS/ABBREVIATIONS

ASP	Activity Scheduling Problem
BAP	Branch and Price
CP	Coverage Problem
DRP	Data Routing Problem
FM	Feasibility Model
HH	Hybrid Heuristic
IMP	Integer Master Problem
LH	Lagrangean Heuristic
LHM	Limited Hop Model
LB	Lower Bound
LD	Lagrangean Dual
LP	Linear Program
ME	Mobility Effect
MP	Master Problem
MILP	Mixed Integer Linear Program
MS	Multiple Sink
MSM	Multiple Sink Model
NLP	Nonlinear Program
PD	Percent Deviation
PIH	Period Iteration Heuristic
RLP	Restricted Linear Problem
RMP	Restricted Master Problem
S1	Subproblem 1
S2	Subproblem 2
S3	Subproblem 3
SAH	Sequential Assignment Heuristic
SLP	Sink Location Problem
SPE	Sensor Placement Effect

SPSE	Sensor Placement and Scheduling Effect
SRP	Sink Routing Problem
TCH	Tour Construction Heuristic
TIH	Tour Improvement Heuristic
TSP	Traveling Salesman Problem
UB	Upper Bound
UHM	Unlimited Hop Model
WSN	Wireless Sensor Network

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of low-power, low-cost, multi-functional electronic devices called *sensors*, which are deployed over a region of interest, the *sensor field*. Sensors use battery power as the energy source and they are capable of observing the area within their sensing range, process the data they collect and communicate with neighboring sensors in their communication range. On the other hand, central devices called sinks are the final destinations of the information transmitted by the sensors in a one-hop or multi-hop fashion.

The collaborative effort of the sensors is the essence of WSNs which provides a distributed monitoring environment even for remote or inaccessible areas. As a consequence, WSNs have a broad applicability range in areas such as military (Winkler *et al.*, 2008; Lee *et al.*, 2009), homeland security (Merrill *et al.*, 2003), health care (Dishongh and McGrath, 2009), environment (Marshall *et al.*, 2003; Martinez *et al.*, 2004), agriculture (Burrell *et al.*, 2004; Butler *et al.*, 2004), logistics (Riem-Vis, 2004), habitat monitoring (Juang *et al.*, 2002; Polastre *et al.*, 2004), smart home or office design (Kappler and Riegel, 2004), disaster management (Michahelles *et al.*, 2003; Werner-Allen *et al.*, 2006) and many others (Callaway, 2004; Dargie and Poellabauer, 2010). Interested readers may refer to the state-of-the-art WSN survey papers such as the ones by Akyıldız *et al.* (2002a,b, 2007); Culler *et al.* (2004); Yick *et al.* (2008) which provide a general perspective. Besides, some surveys provide more specialized settings of WSNs such as node placement strategies (Younis and Akkaya, 2008), efficient routing methods (Al-Karaki and Kamal, 2004), coverage and connectivity issues (Ghosh and Das, 2008; Fan and Jin, 2010).

Sensors have very limited energy resources and deplete them very fast. They become inactive when they run out of energy and the WSN can fail monitoring local areas in the sensor field, which means that the network is not fully operational anymore. Moreover, sensors can vary in terms of their technical characteristics such as sensing range and transmission range. WSNs with only one type of sensor are called

homogeneous whereas WSNs consisting of more than one type of sensor are said to be heterogeneous. Sensors can be categorized according to their working status such as active and standby. An active sensor senses, transmits and receives data and consumes energy for these operations whereas a standby sensor spends negligible energy while not providing any service at all. It should be noted that the standby mode is different than the sleep mode realized in regular duty schedules since a sleeping sensor continues to sense and uses energy while a standby sensor does not.

Almost all of the mathematical optimization models dealing with WSN design have four major concerns: lifetime, cost, coverage and connectivity. They can be addressed either in their objective functions or constraints. This thesis concentrates mainly on the maximization of the network lifetime and defines it as the time elapses until the sensors are unable to meet the coverage requirements of the sensor field. Cost, coverage and connectivity issues are handled in the constraints.

The design of a WSN involves in the optimal deployment and activity scheduling of the sensors as well as optimal deployment or routing of sinks and optimal routing of data flows. From this perspective it is possible to say that it consists of the integration of four fundamental optimization problems. The first one is known as the Coverage Problem (CP) and deals with the determination of optimal sensor locations so as to satisfy sensing quality requirements everywhere in the sensor field with a certain flexibility of handling individual sensor failures. Sensing quality requirements, or simply coverage requirements, can be uniform or non-uniform over the sensor field. In the former, the CP is uniform, which becomes differentiated in the latter. Data transmission is another major sensor activity and the determination of the most efficient sensor-to-sink message flows is the second problem. This is called as the Data Routing Problem (DRP) where it is assumed that both sensor and sink locations are given. Sensor-to-sink data routes have a prominent effect on the network lifetime. One can count on some fixed route data propagation protocols such as the shortest path data routing protocol for the determination of data routes, or amount of data flows on each arc of the network can be optimally determined. The latter approach is superior as it may lead to a network lifetime at least as long as the one obtained by any given

data routing protocol. Since modern sensors can adjust their power according to the desired communication range, the transmitter sensor consumes energy depending on the distance between the receiving sensor and itself. Therefore, sensor locations and data routes should be determined so that the energy load is evenly distributed among the sensors while the coverage requirements are satisfied at the same time.

Third fundamental WSN design problem is the Activity Scheduling Problem (ASP) which is basically an energy concerned operational issue, since it makes it possible to recreate a subset of the sensors for some time periods by entering them into standby mode. A good schedule of the sensor activities can therefore considerably increase the lifetime of the WSN and hence provides a tool for balancing the load of the sensors. It should be noted that an active sensor subset should be able to provide a minimal fully operational WSN.

Sink locations considerably affect the optimality of the routes as well. Hence, finding them optimally, namely solving the Sink Location Problem (SLP), is the fourth problem. This problem can be in the form of Sink Routing Problem (SRP) if the sinks are mobile. Mobility of the sinks are offered as a solution strategy for the phenomenon which is called differently in several sources such as "the crowded center effect" (Popa *et al.*, 2007), "energy hole problem" (Li and Mohapatra, 2007; Wu *et al.*, 2008), and "sink neighborhood problem" (Basagni *et al.*, 2008). It is a well known fact that the sensors connected directly to a sink deplete their energy much faster than the rest of the network due to the fact that these sensors carry all the data produced by the network. Controlled mobility of the sinks has been an effective strategy since neighboring sensor subsets change with the relocation of the sinks. Sink travel times are usually considered as zero in the literature and hence the data collected during the sink travels are neglected. This thesis deviates from the conventional approach and puts an effort to reflect the effect of nonzero sink travel times on the network lifetime and takes the data during the sink travel times into consideration.

There is a rich literature on the optimal WSN design by maximizing the lifetime. Most of the existing mathematical optimization models consider only a subset of the

above mentioned design issues by focusing on one of the four fundamental optimization problems or the integration of SLP or SRP with DRP (SLDRP or SDRP), which assumes that sensor locations are given. There are very few studies which provide mixed integer linear programming (MILP) models integrating CP, DRP and SLP with activity scheduling (Türkoğulları *et al.*, 2009, 2010a,b). These types of integrations are deep integrations as defined by Geoffrion (1994): the subproblems are merged in the form of a monolithic formulation. However, when the integration is functional, we cannot talk about a monolithic formulation but interacting formulations. There is a clear sequential relation between the subproblems, namely an explicit protocol for their solutions and relations is defined (Geoffrion, 1994). For example, the formulation given in the beginning of Türkoğulları et al.'s earlier work (Türkoğulları *et al.*, 2009) presents the same deep integration as in Türkoğulları *et al.* (2010a) and Türkoğulları *et al.* (2010b), but the approach developed in the rest of that work functionally integrates the SLP and deeply integrates CP and DRP with activity scheduling. The authors first locate the sinks in the sensor field; and then, given these sink locations, determine the optimal sensor places and their activity schedules, and optimal information flows. This thesis extends the relevant literature by introducing new mathematical optimization formulations, which deeply integrates CP, DRP, ASP with SLP or SRP in order to design a heterogeneous WSN having multiple mobile or stationary sinks for the situation where the coverage is differentiated.

The rest of the thesis is organized as follows. The next chapter reviews the literature on the development of the mathematical optimization models proposed for solving the design issues of WSNs with mobile sinks. Chapter 3 includes the mathematical programming formulations which assume nonzero sink travel times. The integration of major WSN design issues with multiple mobile sinks are elaborated in Chapter 4 by three new mathematical models and the strategies developed for their solutions are given. Chapter 5 concentrates on the same integration issue but come up with a mathematical program with multiple static sinks and proposes a heuristic as a solution strategy. The positive effect of integration of major design issues and the success of the proposed heuristic solution strategies for the models are demonstrated by extensive numerical experiments which are given in Chapter 6. Finally, Chapter 7 concludes the

thesis and points out possible future research directions.

2. LITERATURE SURVEY

Roughly speaking, it is possible to group the studies on WSNs with mobile sinks in two main classes: the ones that try to coordinate the network in order to compensate the extra overhead due to the mobility of the sink(s) and the ones that concentrate on the determination of efficient route(s) for mobile sink(s).

The members of the first class propose new data communication and propagation protocols to improve some performance criteria such as the energy, throughput, accuracy, message latency and message loss rate under a given (constant or random) trajectory of the sink and do not pay attention to the deliberate decisions on sink(s) movements. Any interested reader is directed to Hamida and Chelius (2008), which reviews the existing state-of-the-art data dissemination protocols with mobile sinks. On the other hand, studies belonging to the second class provide mathematical programming models that optimize some WSN performance criteria such as the network lifetime, the total energy spent and the total cost for given data propagation protocols. They determine the optimal route(s) of the sink(s) with respect to the objective of the network designer. Although this is a rough classification, it helps to position the contribution of this thesis better. It belongs to the second class, therefore a review of all important works that use mathematical optimization to address the design issues of WSNs with mobile sinks is included. For this purpose, a further division of the mathematical optimization studies with mobile sinks in two categories is provided: WSNs with the shortest path data flow protocol and WSNs with optimal route data flow. In the first group, data is routed from sensor to one of the sinks through the shortest sensor-to-sink path while data flow routes and amounts are optimally determined with respect to the employed performance metrics in the second one.

2.1. Shortest Path Data Flow

In the majority of the mobile sink related research papers, it is assumed that the messages are sent from sensors to the sinks through shortest paths. We present some

of the typical examples in this section.

Gu *et al.* (2005) consider a single mobile sink routed through a periodic path, and aim to find one ensuring no data loss. Sensors have buffers and the sink route should be short enough to ensure that no sensor's buffer is filled up between the sink visits. Sensors are first grouped with respect to their data collection rates and geographical locations, and the portion of the sink's route through each sensor group is then independently determined and concatenated to form the final route. Somasundara *et al.* (2007) also analyze the same problem; but their approach has a nonstandard perspective: they treat it as a variant of the single machine flow shop scheduling problem since it is possible to set the analogy between the sink and a processor, and between the sensors and jobs that are to be processed. Vincze *et al.* (2006) try to find a good path for the single mobile sink as well. However, they concentrate mainly on the energy usage of the sensors and assign negative or positive charges to sensors and sinks depending on their residual energy levels. Then, they apply the Coulomb rule in a repetitive manner until the sink locations converge. The paper by Soytürk and Altılar (2007) is also similar. Each sensor is assigned a weight indicating geographical proximity of the sensor to the sink and sensor's residual energy. A sensor that is farthest away from the sinks gets a high weight and sinks have zero weights. Weights of the sensors are also negatively related with the amount of residual energies they have. Data generated by the sensors flow from the high weighted sensors through the low weighted sensors until reaching one of the sinks. In other words, data flow through the shortest path from sensors to the sinks on the induced graph of the original network where arc length between two sensors is equal to the difference of their weights. Although these papers assume a controlled mobility for the sink, they rather depend on the intuitional rule-of-thumb methods instead of representing the issue by the help of some rigorous mathematical optimization models.

On the other hand, the work by Nesamony *et al.* (2006) assumes a mobile sink which travels around network area to gather the data from the sensors with a one-to-one connection. The aim of the paper is to find the shortest sink route that passes through sensor's transmission ranges. The problem is converted into a variant of the

Traveling Salesman Problem (TSP), TSP with neighborhoods, by representing each of the transmission ranges as a set of discrete points. This is a difficult problem and a heuristic method is offered for its solution. In their succeeding work, Valle *et al.* (2009) extend the setting given in Nesamony *et al.* (2006) to WSNs with multiple sinks. They try to minimize the longest sink tour and each sensor is required to be at least a given distance away from one of the sink tours, so that each sensor is able to send its data directly to one of the sinks. They develop an MILP formulation which turns out to be a variant of the uncapacitated vehicle routing problem, so they also propose a heuristic method for its solution. After determining the sink routes, they perform activity scheduling so that any point in the sensor field, represented as a grid, is covered by at least one sensor. These papers propose mathematical optimization models; but they do not concentrate on any WSN performance criterion such as the network lifetime, load balancing, data latency and delay.

In the paper by Wang *et al.* (2005), an optimal sink path maximizing the lifetime is found by the help of a linear program (LP) with energy constraints on the sensor batteries. The sink stops at some predefined visit points and the sum of the sojourn times on these points is defined to be the network lifetime. An optimal solution of the LP consists of the sink visit points with positive durations. However, a sequence of these points, namely a sink route, is not provided. An extension to the models of Wang *et al.* (2005) is due to Basagni *et al.* (2008). The authors aim to come up with not only the visit points and sojourn times but also a sequence of these points as an optimal solution. In order to represent the routing decisions related to the sink they extend the LP model of Wang *et al.* (2005) to a mixed-integer linear program (MILP). Another progress in that line of research is given in a more recent work by Basagni *et al.* (2011). They still concentrate on the maximization of the network lifetime by providing an LP but extend the setting to the one with multiple mobile sinks. They define a period as the duration of the sojourn time on a sink configuration, which is a permutation of the sinks among some predefined set of sink visit points. The lifetime, which is tried to be maximized in the objective function of the LP, is defined to be the sum of the sojourn times on the configurations. Alternatively, Liang *et al.* (2010) is one of the rare papers that considers the sink as an energy limited device that is

mechanically driven by either petrol or electricity. They propose an MILP formulation that is very similar to the one given in Basagni *et al.* (2008). However, in order to reflect the limited nature of the real sink devices, they use a constraint to put an upper bound on the total travel length of the sink. A similar constraint limits the distance between any two consecutive visit points of the sink, while another constraint bounds the sojourn times from below.

2.2. Optimal Data Flow

Instead of assuming a protocol for data flows a priori, the amount of data on each link of the WSN can be optimally found. Many works employ optimal route data flow protocols by focusing on the solutions of DRP and SLDRP or SDRP. Here, we only review the ones with single (multiple) mobile sink(s).

In their early work on SLDRP, Gandham *et al.* (2003) discuss the benefits of multiple stationary sinks for energy efficiency. They also suggest sink mobility as another strategy. This is also one of the earliest studies about the effect of sink mobility on the network lifetime. According to them using multiple sinks and periodically changing their locations increases the network lifetime. They split the lifetime of the WSN into equal periods called rounds, each of which consists of a given number of time frames, and assume that every sensor transmits one packet of data at the beginning of every time frame. They formulate two MILPs to determine sink locations at the beginning of each round and data flows over the time frames so that the maximum of sensor energy consumptions and total sensors' individual energy consumption are minimized. Azad and Chockalingam (2006) borrow their framework; but they offer two heuristics in addition to the exact minimization of the maximum energy consumption. In the first one, the sinks are located at points whose nearest neighbor nodes have the highest residual energies. In the second one, they try to locate the sinks to minimize the difference between the maximum and the minimum residual energies of the sensors. The work by Alsalih *et al.* (2007) is also based on a similar setting; but they try to maximize the minimum residual sensor energy. All these studies consider optimal data flows; but they focus more on the energy issues rather than maximizing the network

lifetime directly.

Luo and Hubaux (2005) address this issue and formulate the problem of lifetime maximization as a min-max problem by combining sink mobility together with data routing. An optimal solution of the corresponding formulation is load balancing. They perform their analysis assuming a circular sensor field and try to minimize the maximum load among the sensors in order to maximize the network lifetime. On the other hand, Papadimitriou and Georgiadis (2005) formulate a nonlinear program (NLP) to show that the maximum network lifetime, which is defined as the sum of sojourn times, can be achieved by solving the scheduling problem that determines the sojourn times of the sink at different locations, and routing problem in order to send the sensed data to the sink in an energy efficient way. A simple variable transformation applied on this NLP turns it into an easy-to-solve LP. This is a variant of the model by Wang *et al.* (2005). Their model jointly considers the problem of determining the optimal sink sojourn times at the given sink sojourn sites, and the routing of data packets to the current position of the sink by redefining the power consumptions at sensor nodes as variables, which are originally assumed constants in Wang *et al.* (2005). In a more recent work, Gatzianas and Georgiadis (2008) revisit the model given in Papadimitriou and Georgiadis (2005) and develop an efficient distributed algorithm. The distributed algorithm exploits the fact that the new formulation reduces to the one proposed by Madan and Lall (2006) where the sink is stationary and benefits from their use of Lagrangean duality and subgradient algorithm; the Lagrangean dual obtained by relaxing the flow conservation constraints using subgradient optimization. Two extensions to the models given in Papadimitriou and Georgiadis (2005) and Gatzianas and Georgiadis (2008) are provided in Yun and Xia (2010) where the data can be buffered at the sensors until the most beneficial instant (in terms of the network lifetime) comes. Since all the data are not immediately sent to the sink, these models are appropriate for delay-tolerant applications. In both of the models, it is assumed that the sink repeats the same trajectory (i.e., tour) many times, and it requires a predefined amount of time for the sink to complete its single tour, which they call the cycle time. This paper is the first one which explicitly considers multiple sink tours during the network lifetime and tries to maximize their number. Moreover, the authors use the sojourn time of the sink

at a sink visit point to define the cycle time: it is the sum of the sojourn times that the sink spends at each visit point. Therefore, the models of Yun and Xia (2010) would be stronger with an additional constraint equating cycle time to the sum of the sojourn times, which will prevent inflated network lifetimes while the real network lifetimes are actually smaller. Yun *et al.* (2010) propose a decomposition algorithm for the queue-based delay tolerant model given in Yun and Xia (2010) in order to come up with a distributed setting. Behdani *et al.* (2012) also propose a decomposition algorithm for the same model, but their methodology is more efficient in terms of computational time it requires. Shi and Hou (2008) provide an approximation algorithm that produces an ϵ-optimal solution for the lifetime maximization problem wherein the single mobile sink is allowed to stop at any point within the two-dimensional (plane) sensor field. Hence, without starting with a set of predefined visit points for the sink, it is possible to let the sink visit any point within the smallest enclosing disk covering all sensors and obtain near optimal solutions by the provided approximation algorithm. Another paper with optimal data routing protocol is due to Luo and Hubaux (2010) in which the maximum network lifetime is sought under the presence of multiple mobile sinks. Similar to Basagni *et al.* (2011) and Papadimitriou and Georgiadis (2005), the network lifetime is defined as the sum of sojourn times on the sink configurations over a predefined set of sink visit points which coincide with sensors' candidate locations. Data flow balance and energy constraints are imposed on each sensor and a heuristic method is proposed for the solution of the MILP model. Finally, the framework provided by Behdani *et al.* (2013) is similar to the one of Liang *et al.* (2010) but they also consider optimal data routing. They consider several MILP models having a sink with constant velocity and come up with a solution strategy based on Bender's decomposition.

None of the works mentioned above deeply integrates CP, ASP, SLP or SRP and DRP in a monolithic formulation; they study individually in general. As we have mentioned earlier, the type of integration most frequently carried is obtained by combining DRP and SLP (i.e., SLDRP) or DRP and SRP (i.e., SDRP). There are also a few studies which try to provide a more unified framework in which the integration of more than one design issues of sensor placement, activity scheduling and sink placement or the determination of the mobile sink's trajectory is accomplished. For instance, Güney

et al. (2010) assume that the sensor locations are known and the authors try to find the optimal sink locations and data routes. Lagrangean relaxation based solution strategies are implemented for MILP models. This paper can be considered as complementary to the studies concentrating on sensor deployment to ensure the coverage as it describes the view from perspective of sinks only. Güney *et al.* (2012) deeply integrates CP, SLP and DRP using the results of their earlier work (Güney *et al.*, 2010). A tabu search metaheuristic solution strategy is employed for the solution of the MILP model. The setting of Türkoğulları *et al.* (2009) is also similar to the one of Güney *et al.* (2012) but they assume that sink locations are given and try to find activity schedules of the sensors. They provide efficient heuristic strategies for the solution of the offered MILP model. Therefore, this paper also presents a deep integration of SLP and DRP with activity scheduling. Moreover, Türkoğulları *et al.* (2010a) work on an extended setting in which sink placement is also a concern on top of sensor placement, activity schedules of the sensors and data routes. A simple heuristic is proposed as a solution method. A column generation based heuristic is derived for their MILP model in Türkoğulları *et al.* (2010b) for the integration of CP, ASP, DRP and SLP, where the first three form the building elements of a monolithic MILP and integrate with SLP functionally: CP, ASP and DRP are solved using the MILP given the sink locations, which are calculated a priori using a simple heuristic. These two papers achieve the highest level of deep integration in the optimal WSN design studies of the literature.

In Table 2.1, a summary of the reviewed studies with respect to the depth of integration they have is provided. First column of the table includes the authors of the papers and the following four columns stand for the coverage, activity scheduling, sink location and sink routing problems. The last column shows whether the paper considers Multiple Sinks (MS) or not. Note that DRP requires to find optimal amount of data on each connection of the network. Therefore, studies employing shortest path data routing protocol do not integrate DRP in their MILP models. The table can further be enlarged with respect to some other criteria such as heterogeneity of the network, whether or not nonzero sink travel times are considered, but we confine ourselves to include the mentioned six criteria.

Table 2.1. Summary of the depth of integration of the literature studies.

Papers	**CP**	**ASP**	**SLP**	**SRP**	**DRP**	**MS**
Gu *et al.* (2005)				X		
Somasundara *et al.* (2007)				X		X
Vincze *et al.* (2006)				X		X
Soytürk and Altılar (2007)				X		X
Nesamony *et al.* (2006)				X		
Valle *et al.* (2009)				X		X
Wang *et al.* (2005)				X		
Basagni *et al.* (2008)				X		
Basagni *et al.* (2011)				X		X
Liang *et al.* (2010)				X		
Gandham *et al.* (2003)				X	X	X
Azad and Chockalingam (2006)				X	X	X
Alsalih *et al.* (2007)				X	X	X
Luo and Hubaux (2005)				X	X	
Papadimitriou and Georgiadis (2005)				X	X	
Gatzianas and Georgiadis (2008)				X	X	
Madan and Lall (2006)			X		X	
Yun and Xia (2010)				X	X	
Yun *et al.* (2010)				X	X	
Behdani *et al.* (2012)				X	X	
Shi and Hou (2008)				X	X	
Luo and Hubaux (2010)				X	X	X
Behdani *et al.* (2013)				X	X	
Güney *et al.* (2010)			X		X	X
Güney *et al.* (2012)	X		X		X	X
Türkoğulları *et al.* (2009)	X	X			X	X
Türkoğulları *et al.* (2010a)	X	X	X		X	X
Türkoğulları *et al.* (2010b)	X	X	X		X	X

In this thesis, we first consider the situation where sink travel times are nonzero and the data accumulated during the sink travel time is taken into account. The total sink travel time is considered as a part of the network lifetime. We provide two MILP models that are flexible enough to handle multiple sink tours as well as a hop limited data routing protocol in which data is routed from sensors towards the sink through the paths including at most a predefined number of hops. We propose heuristic procedures for the solution of the MILP models and show the importance of considering nonzero sink travel times by numerical experiments. An extension to these MILP models that possesses a framework with multiple mobile sinks is also developed and it is shown that sink travel times can be neglected for multiple sinks. Later on, we develop several MILP models which deeply integrate sensor placement, activity scheduling and data routing issues with the static sink placement or mobile sink routing design arguments and then provide practical heuristics and a theoretical exact solution approach. The depth of the integration changes from the integration of the sink routing problem (SRP) with the data routing problem (DRP) (i.e., Sink and Data Routing Problem (SDRP)) to the integration of the coverage problem (CP), SRP and DRP with ASP for heterogeneous WSNs having multiple mobile sinks. As can be understood from Table 2.1, this is the first time in the literature that an integration with such depth is realized. We finally develop heuristic solution strategies and also demonstrate the positive effect of integration of WSN design issues.

3. MOBILE SINK WITH NONZERO TRAVELING TIME

In this chapter,[1] we extend the relevant literature by introducing three new mathematical programming models with four major contributions. First of all, the majority of the existing works assume instantaneous jumps of the sink between the visit points and do not take into account its traveling time. Although it is employed frequently, this is a rather strong assumption. In a military application, the mobility of the sink can be restricted because of security reasons. Another example is the nature surveillance application where too much movement of the sink means too much interference to the wild habitat. Hence, slow sink velocities can be obligatory resulting in long sink travel times, which makes them viable for consideration. Moreover, if the sink moves are assumed to be instantaneous, longer sink tours are obtained which leads to higher real sink travel times in the application, which is the case in the majority of the previous studies. We basically argue that the stationarity and full mobility of the sink are two extremes of its mobility. A limited mobility for the sink may be the most relevant strategy for real WSN applications. Moreover, the data accumulated during the sink travel time are ignored in all of these models, which would damage the reliability and integrity of the collected data. Therefore, as mentioned in the study by Yun and Xia (2010), nonzero sink travel times can lead to interesting conclusions. Hence, we develop three mathematical models that employ a constant sink velocity, which leads to positive sink travel times. In addition, the data accumulated during the sink travel time between two visit points is taken into account. The optimal solutions of the developed mathematical models do not only provide a set of sink visit points and the corresponding sojourn times but also an optimal sequence among the sink visit points. The network lifetime is defined as the sum of the sink travel time and total sojourn times on the sink visit points. As can be seen from the computational results given in Chapter 6, we show that especially for realistic sized networks with single mobile sinks the models that do not take the duration of the sink tour into account yield either relatively bad or infeasible solutions and sink travel times can be neglected for the case of multiple mobile sinks.

[1] The work by Keskin *et al.* (2011) is based on the content of this chapter.

The second contribution of this study is the development of a limited hop model with a single sink in which only the sensors that are close to the sink are allowed to send their data, i.e., sensor's data can reach the sink in no more than a predefined number of hops. Other sensors wait and buffer their data until the sink comes nearby. The model is appropriate for the delay-tolerant applications in which the data delivery postponement can be employed in order to improve the lifetime. Moreover, conventional optimization models cannot handle the networks with disconnected topologies where the data delivery postponement is obligatory. The mathematical model given in Yun and Xia (2010) is general and useful for delay-tolerant applications. The authors define the set of sensors that are able to send their data to the sink for every possible sink visit point. Although they determine the set of sensors for each sink visit point using the Euclidean distance between the sensors and the sink, they do not pay attention to the number of hops directly. This approach disregards the sensors having relatively small Euclidean proximity to the sink but require many hops to reach the sink either due to energy efficient routing protocols or due to the terrain properties. Hence, the new hop limited model can be considered as an improvement over the one given in Yun and Xia (2010).

As the third contribution, multiple sink tours on the optimal sink trajectory is allowed. To the best of our knowledge, all of the existing works with the exception of Yun and Xia (2010) implicitly assume that the lifetime of the network ends as soon as the sink finishes its tour. In other words, it is assumed that the sink completes one single tour during the lifetime of the WSN, which can cause unnecessarily long sojourn times for the sink and delayed transfer of the collected data to a backbone or a stationary main station. However, if the sink can make multiple tours, then it will pass by the sensors and the main station more often and collect and transfer information packets in a timely manner without any need for a long term storage. Also, it may be more realistic to let the sink repeat the same trajectory, at least several times during the entire network lifetime, for technical considerations such as the oil replacement and periodic maintenance. We employ the number of sink tours as a design parameter in the model and computationally analyze the sensitivity of the optimal solutions with respect to it. Notice that it is always possible to reduce our multiple-tour models into

a single-tour model by simply setting this parameter to one.

As the last contribution, we devise fast and accurate heuristics for the solution of the mathematical models with single sink. We show that the proposed heuristics are very efficient and give near-optimal solutions. The details about their performance are provided in the computational results section. We also empirically show that the sink travel times can be neglected for multiple sink case. Hence, we do not provide solution strategies for the model with multiple mobile sinks.

3.1. Mathematical Formulations

We first give two MILP models with single sink and provide an MILP model with multiple mobile sinks at the end.

3.1.1. Single Sink

Here, we propose two models in which the data is sent through the shortest paths from sensors to the sink. In the first one, the length of the data routes is not limited. In other words, gathered data in a sensor is sent to the sink via the shortest path independent of the sink location. On the other hand, the second model limits the number of hops. In that one, the data gathered in a sensor is again sent through the shortest path, but only if the sink is located close to the sensor such that the shortest path from the sensor to the sink obeys the hop limit condition. If the sink is away from the sensor, it has to wait and buffer its collected data until the sink comes nearby. It is worth mentioning that the second model is especially more realistic for large size networks.

In the majority of the proposed mathematical optimization formulations, the target application is assumed as periodic data collection (Basagni *et al.*, 2008, 2011; Gatzianas and Georgiadis, 2008; Papadimitriou and Georgiadis, 2005; Wang *et al.*, 2005; Yun and Xia, 2010). For these applications, sensors are awake only while transmitting/receiving data and remain in sleep mode in other times. For that reason, we

also assume that the energy consumption is proportional to the amount of data transmitted and received. However, we should point out that if the sensors are awake only when they transmit/receive data, then they need to be synchronized; and this comes at an additional cost due to the additional control traffic, which is not captured in the framework presented here.

3.1.1.1. Unlimited Hop Model. In the Unlimited Hop Model (UHM), data are allowed to flow from sensors to the sink following the shortest paths without any restriction on its length. We consider that there are multiple candidate sink locations which are indexed by $i \in \mathcal{N} = \{1, 2, \ldots, N\}$ where $\mathcal{N}$ and N are respectively the set and number of candidate sink locations. We use both i and j as indices for sink locations in single sink models given in this section. Similarly, sensors are indexed by $k \in \mathcal{S}$ where $\mathcal{S}$ is the set of sensors. $\mathcal{S}_{ki}$ is defined to be the set of sensors for which sensor k lies on the shortest path to point i. We let d_{ij} denote the distance between sink locations i and j and D to be an upper bound on the length of a sink tour while v is taken as the finite speed of the sink. R stands for the number of sink tours during the lifetime and h is the amount of data packets produced by a sensor per unit time. $tmin_i$ and $tmax_i$ put lower and upper bounds on the total sojourn time at $i \in \mathcal{N}$. Energy related parameters are represented by c^t_{ki}, c^r_k and E_k which are respectively, the unit data transmission cost of sensor k while the sink is at point i, the unit data reception cost of sensor k while the sink is at point i, and the initial energy of sensor k. On the other hand, decision variables are; x_{ij}, which denotes whether or not arc (i, j) is included in the sink tour or not, t_i, is responsible for total sojourn time at $i \in \mathcal{N}$, u_i represents the order of the visit point $i \in \mathcal{N}$ on the sink route and finally, F_{ki} stands for the amount of data that comes to sensor k while the sink is at point i. Now, the mathematical model is given as:

UHM:

$$\max \sum_{i \in \mathcal{N}} t_i + R \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \frac{d_{ij}}{v} x_{ij} \tag{3.1}$$

s.t.

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} d_{ij} x_{ij} \leq D \tag{3.2}$$

$$tmin_i \sum_{j \in \mathcal{N}} x_{ji} - t_i \leq 0 \qquad i \in \mathcal{N} \quad (3.3)$$

$$t_i - tmax_i \sum_{j \in \mathcal{N}} x_{ji} \leq 0 \qquad i \in \mathcal{N} \quad (3.4)$$

$$F_{ki} - h \sum_{m \in \mathcal{S}_{ki}} \left(t_i + R \sum_{j \in \mathcal{N}} \frac{d_{ji}}{v} x_{ji} \right) = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \quad (3.5)$$

$$\sum_{i \in \mathcal{N}} \left\{ c_{ki}^t \left[F_{ki} + h \left(t_i + R \sum_{j \in \mathcal{N}} \frac{d_{ji}}{v} x_{ji} \right) \right] + c_k^r F_{ki} \right\} \leq E_k \qquad k \in \mathcal{S} \quad (3.6)$$

$$\sum_{i \in \mathcal{N}} x_{1i} = 1 \quad (3.7)$$

$$\sum_{i \in \mathcal{N}} x_{i1} = 1 \quad (3.8)$$

$$\sum_{j \in \mathcal{N}} x_{ji} \leq 1 \qquad i \in \mathcal{N} \quad (3.9)$$

$$\sum_{j \in \mathcal{N}} (x_{ji} - x_{ij}) = 0 \qquad i \in \mathcal{N} \quad (3.10)$$

$$u_i - u_j + N x_{ij} \leq N - 1 \qquad i, j \in \mathcal{N} \backslash \{1\} \quad (3.11)$$

$$t_i \geq 0 \qquad i \in \mathcal{N} \quad (3.12)$$

$$F_{ki} \geq 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \quad (3.13)$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in \mathcal{N} \quad (3.14)$$

In this formulation, the objective function (3.1) represents the network lifetime which is defined as the sum of the total sojourn time and total travel time for R tours of the sink. The second term of the objective function is obtained by multiplying R with the duration of a tour. Notice that $\frac{d_{ij}}{v}$ is the time to go from point i to point j. Constraint (3.2) puts an upper bound on the total length of the sink route. Constraints (3.3) force the sink to stay at least $tmin_i$ amount of time if it visits point i. Constraints (3.4) ensure that the sojourn time at point $i \in \mathcal{N}$ is less than or equal to a specified upper limit $tmax_i$ and it is zero if it is not on the sink's route. Notice that for each sensor belonging to $\mathcal{S}_{ki}$, the shortest path from the sensor to the sink visit point i includes sensor k. Then, the data coming to sensor k while the sink is at visit point i should be equal to the sum of the produced data at the sensors belonging to $\mathcal{S}_{ki}$. This is ensured

by constraints (3.5). It should be noted here that the set $\mathcal{S}_{ki}$ excludes sensor k. Hence the data generated by sensor k is not considered in F_{ki} which contains the received data by sensor k while the sink is visiting point i. Constraints (3.6) force the energy spent by each sensor throughout the network lifetime to be less than the initial battery energy. Observe that the total energy spent by a sensor is the sum of the energies spent by the sensor during the sink's sojourn time at each sink visit point and the energy required for the transmission of the accumulated data during the total sink travel time to point i from the previous point j for R tours. Sink travel time between two points is found by dividing the distance between the points by the sink speed. The accumulated data is computed by multiplying the total sink travel time by the unit data production rate of the sensor. Constraints (3.7) and (3.8) guarantee that the sink starts and ends up at home station 1. Constraints (3.9) make each sink visit point to be traversed by the sink at most once in a single tour. Constraints (3.10) ensure that once the sink enters a visit point, it must leave it as well. Subtours are eliminated by constraints (3.11) which is originally due to Miller *et al.* (1960). It should be noted that it is possible to employ some other subtour elimination constraints which are known to produce tighter linear relaxation bounds such as the ones given by Desrochers and Laporte (1991) and Sherali and Driscoll (2002) as summarized by Öncan *et al.* (2009). However, we confine ourselves to continue with the current form of the subtour elimination constraints in the scope of this thesis. Finally, constraints (3.12) − (3.13) and constraints (3.14) are the nonnegativity and binary restrictions, respectively.

It is important to observe that any sensor can send its data to the sink as soon as it arrives to a visit point. Hence, data is accumulated only when the sink is on its way from one visit point to another; and the amount of data stored in this time interval is very limited, which makes the unlimited sensor storage capacity assumption reasonable. Also, it is possible to observe that packet delays cannot exceed the sink travel times and they are minimal.

3.1.1.2. Limited Hop Model. Data transmission over large distances through unlimited hops can be very inefficient in terms of the energy usage, and data congestion and

loss may increase with the increasing message route distances. Besides, the topology of the network may force the limitation of the data travel length. In other words, in a disconnected network, it would be irrational to assume that all the sensors send their data when the sink is at a visit point. In order to overcome these deficiencies, we limit the number of hops on all message routing paths. We remark that the unlimited hop model is a special case of the Limited Hop Model (LHM) where the number of hops is very large. We mostly use the notation introduced for UHM, and add a few new definitions. One of them is the set $\mathcal{N}_k$; it denotes the set of sink visit points that can serve sensor $k \in \mathcal{S}$. New parameter z_{ki} indicates whether or not sensor $k \in \mathcal{S}$ is served by the sink visit point $i \in \mathcal{N}$ while new variable n_{ki} represents the amount of passive time of sensor $k \in \mathcal{S}$ that passes until the sink is at visit point $i \in \mathcal{N}$. Mathematical model is given as follows:

LHM:

$$\max \sum_{i \in \mathcal{N}} t_i + R \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \frac{d_{ij}}{v} x_{ij} \tag{3.15}$$

s.t.

$$(3.2) - (3.4)$$

$$\sum_{i \in \mathcal{N}_k} \sum_{j \in \mathcal{N}} x_{ji} \geq 1 \qquad k \in \mathcal{S} \tag{3.16}$$

$$n_{ki} - \sum_{j \in \mathcal{N}} x_{ji} \left((t_j + n_{kj}) (1 - z_{kj}) + R \frac{d_{ji}}{v} \right) = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.17}$$

$$F_{ki} - h \sum_{m \in \mathcal{S}_{ki}} (t_i + n_{mi}) z_{mi} = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.18}$$

$$\sum_{i \in \mathcal{N}_k} \left\{ c^t_{ki} \left[F_{ki} + h (t_i + n_{ki}) \right] + c^r_k F_{ki} \right\} \leq E_k \qquad k \in \mathcal{S} \tag{3.19}$$

$$(3.7) - (3.14)$$

$$n_{ki} \geq 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.20}$$

This formulation has the same structure as its unlimited counterpart except constraints $(3.16) - (3.19)$. Constraints (3.16) are newly introduced and they force the sink to visit at least one point from the set $\mathcal{N}_k$ for each sensor $k \in \mathcal{S}$ so that every sensor is eventually able to send its data to the sink in each tour. Constraints (3.19) are the energy limitations and ensure that the total energy spent cannot exceed the initial

battery energy of each sensor. The total energy spent by a sensor is defined as the sum of the required energies of that sensor for each sink visit point. The energy required by sensor $k \in \mathcal{S}$ while the sink is at point $i \in \mathcal{N}$ is equal to the sum of the product of parameter c^t_{ki} with the total data to be sent from sensor k to point i, and the product of c^r_k with the total data received by sensor k while the sink is at point i, namely F_{ki}. On the other hand, the total amount of data to be sent from sensor k to the sink at visit point i is defined as the sum of the received data and the data produced by sensor k during the total of the sink sojourn time at i and the length of time that sensor k is unserved until the sink is at i, which is n_{ki}. They are defined by equalities (3.17), which provide a recursive expression which keeps track of the time that sensor k is not active in a reverse manner from the present until the time when the sink is at some $j \in \mathcal{N}_k$. Notice that each sensor accumulates the data it collects when the sink is not close enough. Constraints (3.18) are the same as constraints (3.5) of UHM, but use the new variables n_{ki}.

LHM is in fact nonlinear because binary variables x_{ji} are multiplied with continuous variables t_j and n_{kj} in constraints (3.17). In order to get rid of the resulting nonlinearities, we define variables p_{ji} and f_{jik} where $p_{ji} = x_{ji}t_j$ and $f_{jik} = x_{ji}n_{kj}$. Moreover, we replace (3.17) with the following constraints.

$$n_{ki} - \sum_{j \in \mathcal{N}} \left((p_{ji} + f_{jik})(1 - z_{kj}) + R\frac{d_{ji}}{v}x_{ji} \right) = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \qquad (3.21)$$

$$p_{ji} - x_{ji}tmax_j \leq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N} \qquad (3.22)$$

$$p_{ji} - t_j \leq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N} \qquad (3.23)$$

$$p_{ji} - t_j + tmax_j(1 - x_{ji}) \geq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N} \qquad (3.24)$$

$$f_{jik} - nmax_{kj}x_{ji} \leq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N}, k \in \mathcal{S} \qquad (3.25)$$

$$f_{jik} - n_{kj} \leq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N}, k \in \mathcal{S} \qquad (3.26)$$

$$f_{jik} - n_{kj} + nmax_{kj}(1 - x_{ji}) \geq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N}, k \in \mathcal{S} \qquad (3.27)$$

$$p_{ji} \geq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N} \qquad (3.28)$$

$$f_{jik} \geq 0 \qquad j \in \mathcal{N}, i \in \mathcal{N}, k \in \mathcal{S} \qquad (3.29)$$

After replacing constraints (3.17) with constraints $(3.21) - (3.27)$ and adding the new nonnegativity restrictions (3.28) and (3.29) to the model, the formulation of LHM becomes an MILP which can be solved using a commercial solver.

In contrast to UHM, a sensor cannot send its data to the sink until it visits a point within the hop count range of the sensor in LHM. As a result, considerable packet delays can occur for the more realistic LHM. This makes the analysis of the effect of major design parameters such as the number of tours the sink repeats during the lifetime of the WSN and the number of deployed sensors, or the average packet delay an important issue. Additional computational results on this performance criterion are provided in Chapter 6.

The existence of a limit on the number of hops can increase the need for data storage in case of LHM. However, since the collected data is scalar and the storage requirement is very small, the storage capacity of the sensors will never be exceeded until the sink returns to a point within the hop count range. As a consequence, we assume that sensors have infinite storage capacities.

Both UHM and LHM are MILP models and their exact solutions can be time-consuming for realistic WSNs. Therefore, it is necessary to employ heuristics to compute good but not necessarily optimal solutions. We propose a new heuristic in Section 3.3.

3.1.1.3. On the Energy and Propagation Models. First of all, we assume that the energy consumption is proportional to the amount of data transmitted and received, which is typical for the mathematical optimization models proposed for message routing in WSNs. This assumption provides reasonable approximations for the applications where the traffic is deterministic and data collected periodically during well-planned intervals. In such cases sensors are awake only while transmitting / receiving data and remain in stand-by mode, during the rest of the WSN's lifetime. However, if the WSN is planned for event driven operation in which the data traffic is random, other energy

models which incorporates more complex scheduling will be required.

Furthermore, we also assume that a sensor can communicate with another one if the distance between the sensors is less than or equal to their transmission ranges. Although we use the disc model to represent the communication between the sensors, we are still able to incorporate more realistic channel models in our problem formulations. The received signal power at a data receiving sensor can be estimated according to the assumed propagation model such as free space propagation model, two-ray ground reflection model (Rappaport, 1996) depending on the distance and the transmitted signal power of the data transmitting sensor. Moreover, one can find out the probability of correctly receiving a packet at a data receiving sensor by making use of the received signal power by employing the Signal-to-Noise Ratio (Zuniga and Krishnamachari, 2004). Therefore, we are able to assign probabilities for each data transmission among the sensors and more interestingly these probabilities can be used to obtain the amount of average data that is correctly transmitted through each arc of the network. For instance, suppose a data is sent from sensor m to sink by following the shortest path to the sink which includes sensor k. Moreover, suppose there are two intermediary sensors between sensors m and k, say sensors a and b. Then we can calculate the percentage of the data that is transmitted by sensor m and correctly received by sensor k as $\alpha_{mk} = 100 \times P_{ma}P_{ab}P_{bk}$ where P_{ij} represents the probability that sensor j correctly receives a packet sent by sensor i. Under this terminology, Constraints (3.5) of UHM and Constraints (3.18) of LHM can be rewritten as follows, respectively. Note that α_{mk} stands for the percentage of data that is correctly received by sensor k among the data sent by sensor m to the sink.

$$F_{ki} - h \sum_{m \in \mathcal{S}_{ki}} \alpha_{mk} \left(t_i + R \sum_{j \in \mathcal{N}} \frac{d_{ji}}{v} x_{ji} \right) = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.30}$$

$$F_{ki} - h \sum_{m \in \mathcal{S}_{ki}} \alpha_{mk} \left(t_i + n_{mi} \right) z_{mi} = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.31}$$

Therefore, we can easily extend our model to incorporate more realistic channel models for sensor communications. Furthermore, the solution procedures that is explained in Section 3.3 will not be affected by these changes at all. However, in this thesis, we

choose not to deviate from the disc model in order to compare our solutions with the competitors in the literature such as Basagni *et al.* (2008) for showing the value of considering nonzero travel time in the formulations.

3.1.2. Multiple Sink Model

Here, we propose an MILP model with multiple mobile sinks (MSM) and traveling time of the mobile sinks are taken into consideration. Data accumulated during the sink travel times are also taken into account both in flow balance and energy constraints. Most of the needed nomenclature for that model is the same with the single sink model. However, we use i and j for the indices of the sensors in that case. Moreover, ℓ and n are the indices of sink visit points while s represents sinks. Several additional definitions are also required. For instance, $\mathcal{S}_i$ is the set of sensors neighboring sensor i and similarly $\mathcal{S}_\ell$ is the set of sensors neighboring the sink location ℓ. $\mathcal{M}_s$ stands for the set of sinks and $\mathcal{N}_i$ is the set of sink locations which is directly reachable from sensor i. We define $\mathcal{T}$ as the set of time periods and let t be the index of them. Finally, Y is the upper bound on the sensor-to-sink arc flows and v_s is the nonzero finite speed of sink s. We also need several set of new variables. w_t represents the length of period t while a_t is the passage time to period t from period $t-1$. Note that by passage time we mean the maximum travel time of the sinks between two consecutive periods $t-1$ and t. We let $z_{\ell ts}$ indicate whether or not sink s is located at ℓ at period t and $q_{n\ell ts}$ shows whether or not sink s goes from n to ℓ between periods $t-1$ and t. Finally, we let x_{ijt} and $y_{i\ell t}$ amount of flow from sensor i, to sensor j at period t and to sink at ℓ at period t, respectively. Below, we give the mathematical model:

MSM:

$$\max \sum_{t\in\mathcal{T}} (w_t + a_t) \tag{3.32}$$

s.t.

$$\sum_{j\in\mathcal{S}_i} x_{jit} + h\,(w_t + a_t) = \sum_{\ell\in\mathcal{N}_i} y_{i\ell t} + \sum_{j\in\mathcal{S}_i} x_{ijt} \qquad i\in\mathcal{S}, t\in\mathcal{T} \tag{3.33}$$

$$\sum_{t\in\mathcal{T}} \left(c^r \sum_{j\in\mathcal{S}_i} x_{jit} + \sum_{j\in\mathcal{S}_i} c^t_{ij} x_{ijt} + \sum_{\ell\in\mathcal{N}_i} c^t_{i\ell} y_{i\ell t} \right) \le E_i \qquad i\in\mathcal{S} \tag{3.34}$$

$$\sum_{i\in\mathcal{S}_j} y_{i\ell t} \leq Y \sum_{s\in\mathcal{M}_s} z_{\ell ts} \qquad \ell \in \mathcal{N}, t \in \mathcal{T} \qquad (3.35)$$

$$\sum_{\ell\in\mathcal{N}} z_{\ell ts} = 1 \qquad s \in \mathcal{M}_s, t \in \mathcal{T} \qquad (3.36)$$

$$q_{n\ell ts} = z_{n(t-1)s} z_{\ell ts} \qquad n, \ell \in \mathcal{N}, t \in \mathcal{T}\backslash\{1\}, s \in \mathcal{M}_s \qquad (3.37)$$

$$a_t \geq \sum_{n\in\mathcal{N}} \sum_{\ell\in\mathcal{N}} q_{n\ell ts} \frac{d_{n\ell}}{v_s} \qquad s \in \mathcal{M}_s, t \in \mathcal{T} \qquad (3.38)$$

$$w_t, a_t \geq 0 \qquad t \in \mathcal{T} \qquad (3.39)$$

$$x_{ijt} \geq 0 \qquad i \in \mathcal{S}, j \in \mathcal{S}_i, t \in \mathcal{T} \qquad (3.40)$$

$$y_{i\ell t} \geq 0 \qquad i \in \mathcal{S}, \ell \in \mathcal{N}_i, t \in \mathcal{T} \qquad (3.41)$$

$$q_{n\ell ts} \geq 0 \qquad n, \ell \in \mathcal{N}, t \in \mathcal{T}, s \in \mathcal{M}_s \qquad (3.42)$$

$$z_{\ell ts} \in \{0, 1\} \qquad \ell \in \mathcal{N}, t \in \mathcal{T}, s \in \mathcal{M}_s \qquad (3.43)$$

We define the lifetime as the sum of the period lengths and the travel time between the periods and try to maximize it in the objective function (3.32). Constraints (3.33) and (3.34) are usual flow balance and energy constraints. We assume that the data accumulated during the travel time between sink permutations are sent during the period corresponding to latter coming sink permutation. Constraints (3.35) ensure that no flow is transferred to a point if there is no sink visiting that point. Constraints (3.36) force each sink to be located somewhere among potential sink visit points. On the other hand, constraints (3.37) define the variable q which is an indicator of a travel between visit points. Namely, if a sink is located at n in period $t-1$ and located at ℓ in period t, then it must have moved from point n to point ℓ between periods $t-1$ and t. Therefore, the corresponding q variable takes value 1. Last constraints (3.38) put lower bound for the travel time between sink permutations. It simply says that the travel time between the sink permutations is at least as big as the longest sink travel time between these permutations. Constraints (3.40) $-$ (3.41) and (3.43) $-$ (3.42) are respectively the nonnegativity and binary restrictions.

Model is nonlinear due to the binary variable multiplication in constraint (3.37), but can be linearized easily. On the other hand, instead of defining separate w_t and a_t variables one can define an aggregate variable w_t. However, the new variable should be

interpreted as the sum of the sojourn time and the travel time. The linearized version of the model with this new definition can be given as:

$$\max \sum_{t\in\mathcal{T}} w_t \tag{3.44}$$

s.t.

$$\sum_{j\in\mathcal{S}_i} x_{jit} + h_i w_t = \sum_{j\in\mathcal{N}_i} y_{ijt} + \sum_{j\in\mathcal{S}_i} x_{ijt} \qquad i\in\mathcal{S}, t\in\mathcal{T} \tag{3.45}$$

$(3.34) - (3.36)$

$$q_{n\ell ts} \le z_{n(t-1)s} \qquad n,\ell\in\mathcal{N}, t\in\mathcal{T}\backslash\{1\}, s\in\mathcal{M}_s \tag{3.46}$$

$$q_{n\ell ts} \le z_{\ell ts} \qquad n,\ell\in\mathcal{N}, t\in\mathcal{T}\backslash\{1\}, s\in\mathcal{M}_s \tag{3.47}$$

$$q_{n\ell ts} \ge z_{n(t-1)s} + z_{\ell ts} - 1 \qquad n,\ell\in\mathcal{N}, t\in\mathcal{T}\backslash\{1\}, s\in\mathcal{M}_s \tag{3.48}$$

$$w_t \ge \sum_{n\in\mathcal{N}}\sum_{\ell\in\mathcal{N}} q_{n\ell ts}\frac{d_{n\ell}}{v_s} \qquad s\in\mathcal{M}_s, t\in\mathcal{T} \tag{3.49}$$

$(3.39) - (3.42)$

MSM is an extension to the models of Luo and Hubaux (2010). We simply incorporate the sink travel times to their models to create MSM. Here, one needs to check whether the consideration of nonzero travel times for the sinks improves the network lifetime or not. For that reason, optimal lifetimes of MSM and the models of Luo and Hubaux (2010) should be compared for some problem instances which we achieve and report in Chapter 6. As a result of the numerical experiments, we conclude that the sink travel times can be neglected if there are multiple mobile sinks even for large networks.

3.2. Solution Strategies for Single Sink Models

In this section, we describe the details of a heuristic approach for the solution of the UHM and LHM MILP models presented in the previous section. There are two phases of this approach. First, an initial feasible tour is constructed by a heuristic called TOUR-CONSTRUCTION (TCH). Then, it is improved by the TOUR-IMPROVEMENT heuristic (TIH). The quality of a tour is evaluated by the optimal value of the LP obtained after setting the values of the binary x_{ij} variables to the ones of the feasible tour at hand in the UHM and LHM formulations. As a matter of

convenience, the LP obtained by fixing the binary variables is called the Restricted LP (RLP) in the sequel. For instance, RLP for the LHM is given below.

RLP-LHM:

$$\max \; Z = \sum_{i \in \mathcal{N}} t_i + R \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \frac{d_{ij}}{v} x_{ij} \tag{3.50}$$

s.t.

$$tmin_i \sum_{j \in \mathcal{N}} x_{ji} - t_i \leq 0 \qquad i \in \mathcal{N} \tag{3.51}$$

$$t_i - tmax_i \sum_{j \in \mathcal{N}} x_{ji} \leq 0 \qquad i \in \mathcal{N} \tag{3.52}$$

$$n_{ki} - \sum_{j \in \mathcal{N}} x_{ji} \left((t_j + n_{kj})(1 - z_{kj}) + R \frac{d_{ji}}{v} \right) = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.53}$$

$$F_{ki} - h \sum_{m \in \mathcal{S}_{ki}} (t_i + n_{mi}) z_{mi} = 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.54}$$

$$\sum_{i \in \mathcal{N}_k} \left\{ c^t_{ki} \left[F_{ki} + h (t_i + n_{ki}) \right] + c^r_k F_{ki} \right\} \leq E_k \qquad k \in \mathcal{S} \tag{3.55}$$

$$t_i \geq 0 \qquad i \in \mathcal{N} \tag{3.56}$$

$$n_{ki}, F_{ki} \geq 0 \qquad k \in \mathcal{S}, i \in \mathcal{N} \tag{3.57}$$

It should be observed that x_{ij}'s are not variables anymore since their values are already known from the feasible tour at hand. Hence, there are only continuous variable sets t_i, n_{ki} and F_{ki}, which makes the above model (RLP-LHM) a linear programming problem. Moreover, the term $R \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \frac{d_{ij}}{v} x_{ij}$ in the objective function becomes a constant and can be added to the optimal value of the RLP.

On the other hand, both TCH and TIH repeat some primitive operations as sub-procedures. Hence, before proceeding to the explanation of the algorithmic structures, it would be helpful to give their detailed descriptions. Let $\mathcal{N}^{(t)}$ represent the set of sink visit points that are currently (i.e., at iteration t) on the tour and let l_p be the position of point $p \in \mathcal{N}^{(t)}$ on the tour. In other words, l_p is the order of the point p on the tour. Finally, a move is said to be feasible if its execution on a feasible tour does not result in an infeasible one. We call a tour infeasible if it violates the tour length restriction (3.2) or the sink visit points on it do not cover all sensors. We explain the

primitive operations below:

- $SWAP(p, q)$: Given $p, q \in \mathcal{N}^{(t)}$, this operation interchanges their locations in the current tour.
- *BEST_SWAP*: Among all feasible $SWAP(p, q)$ operations, the one which yields the maximum increase in the optimal value of the RLP.
- 2-$OPT(p, q, p', q')$: Given $p, q, p', q' \in \mathcal{N}^{(t)}$ such that $l_q = l_p + 1$ and $l_{q'} = l_{p'} + 1$ and $|l_p - l_{p'}| > 1$, this operation deletes the direct arcs $(p,\ q)$ and $(p',\ q')$ and constructs new direct arcs $(p,\ p')$ and $(q,\ q')$. In other words, x_{pq} and $x_{p'q'}$ are set to zero while $x_{pp'}$ and $x_{qq'}$ being set to 1. Moreover, in order to have a legitimate directed sink tour, for all the points $p'', q'' \in \mathcal{N}^{(t)}$ so that $l_q \leq l_{p''} < l_{p'}$ and $l_{q''} = l_{p''} + 1$, $x_{p''q''}$ is set to zero while $x_{q''p''}$ being set to 1.
- *BEST_2-OPT*: Among all feasible 2-OPT moves, the one which gives the maximum increase in the optimal value of the RLP.
- $INSERT(p, l)$: Given $p \in \mathcal{N} \backslash \mathcal{N}^{(t)}$ and l a positive integer less than or equal to the length of the current tour, this operation inserts point p into the tour so that its location in the tour becomes l, namely $l_p = l$.
- *BEST_INSERT* : Among all feasible $INSERT(p, l)$ moves, the one which results in the maximum increase in the optimal value of the RLP.
- $DROP(p)$: Given $p \in \mathcal{N}^{(t)}$, this operation drops the point p from the tour.
- *BEST_DROP* : Among all feasible $DROP(p)$ moves, the one which gives the maximum increase in the optimal value of the RLP.
- $REPLACE(p, q)$: Given $p \in \mathcal{N}^{(t)}$ and $q \in \mathcal{N} \backslash \mathcal{N}^{(t)}$, this operation replaces point p with point q.
- *BEST_REPLACE* : Among all feasible $REPLACE(p, q)$ moves, the one which yields the maximum increase in the optimal value of the RLP.

3.2.1. Tour Construction Heuristic

A tour that is capable of serving all sensors and satisfying the tour length restriction is constructed by TCH. Initially, the first sink visit point is added to the tour since all the tours are required to begin and end with the initial visit point; this is forced

by constraint sets (3.7) and (3.8). Therefore, initially $\mathcal{N}^{(t)} = \{1\}$. Later on, if there exists at least one unserved sensor, among all the sink visit points outside the tour, the one that can serve the largest number of unserved sensors, say $p \in \mathcal{N} \backslash \mathcal{N}^{(t)}$, is selected and operation $INSERT(p, l)$ is executed, which sets $\mathcal{N}^{(t)} = \mathcal{N}^{(t)} \cup \{p\}$. The insertion position l of the selected sink visit point p is determined so that the increase in the length of the tour is kept at minimum level. The insertion of the sink visit points to the tour is continued in this manner until no unserved sensor is left. If the obtained final tour violates the tour length constraint (3.2), its length is reduced by employing 2-$OPT(p, q, p', q')$ which leads to the highest reduction in the tour length among all possible 2-OPT operations. Execution of the 2-OPT moves are kept until the tour length satisfies the tour length constraint. The pseudo-code of TCH is provided in Figure 3.1.

Insert the sink visit point 1 to the tour, $\mathcal{N}^{(t)} = \{1\}$, $t \leftarrow 0$;
while (there is an unserved sensor) **do**
  Determine a point, say p, among sink visit points that are outside of the tour which serve the largest number of the unserved sensors
  Determine location l so that $INSERT(p, l)$ causes the minimum increase in the tour length
  Execute $INSERT(p, l)$ and set $\mathcal{N}^{(t)} = \mathcal{N}^{(t)} \cup \{p\}$, $t \leftarrow t + 1$
end while
while (tour length exceeds the limit) **do**
  Among all possible $p, q, p', q' \in \mathcal{N}^{(t)}$ determine p^*, q^*, p'^*, q'^* so that 2-$OPT(p^*, q^*, p'^*, q'^*)$ leads to the maximum reduction in the tour length
  Employ the 2-$OPT(p^*, q^*, p'^*, q'^*)$
end while
Report the final tour and its length

Figure 3.1. Tour construction heuristic.

For UHM, all of the sensors can be served from all sink visit points. Therefore, the construction heuristic stops after initializing the tour with the single initial sink visit point. In order to avoid initiating TIH with such a trivial tour, we use the initial tour obtained by TCH for LHM.

It should be noted that a feasible tour always exists since during the data generation, locations of the sink visit points are chosen in such a way that they can serve all sensors. In other words, for each sensor there is at least one sink visit point that the sensor can be served from. Moreover, the distance limitation for the tour length is assigned as a large enough number for all the problem instances which is larger than the shortest tour length that passes through all the sink visit points. The shortest tour length that passes through every sink visit point for all problem instances is calculated by using the CONCORDE software (Applegate *et al.*, 2010). The computation time for the solution of the traveling salesman problem (TSP) over the sink visit points is negligible because the number of the maximum sink visits point is less than 60 for all problem instances and a TSP with 60 cities is a relatively small problem for CONCORDE. In short, TCH is able to find a feasible tour for all problem instances which is going to be used as an initial tour for TIH, as explained in the next section.

while (There is room for improvement) **do**

Employ the one that gives the maximum increase in the objective value of the RLP among the *BEST_SWAP*, *BEST_2-OPT*, *BEST_INSERT*, *BEST_DROP* and *BEST_REPLACE* moves

end while

Report the final tour and its length

Figure 3.2. Tour improvement heuristic.

3.2.2. Tour Improvement Heuristic

Here we present TIH which improves the quality of a feasible tour by using the primitive operations explained above until no improvement in the quality of the tour is possible. At each iteration, among the *BEST_SWAP*, *BEST_2-OPT*, *BEST_INSERT*, *BEST_DROP*, and *BEST_REPLACE* moves, the one that yields the maximum increase in the objective function value of the RLP is executed and iterations are continued until no increase in the objective value of the RLP can be achieved by any possible *SWAP*, *2-OPT*, *INSERT*, *DROP*, and *REPLACE* moves. A pseudo-code for the heuristic is provided in Figure 3.2.

4. MOBILE SINK WITH ZERO TRAVELING TIME

After observing that sink travel times can be neglected under the presence of multiple mobile sinks, we concentrate on developing more extended MILP models with multiple mobile sinks. They assume zero sink travel times and integrates major WSN design problems such as CP, ASP, SLP, SRP and DRP. In this chapter, we present three models that integrate SRP and DRP, CP with SRP and DRP, and CP and ASP together with SRP and DRP. Therefore, the models of this chapter include multiple mobile sink routing decisions, and assume that sink travels happen instantaneously. After giving the mathematical formulations of the models we propose heuristics and an exact method for the solution of the third model, which is the one achieving the deepest level of integration. [2]

4.1. Mathematical Programming Formulations

The first MILP model we present in this section is an extension of the recent formulation by Luo and Hubaux (2010). The other two are further generalizations which improve the first model. The three subsections in this order provide a good presentation of the effect of increasing, the breadth of integration. All of the models share the same notation. Therefore, before proceeding with mathematical formulations of the models, we give their common notations. First of all, we assume that there are more than one type of sensors indexed by r while $\mathcal{R}$ is the set of sensor types. $\mathcal{S}$ denotes the set of candidate sensor locations and we let i and j be the indices of the candidate sensor locations. Moreover, $\mathcal{S}_{ir}$ is responsible from the set of sensor locations neighboring sensor (i, r) while sensor (i, r) denotes a type r sensor deployed at point i. $\mathcal{N}$ is the set of sink locations while $\mathcal{N}_{ir}$ is the set of sink locations neighboring sensor (i, r). Similarly, we represent the set of points to be covered by $\mathcal{K}$ and $\mathcal{N}_{ir}$ is the set of points covered by sensor (i, r). We assume that the coverage requirement of point k is represented by d_k. Besides, $\mathcal{T}$ indicates the set of time periods which are indexed by t. We assume that f_{ir} denotes the cost of placing sensor (i, r) while B stands for the total

[2] Works by Keskin *et al.* (2013a) and Keskin *et al.* (2013b) are based on the content of this chapter.

sensor deployment budget. The number of sinks are chosen as P while M is a very large number (i.e., big M) used as an upper bound where required. Data production rate of a type r sensor is given by h_r. Finally, energy related parameters are E_r, which is the battery energy of type r sensor, c^r which relates to the unit data reception cost, c^s which stands for the unit sensing and coordination cost, and lastly, c_{ij}^t is the unit data transmission cost between sensors deployed at candidate sensor locations i and j. On the other hand, models also require several variable sets. For instance, p_{ir} indicates whether or not sensor (i, r) is deployed, while q_{irt} shows whether or not sensor (i, r) is active in period t. w_t is the length of period t and $z_{\ell t}$ demonstrates whether or not a sink is located at point ℓ in period t. There are two auxiliary variable sets a_{irt} and b_{irt} which respectively replace product terms $w_t q_{irt}$ and $w_t p_{ir}$ in the formulations in order to linearize the formulations. Finally, x_{irjst} and $y_{ir\ell t}$ respectively reveals the amount of flow from sensor (i, r) to sensor (j, s) in period t and the amount of flow from sensor (i, r) to sink deployed at point ℓ in period t.

4.1.1. Moving Multiple Mobile Sinks and Routing Data Simultaneously

The first formulation is an enhancement of the MILP formulation given in Luo and Hubaux (2010), which models the SDRP for multiple mobile sinks and homogeneous WSNs. In its original form, the sensor locations, which are known, coincide with the sink visit points. Here, we present the sink visit points as a separate set, which may also include some of the sensor locations. This is an uncapacitated model and there is no capacity restrictions on each link. However, it is possible to transform it into a capacitated one by simply replacing constraint (4.6) by the upper bounding inequalities

$$y_{ir\ell t} \leq u_{ir\ell t} z_{\ell t} \qquad i \in \mathcal{S}, \ell \in \mathcal{N}, r \in \mathcal{R}, t \in \mathcal{T} \qquad (4.1)$$

$$x_{irjst} \leq u_{irjst} \qquad i, j \in \mathcal{S}, r, s \in \mathcal{R}, t \in \mathcal{T} \qquad (4.2)$$

with $u_{ir\ell t}$ and u_{irjst} denoting the given upper bound on the flow from a type r sensor located at point i to a sink located at point ℓ in period t and to a sensor of type s located at point j, respectively. We will continue with the uncapacitated flows, although it is

a special form of the capacitated model, since this does not affect the determination of the improvement introduced by the increase in the breadth of the integration. Finally, we consider heterogeneous WSNs by explicitly differentiating flows between different sensor types. As mentioned in some of the earlier works on sensor deployment, using different sensor types and allowing more than one sensor placement at a point can be more advantageous (Wang and Zhong, 2006; Altınel *et al.*, 2008). The new model can also benefit from such deployment results since it considers heterogeneous sensor networks. As described by Chang and Tassiulas (2004), different type of information, such as acoustic, magnetic, or seismic information, can be collected and routed to the sinks in a very likely scenario for a WSN. For all practical purposes, such WSNs can be realized by placing more than one sensor, each of a different type, at the same point, communicating their own messages. They formulate this scenario as an explicit multi-commodity message flow in the WSN, which can be seen as the major differentiating feature of their mathematical model. Recall that they assume that sink locations are given and thus their model solves only DRP. However, this is not true for the models of Luo and Hubaux (2010) and the one given below. Finally, it should be underlined that the sensor places are assumed to be known a priori for the model. Therefore, the set $\mathcal{S}$ includes the locations of the sensors. The mathematical formulation of the mentioned model for the SLRP with multiple mobile sinks (MSLRP) is as follows:

$$\max \sum_{t\in\mathcal{T}} w_t \tag{4.3}$$

s.t.

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + h_r w_t = \sum_{\ell\in\mathcal{N}_{ir}} y_{ir\ell t} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.4}$$

$$\sum_{t\in\mathcal{T}}\left(c^s w_t + c^r \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} c^t_{ij} x_{irjst} + \sum_{\ell\in\mathcal{N}_{ir}} c^t_{i\ell} y_{ir\ell t}\right) \le E_r \qquad i\in\mathcal{S}, r\in\mathcal{R} \tag{4.5}$$

$$\sum_{r\in\mathcal{R}}\sum_{i:\ell\in\mathcal{N}_{ir}} y_{ir\ell t} \le M z_{\ell t} \qquad \ell\in\mathcal{N}, t\in\mathcal{T} \tag{4.6}$$

$$\sum_{\ell\in\mathcal{N}} z_{\ell t} = P \qquad t\in\mathcal{T} \tag{4.7}$$

$$w_t, y_{ir\ell t}, x_{irjst} \geq 0 \tag{4.8}$$

$$z_{\ell t} \in \{0, 1\} \tag{4.9}$$

In this formulation, the objective function (4.3) represents the network lifetime which is defined as the sum of the period lengths. Constraint set (4.4) is the flow balance constraints. They are added for each sensor of each type for each period and they simply equate the amount of data that is either produced in the sensor or taken from the neighboring sensors to the total data sent out of sensor to the neighboring sinks and sensors. Constraints (4.5) set an upper bound on the energy spent by each sensor throughout the network lifetime. Observe that each sensor spends energy for their sensing and processing duties which is taken into account at the first term in (4.5). Moreover, energy used for receiving data from the neighboring sensors is handled in the second term while the transmission energies to neighboring sensors and sinks are considered in the third and forth terms. Each of these energy fractions are summed up over all periods to account for the energy spent by the sensor. Constraints (4.6) simply imply that there cannot be a sensor-to-sink flow from a sensor to a point if there is no sink placed at that point. Besides, constraints (4.7) ensure that all of the P sinks are located in each period. Finally, constraints (4.8) and (4.9) present the usual nonnegativity and binary restrictions.

4.1.2. Integrating the Coverage Problem with Data and Multiple Mobile Sink Routing Problems

As mentioned before, MSLRP assumes that sensor locations are given and determines optimal locations for each of the P sinks in each period so that the lifetime is maximized subject to the additional energy and data flow constraints. Here, we provide a model which inherits the setting of MSLRP but also includes the sensor placement decisions for lifetime maximization instead of taking them as predefined. Hence, we impose the sensor placement decisions upon the setting of MSLRP to obtain MCSLRP. Similar to the candidate sink visit locations, we define a set of candidate sensor locations in advance. In order to be coherent with real application environments, we assign

varying costs for deploying each type of sensor to each of the possible sink locations and envisage a budget limitation. We also take the sensing coverage requirements of the sensor field into consideration by representing the field as a set of finite points and by assigning coverage requirements to these points. In other words, we approximately represent the coverage requirements of a continuous surface by means of a finite discrete set of points (i.e., the sensor field model of the point coverage problem). A final note that should be mentioned here is that the set $\mathcal{S}$ used in the body of the following model includes all candidate sensor locations as no sensor location is known a priori. The mathematical formulation of the MCSLRP is given as follows:

$$\max \sum_{t\in\mathcal{T}} w_t \tag{4.10}$$

s.t.

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + h_r b_{irt} = \sum_{\ell\in\mathcal{N}_{ir}} y_{ir\ell t} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.11}$$

$$\sum_{t\in\mathcal{T}}\Bigg(c^s b_{irt} + c^r \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} c^t_{ij} x_{irjst} + \sum_{\ell\in\mathcal{N}_{ir}} c^t_{i\ell} y_{ir\ell t}\Bigg) \le E_r \qquad i\in\mathcal{S}, r\in\mathcal{R} \tag{4.12}$$

$(4.6) - (4.7)$

$$\sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} p_{ir} \ge d_k \qquad k\in\mathcal{K} \tag{4.13}$$

$$\sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{ir} p_{ir} \le B \tag{4.14}$$

$$\sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \le M p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.15}$$

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} \le M p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.16}$$

$$b_{irt} \le w_t \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.17}$$

$$b_{irt} \le M p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.18}$$

$$b_{irt} \ge w_t - M(1-p_{ir}) \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.19}$$

$$w_t, b_{irt}, y_{ir\ell t}, x_{irjst} \ge 0 \tag{4.20}$$

$$p_{ir}, z_{\ell t} \in \{0,1\} \tag{4.21}$$

Similar to MSLRP, we maximize the sum of the period lengths which is the network lifetime in the objective function. Constraints (4.11) and (4.12) stand for the data flow balance and energy constraints. Note that b_{irt} replaces the product term $w_t p_{ir}$, where w_t is the length of period t and p_{ir} indicates whether a type r sensor is located at point i or not. Therefore, if a type r sensor is located at i, it generates data that should be taken into consideration in constraint (4.11). It also spends energy for sensing and processing, which is restricted by constraint (4.12). The third and the fourth set of constraints of MCSLRP has the same formulation and interpretation with their corresponding constraints in the MSLRP. Constraints (4.13) guarantee that the coverage requirements are satisfied. Constraints (4.14) restrict the total amount of money that can be spent by the available budget B. Constraints (4.15) and (4.16) imply that the amount of data outflow and, data inflow of a sensor is zero if the sensor is not deployed. Constraints (4.17), (4.18) and (4.19) are included for the purpose of linearization of the product terms $w_t p_{ir}$ by b_{irt}. Observe also that one of the constraints (4.18) and (4.19) becomes redundant depending on the value of the p_{ir} variable. For instance, if $p_{ir} = 0$, constraints (4.19) become redundant, and b_{irt} is set to zero by constraints (4.18) together with the nonnegativity restriction of b_{irt}. Similarly, if $p_{irt} = 1$, constraints (4.18) becomes redundant this time, and constraints (4.19) set w_t as a lower bound for b_{irt} and p_{irt} is equal to w_t by collaborative effort of constraints (4.17) and (4.19). Therefore, in any case b_{irt} represents $w_t p_{ir}$ by constraints (4.17), (4.18) and (4.19) together with the nonnegativity restriction on b_{irt}. Finally, nonnegativity and binary requirements are satisfied by constraints (4.20) and (4.21).

MCSLRP extends the setting of MSLRP such that sensor deployment and multiple mobile sink routing decisions are also determined. Thus, MCSLRP provides a tool for quantifying the effect of integrating the sensor deployment decisions into the WSN design decisions. Note that MSLRP assumes a predefined set of sensor locations which can be suboptimal in terms of the network lifetime. A simple comparison over the optimal objective values of MSLRP with a given set of sensor locations and MCSLRP reveals the positive effect of an integrated approach. However, the determination of sensor locations for MSLRP at the first hand is a matter of question. We assume in our experimental comparative analysis that these sensor locations are determined so

that the total deployment cost is minimized while the coverage requirements of the field is satisfied, and the set of sensors is required to form a connected network for administrative purposes.

4.1.3. Integrating the Coverage, Sink Routing, Data Routing Problems with Activity Scheduling For Heterogeneous WSNs with Multiple Mobile Sinks

We have introduced two MILP models so far. The first one, i.e., MSLRP, integrates Sink Routing Problem (SRP) and Data Routing Problem (DRP) for heterogeneous WSNs with multiple mobile sinks. The second one, i.e., MCSLRP, adds the Coverage Problem (CP) to this integration to obtain a formulation for the Coverage, Sink and Data Routing Problem for heterogeneous WSNs with multiple mobile sinks. All of them are event based, namely clock ticks at an event, which makes the length of the time intervals elapsing between two consecutive events unequal. The next MILP formulation eventually integrates CP, SRP and DRP with activity scheduling for the design of heterogeneous WSNs with multiple mobile sinks (MLSRP) using an event based approach. This is actually the highest level of integration we have achieved so far.

$$\max \sum_{t\in\mathcal{T}} w_t \tag{4.22}$$

s.t.

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + h_r a_{irt} = \sum_{\ell\in\mathcal{N}_{ir}} y_{ir\ell t} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.23}$$

$$\sum_{t\in\mathcal{T}}\left(c^s a_{irt} + c^r \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} c^t_{ij} x_{irjst} + \sum_{\ell\in\mathcal{N}_{ir}} c^t_{i\ell} y_{ir\ell t}\right) \le E_r \qquad i\in\mathcal{S}, r\in\mathcal{R} \tag{4.24}$$

$$\sum_{r\in\mathcal{R}}\sum_{i:\ell\in\mathcal{N}_{ir}} y_{ir\ell t} \le M z_{\ell t} \qquad \ell\in\mathcal{N}, t\in\mathcal{T} \tag{4.25}$$

$$\sum_{\ell\in\mathcal{N}} z_{\ell t} = P \qquad t\in\mathcal{T} \tag{4.26}$$

$$\sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} q_{irt} \geq d_k \qquad k\in\mathcal{K}, t\in\mathcal{T} \tag{4.27}$$

$$\sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{ir}p_{ir} \leq B \tag{4.28}$$

$$q_{irt} \leq p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.29}$$

$$\sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \leq Mq_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.30}$$

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} \leq Mq_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.31}$$

$$a_{irt} \leq w_t \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.32}$$

$$a_{irt} \leq Mq_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.33}$$

$$a_{irt} \geq w_t - M(1-q_{irt}) \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{4.34}$$

$$w_t, a_{irt}, y_{ir\ell t}, x_{irjst} \geq 0 \tag{4.35}$$

$$z_{\ell t}, p_{ir}, q_{irt} \in \{0,1\} \tag{4.36}$$

As can be seen from the formulation of MLSRP, the objective function has the same structure and interpretation with the objective functions of MSLRP and MCSLRP. Constraints (4.23) and (4.24) represent the data flow balance equalities and energy restrictions, respectively. Note that a_{irt} in the formulation of constraints (4.23) and (4.24) is equal to $w_t q_{irt}$. It simply states that if sensor (i,r) is active in period t, it generates data, and consumes energy which has to be considered in data flow balance and energy constraints. Constraints (4.25) say that no sensor-to-sink flow is possible to a point if no sink is placed there. Constraints (4.26) ensure that all of the sinks are located somewhere at each period. Constraints (4.27) are similar to constraints (4.13) of MCSLRP, but p_{ir}'s are replaced with q_{irts}'s since that we require each coverage demand be satisfied throughout the network lifetime. We need such a modification due to the fact that we do not only want to place the sensors but also want to keep a subset of them active to satisfy all the coverage constraints for each period. Constraint (4.28) is the budget constraint stating that total sensor deployment cost cannot exceed the allocated budget. Constraints (4.29) guarantee that a sensor cannot be active if it is not deployed at the first hand. Constraints (4.30) and (4.31) have also similar interpretations to those constraints (4.15) and (4.16) of MSLRP but with p_{ir}'s replacing q_{irt}'s. They simply

state that there can not be any inflow to or outflow from a sensor that is not active (or not deployed at all). Constraints (4.32), (4.33) and (4.34) are responsible for the linearization of the nonlinear terms $a_{irt} = w_t q_{irt}$. Finally, nonnegativity and binary restrictions (4.35) and (4.36) complete the formulation.

MLSRP formulation extends the MSLRP by integrating CP and ASP and extends MCSLRP by integrating ASP. Besides, MLSRP can also be considered as an extension of the model by Türkoğulları *et al.* (2010a) since their model also considers CP, SLP, DRP and ASP with multiple stationary sinks. This extension is a remedy for the "sink neighborhood problem" mentioned before. It is possible to quantify the favorable effect of mobile sinks by simply comparing the optimal lifetime values obtained using both models. However, it should be noted their model incorporate discrete set of periods with fixed lengths which cause the number of binary variables in the model to be very high. We also come up with a variant of their model in Chapter 5 in which we consider stationary sinks but incorporate flexible period lengths by taking them as decision variables.

Because of its level of integration, it may seem that the solution of MLSRP is practically impossible for even small instances. However, as will be demonstrated in Chapter 6, it is possible to obtain good results using MLSRP formulation in a reasonable amount of computational time. This is clearly due to the reduction in the number of binary variables because of the event based formulation approach. In addition, the resulting WSN design has considerably longer lifetime than the ones obtained with the previous MILP formulations as a consequence of the breadth of the integration it involves.

4.2. Solution Strategies

In this section, we propose two heuristics and one exact solution method for MLSRP. Both heuristics try to reduce the number of binary variables in a systematic manner iteratively to ease the solution of the model and use the solution of the previous iterations in the remaining iterations. The first heuristic is called as the Period Itera-

tion Heuristic (PIH) since it reduces the number of periods and increments it for the reduction of binary variables. The latter one is named as the Sequential Assignment Heuristic (SAH) because it assigns the values of the variables sequentially in the order of natural hierarchy existing among the variables. Finally, exact solution method is a branch and price algorithm.

4.2.1. Period Iteration Heuristic

The difficulty of solving realistic sized MLSRP instances arises mostly due to the large number of binary decision variables included in the model. This observation suggests to decrease the number of binary decision variables by fixing a subset of them a priori in the model in such a way that an infeasibility does not occur in the remaining model. Later on, a near optimal solution can be found by much less effort as there are relatively lower number of binary variables involved. Moreover, the obtained solution can also be input as a starting feasible solution of the original model to the state-of-the-art MILP solvers like Gurobi 4.0 (2010) or Cplex 11.0 (2007) in a further attempt to find a better solution.

In order to decrease the number of binary variables of the MLSRP formulation, we concentrate on the total allowable number of periods, T. Observe that the binary variables $z_{\ell t}$ (related to the sojourn locations of the mobile sinks) and q_{irt} (related to the activity schedules of the sensors), and continuous variables w_t (related to the period lengths), $y_{ir\ell t}$ (related to the message flow between sensors and sinks), and x_{irjst} (related to the message flow between sensors) include a period index implying that a reduction in the number of periods would result a noteworthy decrease in the number of decision variables. Therefore, the total number of periods T is an important parameter of the model. If it is set to a very low value such as 1 or 2, the solution of the MLSRP formulation becomes much easier. However, the quality of the resulting solution can be relatively bad. On the other hand, setting T to high values makes the solution of the formulation very difficult. Moreover, it is possible to obtain the maximum network lifetime for a lower number of periods as well. There is an economies of scale between the lifetime and total number of periods: the improvement in the lifetime gradually

decreases with the increasing total number of periods, eventually reaches to zero. In PIH, we try to determine the lowest possible number of periods leading to the maximum network lifetime, namely T^*, by the following period iteration idea.

Let a parameter, say ϕ, represent the number of periods and let its value be 1 at the beginning of the algorithm. This new parameter ϕ serves as a limit on the number of periods so that all the continuous variables w_t, $y_{ir\ell t}$ and x_{irjst} are allowed to take nonzero values only for period indices less than or equal to ϕ, i.e., $t \leq \phi$. In other words, all the w_t, $y_{ir\ell t}$, and x_{irjst} variables are fixed to zero for periods beyond ϕ, i.e., $t > \phi$. On the other hand, values of the binary $z_{\ell t}$ and q_{irt} variables corresponding to periods with $t > \phi$ are forced to take the values of the same variables for period ϕ, i.e., $z_{\ell\phi}$ and $q_{ir\phi}$. This can be achieved easily by simply adding constraints $z_{\ell t} = z_{\ell\phi}$ and $q_{irt} = q_{ir\phi}$ for $t > \phi$ to the model. A simpler approach suggests equating the number of periods T to ϕ, i.e., $T = \phi$, but we choose the former approach in order to obtain a starting feasible solution for the original model with a large number of periods.

After fixing the variables, we end up with a model having a considerably lower number of variables. We call this model as the restricted model in the sequel, which has only the p_{ir} variables related to the sensor location decisions and w_t, $z_{\ell t}$, $y_{ir\ell t}$, and x_{irjst} variables for periods $t \leq \phi$. Therefore, the restricted model is expected to be relatively easier. We run Gurobi to solve the restricted model in a short amount of time and as there are relatively small number of binary variables especially for small ϕ values, the solver is able to find the optimal or good near optimal feasible solutions for the restricted model. Observe also that the feasible solution found for the restricted model is also a feasible solution for the original model.

After finding an optimal or near optimal solution for the restricted model corresponding to a specific value of ϕ, we increase it by 1, i.e., $\phi \leftarrow \phi + 1$, and rerun the solver for the new value of ϕ. Incrementing the value of ϕ by one unit provides a slightly more flexible optimization framework for the solver by letting a larger number of decision variables. Therefore, we expect an improvement in the objective value after each increase of parameter ϕ, namely the network lifetime increases with increasing

ϕ. This approach may seem greedy at the first look, but it should be noted that the restricted model is run not only for the variables corresponding to the last period (i.e., w_ϕ, $z_{\ell\phi}$, $y_{ir\ell\phi}$, $x_{irjs\phi}$), but also for all related variables from the first to the last period (i.e., w_t, $z_{\ell t}$, $y_{ir\ell t}$, x_{irjst} for $t \leq \phi$). Hence, at each iteration, the breadth of the variables included in the restricted model is increased as a consequence of a unit increase in the number of periods and the restricted model is run not only for the last period but also for the previous ones, which avoids PIH from being myopic as greedy algorithms mostly are. However, although an improvement in the network lifetime is expected for larger ϕ values, the solution of the restricted model becomes harder as the number of binary variables of the model increases with ϕ. Fortunately, it is possible to speed up the solution procedure of the new restricted model by using the solution of the previous restricted model as a starting feasible solution. The objective value of the previous solution serves as a lower bound of the optimal objective value of the new restricted model. More interestingly, one can come up with a simple but powerful upper bound for the optimal objective value of the new restricted model as well by making use of the optimal solution and corresponding objective value of the previous restricted model. The idea depends on the following simple observation. The improvement in the network lifetime of the next restricted model cannot exceed the largest period length of the previous optimal solution. This idea is expressed in a more formal way in the proposition given below.

Proposition 1. *Let $O^{(\phi)}$ and $O^{(\phi+1)}$ denote respectively the optimal objective values of the restricted model for parameter values ϕ and $\phi+1$. Besides, let $w_t^{(\phi)}$ represent the length of period t for the optimal solution of the ϕ^{th} restricted model and assume without loss of generality that period lengths corresponding to the ϕ^{th} and $\phi+1^{st}$ restricted models are ordered as $w_1^{(\phi)} \leq w_2^{(\phi)} \leq \cdots \leq w_\phi^{(\phi)}$ in non-decreasing order, and $w_{\phi+1}^{(\phi)} = \cdots = w_T^{(\phi)} = 0$. Similarly, $w_1^{(\phi+1)} \leq w_2^{(\phi+1)} \leq \cdots \leq w_{\phi+1}^{(\phi+1)}$ and $w_{\phi+2}^{(\phi+1)} = \cdots = w_T^{(\phi+1)} = 0$. Then, $O^{(\phi+1)} \leq O^{(\phi)} + w_\phi^{(\phi)}$.*

Proof. First of all, observe that $O^{(\phi)} = \sum_{t=1}^{\phi} w_t^{(\phi)} \geq \sum_{t=2}^{\phi+1} w_t^{(\phi+1)}$ since $O^{(\phi)}$ is the maximum attainable network lifetime within ϕ periods. Here, we claim that $w_\phi^{(\phi)} \geq w_1^{(\phi+1)}$. Suppose $w_\phi^{(\phi)} < w_1^{(\phi+1)}$, then since $w_1^{(\phi)} \leq w_2^{(\phi)} \leq \cdots \leq w_\phi^{(\phi)} < w_1^{(\phi+1)} \leq w_2^{(\phi+1)} \leq$

$\cdots \leq w_{\phi+1}^{(\phi+1)}$, we have $\sum_{t=1}^{\phi} w_t^{(\phi)} < \sum_{t=2}^{\phi+1} w_t^{(\phi+1)}$ which is a contradiction. Therefore, the claim holds. Hence, by simply summing up two inequalities $\sum_{t=2}^{\phi+1} w_t^{(\phi+1)} \leq \sum_{t=1}^{\phi} w_t^{(\phi)}$ and $w_1^{(\phi+1)} \leq w_\phi^{(\phi)}$, we end up with the desired result. □

Proposition 1 provides a tool for determining an upper bound for the optimal objective value of the next restricted model by benefiting from the optimal solution and the corresponding objective value of the current restricted model. Therefore, as we are enclosing the optimal objective value of the next restricted model in a relatively small interval, it is much easier to find its value. However, for large values of parameter ϕ, it is still very difficult to find the optimal value in a reasonable computation time. Therefore, we confine ourselves to good feasible solutions by letting the solver run for some predefined short time rather than expecting the optimal solution at the expense of prohibitive computation times. Note that although it is possible to improve the solution quality by allowing longer solution times in each round of the algorithm, we do not prefer doing this since this also causes slow convergence of the algorithm. Another approach that can be employed is the usage of a heuristic for the solution of the restricted model in each round instead of running Gurobi but we do not need it since PIH is efficient in its current form. Observe that lower bound (LB) and an upper bound (UB) for the optimal objective value still helps the solver to improve the previously computed solution faster. The process continues until the improvement in the objective value after a unit increase in the value of ϕ becomes less than or equal to ϵ, which is a parameter to determine the final precision. Note that the final value of ϕ is an estimator of T^* since we do not wait until finding the optimal solution of the restricted models. Steps of the PIH are formally summarized in Figure 4.1.

4.2.2. Sequential Assignment Heuristic

One may observe a functional hierarchy among the WSN design criteria. For instance, one cannot arrange activity schedules of the sensors without determining the locations of the sensors. Therefore, the determination of the sensor locations is

Let $\phi = 1$, $DIF = 100$ and initiate value of ϵ;
while $(DIF > \epsilon)$ **do**
  if $(\phi > 1)$ **then**
    Set LB $= O^{(\phi-1)}$ and UB $= O^{(\phi-1)} + w_{\phi-1}^{(\phi-1)}$;
  end if
  Let solver run for ϕ^{th} restricted model for a small amount of time, use LB and UB to speed up the solution process
  Set $O^{(\phi)}$ to the best objective value obtained by the solver
  Set $DIF = O^{(\phi)} - O^{(\phi-1)}$; $\phi \leftarrow \phi + 1$
end while
$T^* \leftarrow \phi$
Report the final solution and corresponding objective value

Figure 4.1. Period iteration heuristic.

a prerequisite for the activity schedules of the sensors. Moreover, since the mobile sinks travel for collecting data from the active sensors, the determination of the mobile sink and data routes come after the determination of the activity schedules. Thus, there is a functional hierarchy among the WSN design issues and one may think of making use of it for the reduction of the binary variables in order to obtain easy to solve mathematical programs. As the determination of the sensor locations is a priority, one should determine the best candidate sensor locations initially. The activity schedule of the deployed sensors comes next, and finally mobile sink and data routes are determined.

We present three subproblems for the determination of good WSN designs hierarchically. First, we concentrate on the determination of the sensor locations. This is achieved by the subproblem S1.

S1:

$$\max \sum_{k \in \mathcal{K}} u_k \tag{4.37}$$

s.t.

$$u_k \leq \sum_{r\in\mathcal{R}} \sum_{i:k\in\mathcal{K}_{ir}} p_{ir} \qquad k \in \mathcal{K} \tag{4.38}$$

$$\sum_{r\in\mathcal{R}} \sum_{i:k\in\mathcal{K}_{ir}} p_{ir} \geq d_k + 2 \qquad k \in \mathcal{K} \tag{4.39}$$

$$\sum_{i\in\mathcal{S}} \sum_{r\in\mathcal{R}} f_{ir} p_{ir} \leq B \tag{4.40}$$

$$u_k \geq 0 \qquad k \in \mathcal{K} \tag{4.41}$$

$$p_{ir} \in \{0,1\} \qquad i \in \mathcal{S}, r \in \mathcal{R} \tag{4.42}$$

A new variable u_k, which represents the number of sensors that can cover point k, is defined for all points $k \in \mathcal{K}$. In order to allow additional flexibility for activity scheduling, we try to deploy as much sensors as possible. This is the motivation for maximizing the sum of u_k's in the objective function (4.37). Constraints (4.38) stand for the definition of u_k variables. Constraints (4.39) ensure that there is a sufficient number of sensors located near point k so that at least $d_k + 2$ sensors are available for the surveillance of point k. It should be noted that we require $d_k + 2$ sensors (not d_k) for point k because we try to provide some flexibility for the activity scheduling part of the heuristic for each point $k \in \mathcal{K}$. If there are exactly d_k sensors that can cover point k, the designer would need to keep them active throughout the network lifetime in order to cover that point as much as its quality requirement d_k. This would cause depletion of the batteries of these sensors quickly, and coverage loss at a relatively early stage of the network. Thus, we use $d_k + 2$ instead of d_k in these constraints. During the numerical experiments we observe that requiring more than $d_k + 2$ causes infeasibility for some instances since a limited budget is allocated for sensor deployment. Therefore, for the most of the instances $d_k + 2$ is the largest number that can be chosen without causing infeasibility. However, it could be mentioned that if the budget is very tight, $d_k + 1$ can be used instead of $d_k + 2$. Constraint (4.40) is put for the budget restriction. Finally, constraints (4.41) and (4.42) are usual nonnegativity and binary restrictions.

After determining sensor locations by solving S1, the activity schedules of the deployed sensors are obtained, which is followed by the determination of the sink

locations and data routes. For this purpose, we refer to an iterative procedure that alternates between two subproblems. In the first one, we maximize the network lifetime over the activity schedules of the deployed sensors and data routes for given sink movements, while the maximization is over sink movements and data routes for the given activity schedules of the deployed sensors in the second subproblem. We restrict the solution time of each subproblem by an upper bound and take the best feasible solution. Solutions of both models converge to a point after several iterations. In order to initiate the procedure, we need an initial sink movement scheme which is determined by fixing the sinks at the points from where the highest number of sensors can be directly reachable. It should be noted that the quality of the subproblem solutions can be increased by allowing longer solution times. However, this will slow down the convergence of SAH and the algorithm may end up with a worse final solution since total computation time is limited. One may also refer to heuristic procedures for the solutions of the subproblems instead of using Gurobi but we do not need this since SAH takes only seconds in its current form.

Subproblems can be constructed easily by fixing relevant variables in the MLSRP formulation. For instance, subproblem 2, in which we determine activity schedules of the deployed sensors, is constructed by fixing the locations of the deployed sensors and sink movement decisions in MLSRP. Similarly, subproblem 3 is obtained by fixing sensor places and activity schedules of the deployed sensors in MLSRP. Hence, despite very little changes, formulations of subproblems 2 and 3 are just alike to MLSRP, wherein p_{ir} and z_{lt} are considered as parameters for subproblem 2 (and constraint (7) drops from the MLSRP formulation), while p_{ir} and q_{irt} are treated as parameters in subproblem 3 (and constraint (5) drops from the MLSRP formulation).

Nevertheless, since all of the activity schedule, sink movement and data route decisions have a time index, determining the optimal or near optimal solutions of the subproblems may require very long computation times especially for large instances. In other words, the best feasible solutions of the subproblems found in the allocated running times cannot be very accurate. Hence, if we proceed with a large number of periods, the procedure may end up with a solution of relatively low quality. In order to

improve the efficiency of the procedure, we borrow the period iteration idea from PIH. Namely, we begin the procedure as if the number of periods was one, i.e, $\phi = 1$, and we increase the number of periods by one after the convergence of the inner alternating procedure, until no improvement is observed in the network lifetime. Therefore, the period iteration idea, that is used to find the lowest number of periods leading to the maximum network lifetime, is the backbone of both heuristics. A final note we should point that, after an increase in the number of periods we assume that sinks stay at their final positions during the last period, i.e, $z_{\ell\phi} = z_{\ell(\phi-1)}$ for $\ell \in \mathcal{N}$. The steps of SAH are listed in Figure 4.2 where L_2 and L_3 represent the network lifetimes computed by solving subproblems S2 and S3, respectively, while L is the network lifetime that we report at the end.

Solve subproblem S1 to determine sensor locations (p variables)
Initiate the sink movement decisions (z variables)
$\phi = 1$, $DIF_1 = 100$, and initiate the values of ϵ_1 and ϵ_2
while $(DIF_1 > \epsilon_1)$ **do**
  $DIF_2 = 100$
  while $(DIF_2 > \epsilon_2)$ **do**
    Solve subproblem S2 to obtain q and L_2 for given variables p and z
    Solve subproblem S3 to obtain z and L_3 for given variables p and q
    $DIF_2 = L_3 - L_2$
  end while
  $\phi \leftarrow \phi + 1$, $DIF_1 = L_3 - L$ and $L = L_3$
  Update $z_{\ell\phi} = z_{\ell(\phi-1)}$ for $\ell \in \mathcal{N}$
end while
$T^* \leftarrow \phi$
Report the final solution and corresponding objective value

Figure 4.2. Sequential assignment heuristic.

4.2.3. Branch and Price Algorithm

Branch and Price (BAP) is an exact solution procedure that is used in the exact solution of a wide range of integer programming problems (Barnhart *et al.*, 1998). It

simply offers solving Lagrangean dual by column generation at each node of the branch and bound tree. It is expected to perform better if the Lagrangean dual is efficiently solved but especially when the Lagrangean bound is significantly better than the linear relaxation bound.

In our BAP setting, we first define two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ as

$$\mathcal{X}_1 = \Bigg\{ (p_{ir}, q_{irt}) \in \{0,1\}^2 : \sum_{r\in\mathcal{R}} \sum_{i:k\in\mathcal{K}_{ir}} q_{irt} \geq d_k \quad k \in \mathcal{K}, t \in \mathcal{T}, \\ \sum_{i\in\mathcal{S}} \sum_{r\in\mathcal{R}} f_{ir} p_{ir} \leq B, \quad q_{irt} \leq p_{ir} \quad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \Bigg\},$$

$$\mathcal{X}_2 = \Bigg\{ z_{\ell t} \in \{0,1\} : \sum_{\ell\in\mathcal{N}} z_{\ell t} = P \quad t \in \mathcal{T} \Bigg\},$$

and reformulate MLSRP as:

$$\max \sum_{t\in\mathcal{T}} w_t \tag{4.43}$$

s.t.

$$(4.23) - (4.25), (4.30) - (4.35)$$

$$(p_{ir}, q_{irt}) \in \mathcal{X}_1 \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.44}$$

$$z_{\ell t} \in \mathcal{X}_2 \qquad \ell \in \mathcal{N}, t \in \mathcal{T} \tag{4.45}$$

Since $\mathcal{X}_1$ and $\mathcal{X}_2$ are bounded and consist of binary variables only, the number of elements in $\mathcal{X}_1$ and $\mathcal{X}_2$, say T_1 and T_2, are finite. Suppose we respectively index the elements of $\mathcal{X}_1$ and $\mathcal{X}_2$ by t_1 and t_2, i.e., $(p_{ir}^{t_1}, q_{irt}^{t_1}) \in \mathcal{X}_1$ for $t_1 = 1, \ldots, T_1$ and $z_{\ell t}^{t_2} \in \mathcal{X}_2$ for $t_2 = 1, \ldots, T_2$. Upon this formalization, it is possible to obtain p_{ir} and q_{irt} variables as convex combinations of the members of $\mathcal{X}_1$, and similarly $z_{\ell t}$ can be written as a convex combination of the members of $\mathcal{X}_2$. Mathematically, $p_{ir} = \sum_{t_1=1}^{T_1} \beta^{t_1} p_{ir}^{t_1}$ and $q_{irt} = \sum_{t_1=1}^{T_1} \beta^{t_1} q_{irt}^{t_1}$ where $\sum_{t_1=1}^{T_1} \beta^{t_1} = 1, \ \beta^{t_1} \in \{0,1\}, \ t_1 \in \{1, \ldots, T_1\}$, and $z_{\ell t} = \sum_{t_2=1}^{T_2} \alpha^{t_2} z_{\ell t}^{t_2}$ where $\sum_{t_2=1}^{T_2} \alpha^{t_2} = 1, \quad \alpha^{t_2} \in \{0,1\}, \quad t_2 \in \{1, \ldots, T_2\}$. Now, if we replace p_{ir}, q_{irt} and $z_{\ell t}$ by these definitions in the reformulation of MLSRP given above we obtain the following MIP, which is called Integer Master Problem (IMP):

$$\max \sum_{t\in\mathcal{T}} w_t \tag{4.46}$$

s.t.

$(4.23) - (4.24)$

$$\sum_{r\in\mathcal{R}} \sum_{i:\ell\in\mathcal{N}_{ir}} y_{ir\ell t} \leq M \sum_{t_2=1}^{T_2} \alpha^{t_2} z_{\ell t}^{t_2} \qquad \ell \in \mathcal{N}, t \in \mathcal{T} \tag{4.47}$$

$$\sum_{s\in\mathcal{R}} \sum_{j\in\mathcal{S}_{ir}} x_{irjst} \leq M \sum_{t_1=1}^{T_1} \beta^{t_1} q_{irt}^{t_1} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.48}$$

$$\sum_{s\in\mathcal{R}} \sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} \leq M \sum_{t_1=1}^{T_1} \beta^{t_1} q_{irt}^{t_1} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.49}$$

$$a_{irt} \leq w_t \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.50}$$

$$a_{irt} \leq M \sum_{t_1=1}^{T_1} \beta^{t_1} q_{irt}^{t_1} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.51}$$

$$a_{irt} \geq w_t - M(1 - \sum_{t_1=1}^{T_1} \beta^{t_1} q_{irt}^{t_1}) \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{4.52}$$

$$\sum_{t_1=1}^{T_2} \beta^{t_1} = 1 \tag{4.53}$$

$$\sum_{t_2=1}^{T_2} \alpha^{t_2} = 1 \tag{4.54}$$

$$w_t, a_{irt}, y_{ir\ell t}, x_{irjst} \geq 0 \tag{4.55}$$

$$\beta^{t_1}, \alpha^{t_2} \in \{0, 1\} \tag{4.56}$$

It can be observed that the explicit determination of $(p_{ir}^{t_1}, q_{irt}^{t_1})$ for $t_1 \in \{1, 2, \ldots, T_1\}$ and $z_{\ell t}^{t_2}$ for $t_2 \in \{1, 2, \ldots, T_2\}$ can be impossible or can require prohibitive computation time since T_1 and T_2 can be very large. Hence, instead of constituting all columns of IMP a priori, one may try solving its linear relaxation first with a subset of columns and price out new columns until no column with positive reduced cost is left. We define the linear relaxation of IMP with a subset of its columns as the Restricted Master Problem (RMP). If there are fractional values in the solution of RMP, one may branch on one of them until integrality is obtained. In other words, a branch-and-bound scheme, in each node of which, linear relaxation of IMP with possibly less columns is solved by

column generation, can be devised.

During the pricing phase, the variable having the largest reduced cost value is sought. Recall that the reduced cost of a nonbasic variable x_j can be calculated as $c_j - c_B B^{-1} a_j$ where c_j and a_j are the objective function coefficient and column of the variable x_j, respectively, while $c_B B^{-1}$ is the vector of the dual multipliers. Therefore, if $\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\sigma}, \hat{\beta}, \hat{\alpha}$ are respective dual multipliers of constraints (4.47), (4.48), (4.49), (4.51), (4.52), (4.53) and (4.54) then β and α variables having the maximum reduced cost values can be obtained by solving the following subproblems;

Subproblem 1:

$$\max \sum_{i\in\mathcal{S}} \sum_{r\in\mathcal{R}} \sum_{t\in\mathcal{T}} M\{\gamma_{irt} + \phi_{irt} + \eta_{irt} - \sigma_{irt}\} q_{irt} - \hat{\beta} \tag{4.57}$$

s.t.

$$(4.27) - (4.29)$$

$$p_{ir}, q_{irt} \in \{0, 1\}, \tag{4.58}$$

Subproblem 2:

$$\max \sum_{\ell\in\mathcal{N}} \sum_{t\in\mathcal{T}} M\theta_{\ell t} z_{\ell t} - \hat{\alpha} \tag{4.59}$$

s.t.

$$(4.26)$$

$$z_{\ell t} \in \{0, 1\}. \tag{4.60}$$

It is easy to see that subproblem 2 can be further decomposed into T smaller subproblems for each period $t \in \mathcal{T}$. The formulation of subproblem 2 for a given t can be given as;

Subproblem 2(t):

$$\max \sum_{\ell\in\mathcal{N}} M\theta_{\ell t} z_{\ell t} - \hat{\alpha} \tag{4.61}$$

s.t.

$$(4.26) \text{ for } t \text{ only}$$

$$z_{\ell t} \in \{0,1\} \tag{4.62}$$

Branching can be accomplished either on the binary variables β and α of IMP, or on the original binary variables p_{ir}, q_{irt} and $z_{\ell t}$. We prefer branching on the original variables since as indicated by Barnhart *et al.* (1998), it leads to a more balanced branching tree. Values of the original variables at each node can be restored from the values of the generated β and α values and the variable values obtained from the subproblems. Suppose index set of the generated β and α at the current node are called as $\mathcal{X}_1^c$ and $\mathcal{X}_2^c$, then $p_{ir} = \sum_{t_1 \in \mathcal{X}_1^c} \beta^{t_1} p_{ir}^{t_1}$, $q_{irt} = \sum_{t_1 \in \mathcal{X}_1^c} \beta^{t_1} q_{irt}^{t_1}$ and $z_{\ell t} = \sum_{t_2 \in \mathcal{X}_2^c} \alpha^{t_2} z_{\ell t}^{t_2}$. Finally, suppose p_{ir} (or q_{irt}) variable is set to value i ($i \in \{0,1\}$) during the branching process. Then β^{t_1}, $t_1 \in \mathcal{X}_1^c$ variables for which $p_{ir}^{t_1} \neq i$ (or $q_{irt}^{t_1} \neq i$) is eliminated from the RMP. Similarly, if $z_{\ell t}$ is set to j ($j \in \{0,1\}$), then α^{t_2}, $t_2 \in \mathcal{X}_2^c$ variables for which $z_{\ell t} \neq j$ are eliminated from the RMP. On the other hand, decisions made during the branching process can be easily imposed on p_{ir} and q_{irt} variables of subproblem 1 and $z_{\ell t}$ variables of subproblem 2 by setting their upper bound to zero if the value of the branching variable is 0, or their lower bound to 1 if the value of the branching variable is 1. We let z_{RMP}, $z_{S_1}$and z_{S_2} and respectively denote the optimal objective values of RMP, subproblem 1 and subproblem 2. Additionally, z_{BEST} denote the objective value of the best feasible solution. Similarly, p^*, q^* and z^* are optimal values of p, q and z. Moreover, $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{Z}$ represent the set of p, q and z variables, repectively, for which β and α variables are produced for RMP. In addition, $\bar{p}$, $\bar{q}$ and $\bar{z}$ respectively specify the values of p, q and z which are fixed in the branching process. Finally, we let NL denote the set of nodes which are not explored yet and $\bar{rc}$ represent the maximum reduced cost of the nonbasic columns. Now, based on these formalizations, we summarize the BAP procedure in Figure 4.3.

Unfortunately, BAP algorithm is inefficient for solving MLSRP instances. Hence, we are not able to provide numerical results for BAP algorithm. To be able to find exact optimal solutions for realistic sized MLSRP instances using BAP, it is mandatory to increase its efficiency by, for instance, finding a better reformulation of IMP by relaxing different constraints. Instead, one may try solving RMP with a more efficient

Initiation: $\bar{p}^0_{ir} = null\ \forall i, r$; $\bar{q}^0_{irt} = null\ \forall i, r, t$; $\bar{z}^0_{\ell t} = null\ \forall \ell, t$; $z^0_{RMP} \leftarrow \infty$;
$z_{BEST} \leftarrow 0$; $\mathcal{P} = \mathcal{Q} = \mathcal{Z} = \{\}$; $NL = \{(z^0_{RMP}, (\bar{p}^0, \bar{q}^0, \bar{z}^0))\}$; $\mathcal{X}^c_1 = \mathcal{X}^c_2 = \{\}$;
while $(|NL| \neq 1)$ **do**
  Let $s = argmax_{h \in \{1,2,\ldots,|NL|\}} z^h_{RMP}$, delete $(z^s_{RMP}, (\bar{p}^s, \bar{q}^s, \bar{z}^s))$ from NL
  Discard the columns of RMP corresponding to p, q and z deviating from non-null $\bar{p}^s$, $\bar{q}^s$ and $\bar{z}^s$
  Set $p = \bar{p}^s$, $q = \bar{q}^s$ and $z = \bar{z}^s$ in subproblems for non-null $\bar{p}^s$, $\bar{q}^s$ and $\bar{z}^s$
  $\bar{rc} \leftarrow \infty$
  while $(\bar{rc} > 0)$ **do**
    Solve RMP, update z_{RMP} and $\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\sigma}, \hat{\beta}, \hat{\alpha}$, let $\bar{rc} \leftarrow 0$
    Solve subproblems 1 and 2, obtain z_{S_1}, p^*, q^* and z_{S_2}, z^*
    if $((z_{S_1} > z_{S_2})$ and $(z_{S_1} > 0))$ **then**
      $\bar{rc} \leftarrow z_{S_1}$, $\mathcal{P} = \mathcal{P} \cup \{p^*\}$, $\mathcal{Q} = \mathcal{Q} \cup \{q^*\}$
      Add a column to RMP and to $\mathcal{X}^c_1$ producing from p^* and q^*
    else if $((z_{S_2} > z_{S_1})$ and $(z_{S_2} > 0))$ **then**
      $\bar{rc} \leftarrow z_{S_2}$, $\mathcal{Z} = \mathcal{Z} \cup \{z^*\}$
      Add a column to RMP and to $\mathcal{X}^c_2$ producing from z^*
    end if
  end while
  $p_{ir} = \sum_{t_1 \in \mathcal{X}^c_1} \beta^{t_1} p^{t_1}_{ir}$; $q_{irt} = \sum_{t_1 \in \mathcal{X}^c_1} \beta^{t_1} q^{t_1}_{irt}$; $z_{\ell t} = \sum_{t_2 \in \mathcal{X}^c_2} \alpha^{t_2} z^{t_2}_{\ell t}$;
  if $(z_{RMP} > z_{BEST})$ **then**
    if (there are fractional p, q and z values) **then**
      Find p_{ir}, q_{irt}, $z_{\ell t}$ value that is closest to 0.5 (for instance, say $p_{\hat{i}\hat{r}}$)
      $\bar{p}^s_{\hat{i}\hat{r}} \leftarrow 0$, $NL = NL \cup \{(z_{RMP}, (\bar{p}^s, \bar{q}^s, \bar{z}^s))\}$
      $\bar{p}^s_{\hat{i}\hat{r}} \leftarrow 1$, $NL = NL \cup \{(z_{RMP}, (\bar{p}^s, \bar{q}^s, \bar{z}^s))\}$
    else
      $z_{BEST} \leftarrow z_{RMP}$
    end if
  end if
end while

Figure 4.3. Branch and Price algorithm.

technique such as weighted Dantzig-Wolfe algorithm, or subgradient algorithm, or a hybrid algorithm combining Dantzig-Wolfe and subgradient algorithms.

5. STATIONARY SINK

In this chapter,[3] we first provide an MILP model with multiple stationary sinks. The model integrates CP, ASP and DRP with SLP. Later on, we provide an efficient metaheuristic strategy for its solution. We experimentally demonstrate the success of the proposed solution strategy in Chapter 6.

5.1. Integrating the Coverage, Sink Location, Routing Problems and Activity Scheduling For Heterogeneous WSNs with Stationary Sinks

The model presented in this section integrates the three fundamental subproblems CP, SLP and DRP with sensor activity scheduling to maximize the network lifetime, and thus may seem conceptually similar to the one by Türkoğulları *et al.* (2009), which is also concerned with the solution of the sensor and sink Location, Scheduling and data Routing Problem (LSRP) for the design of heterogeneous WSNs with stationary sinks. However, we use an event based approach and define the length of each period (time elapsed between two consecutive events) as a decision variable whose optimal value is to be determined. This makes our model considerably more efficient compared to the models of Türkoğulları *et al.* (2009), where time is discretized into finitely many countable intervals of fixed length. Since most of the variables and constraints have time index, letting period lengths to be optimally determined by the model helps to improve its efficiency by decreasing the number of variables and constraints of the model significantly. The formulation of LSRP is given below:

$$\max \sum_{t\in\mathcal{T}} w_t \tag{5.1}$$

s.t.

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + h_r a_{irt} = \sum_{\ell\in\mathcal{N}_{ir}} y_{ir\ell t} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \tag{5.2}$$

[3]The work by Keskin *et al.* (2013c) is based on the content of this chapter.

$$\sum_{t\in\mathcal{T}}\Bigg(c^s a_{irt} + c^r \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt}$$

$$+\sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} c^t_{ij}x_{irjst} + \sum_{\ell\in\mathcal{N}_{ir}} c^t_{i\ell}y_{ir\ell t}\Bigg) \le E_r \qquad i\in\mathcal{S}, r\in\mathcal{R} \qquad (5.3)$$

$$\sum_{r\in\mathcal{R}}\sum_{i:\ell\in\mathcal{N}_{ir}} y_{ir\ell t} \le M z_\ell \qquad \ell\in\mathcal{N}, t\in\mathcal{T} \qquad (5.4)$$

$$\sum_{\ell\in\mathcal{N}} z_\ell = P \qquad (5.5)$$

$$\sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} q_{irt} \ge d_k \qquad k\in\mathcal{K}, t\in\mathcal{T} \qquad (5.6)$$

$$\sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{ir}p_{ir} \le B \qquad (5.7)$$

$$q_{irt} \le p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.8)$$

$$\sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \le M q_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.9)$$

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} \le M q_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.10)$$

$$a_{irt} \le w_t \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.11)$$

$$a_{irt} \le M q_{irt} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.12)$$

$$a_{irt} \ge w_t - M(1-q_{irt}) \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.13)$$

$$w_t, a_{irt}, y_{ir\ell t}, x_{irjst} \ge 0 \qquad (5.14)$$

$$z_\ell, p_{ir}, q_{irt} \in \{0,1\} \qquad (5.15)$$

As can be seen from the formulation of LSRP, the objective function and the majority of its constraints are presented exactly in their forms given in the MSLRP, MCSLRP and MLSRP formulations. They do have the same interpretations as well. Hence, we do not explain them one more time. The difference between the models MLSRP and LSRP is that the sinks are stationary in LSRP while they are mobile in MLSRP, the sink routing decisions $z_{\ell t}$ lack time period indices t in LSRP and becomes z_ℓ. They appear in this form in the relevant constraints (5.4) and (5.5).

LSRP is a bulky model since it possesses a high level of integration. Hence, finding the optimal solution of the realistic sized LSRP instances is either impossible

or requires very long computation times. Instead, one has to refer to heuristic solution strategies which usually promise to find solutions of better quality in lower amount of computation time for hard problems. In the next section, we devise a hybrid heuristic algorithm for the solution of LSRP.

5.2. Solution Strategies

Computational time that the exact solution of LSRP requires can be prohibitively large for even small and moderate size instances. One reason behind this difficulty is the large number of binary decision variables included in the formulation of LSRP. This observation suggests decreasing the number of binary decision variables through an iterative procedure gradually fixing some of them at each iteration and treating as parameters in the succeeding ones. After fixing some of the variables, the model that is left behind, which we name as the *restricted model*, can be solved by means of an MILP solver or a heuristic by spending much less computational effort as the number of binary variables it has is considerably lower. Having observed these facts, we propose a Hybrid Heuristic (HH) as the solution method. It consists of a Simulated Annealing search over the candidate sink places combined with a Lagrangean Heuristic (LH) computing a solution of the restricted model with respect to the decision variables involving in sensor placement, activity scheduling and data routing. We elaborate the details of the HH and LH in the following subsections.

5.2.1. Hybrid Heuristic

Simulated Annealing (Kirkpatrick *et al.*, 1983) is a highly reliable and commonly used metaheuristic for solving hard combinatorial optimization problems. We employ simulated annealing to search over the candidate sink locations. After locating the sinks at a configuration, sink location variables corresponding to these locations are set to one otherwise they are set to zero in LSRP. We also devise a Lagrangean heuristic LH for the solution of the restricted model obtained by setting the values of the sink location variables. The maximum network lifetime obtained by running LH is used to measure the quality of the sink positions.

Nevertheless, LH usually generates relatively low quality solutions since the restricted model can still have numerous binary variables especially for very large instances. In order to overcome this difficulty and improve the performance of the Lagrangean heuristic, we borrow the period iteration idea which is successfully employed in PIH. The period iteration approach proposes to increase the number of periods one by one starting from one until there is no improvement in the network lifetime. In order to devote large enough computation time for simulated annealing to search over the candidate sink locations, we employ period iteration idea only when we observe that the evaluated neighboring sink location is the best one so far.

We use a move operation while obtaining neighboring solutions of the current sink locations. During the move operation, one of the sinks is randomly selected and moved from its current location to one of the unoccupied candidate sink locations, which is also randomly chosen. At the beginning of HH we initially locate sinks at candidate locations which is directly reachable by the largest possible unserved sensors. Here, sensor (i, r) is considered as unserved if it cannot directly reach to a sink. This procedure is repeated until the sinks are located. Before giving the pseudo-code of HH, we introduce the mathematical notation used in it. First of all, ϕ_z and $\phi_{ITERBEST}$ denote respectively the maximum network lifetime obtained by the Lagrangean heuristic in the allowed computation time for the given sink locations (i.e., the values of $\mathbf{z}$ vector) and the maximum network lifetime achieved in the current period iteration step. Similarly, $temp_0$ and $temp$ are respectively the initial and current temperature parameter values. Variable $iterlim$ counts the number of moves for a given temperature value while $bestiterlim$ is used in the stopping criterion together with the allowed total computation time. Namely, if the best network lifetime is not updated throughout the $bestiterlim$ number of temperatures or the total computation time exceeds the allowed computation time, the algorithm stops and reports the best sink values, $\mathbf{z}_{BEST}$, and corresponding objective function values, ϕ_{BEST}. Finally, $temprate$ is the constant used for temperature updates and $R(0, 1)$ denotes a real number randomly chosen from interval $[0, 1]$. The pseudocode is given as Figure 5.1 in the following.

$firstcheck = true$, $temp = temp_0$, $T = 1$, initiate z values, $z_{BEST} = z$

Employ Lagrangean Heuristic (LH) for given $\mathbf{z}$ values to obtain ϕ_z, let $\phi_{BEST} = \phi_z$

while ($firstcheck = true$ and allowed computation time is not exceeded) **do**

  Let $iter = 0$

  while ($iter < iterlim$) **do**

    Employ move operation to obtain neighboring $\mathbf{z}'$

    Employ LH for $\mathbf{z}'$ and T to find $\phi_{z'}$

    $\phi_{ITERBEST} = \phi_{BEST}$

    while ($\phi_{z'} > \phi_{ITERBEST}$) **do**

      $\phi_{ITERBEST} = \phi_{z'}$, $T \leftarrow T + 1$

      Employ LH with for $\mathbf{z}'$ and T to find $\phi_{z'}$

    end while

    if ($\phi_{z'} > \phi_z$) or if ($e^{(\phi_{z'} - \phi_z)/temp} > R(0,1)$), $\mathbf{z} = \mathbf{z}'$ and $\phi_z = \phi_{z'}$

    if ($\phi_z > \phi_{BEST}$), $\mathbf{z}_{BEST} = \mathbf{z}$ and $\phi_{BEST} = \phi_z$

    $iter \leftarrow iter + 1$

  end while

  $temp \leftarrow temp \times temprate$

  if ($\mathbf{z}_{BEST}$ is not updated through $bestiterlim$ temperatures), $firstcheck = false$

end while

Report the best sink locations, $\mathbf{z}_{BEST}$, and corresponding objective function value, ϕ_{BEST}

Figure 5.1. Hybrid heuristic.

5.2.2. Lagrangean Heuristic

In this section, we provide the details of LH that calculates a solution of the restricted model obtained by fixing the sink variables of LSRP. After the determination of sink locations, z_ℓ variables become parameters and constraint (5.5) can be dropped from the formulation of LSRP. It should be noted that there may be several ways of constructing Lagrangean relaxation scheme depending on the constraint set chosen for relaxation. We employ several Lagrangean schemes and the one explained below produces the longest feasible network lifetimes on the average.

5.2.2.1. Using Lagrangean Relaxation for Solving the Restricted Model. We relax constraints (5.6), (5.7), (5.9), (5.10), (5.12) and (5.13) with the intention of decomposing the model into two subproblems one of which has only continuous variables while the other one has only integer variables. We carry the relaxed constraints to the objective function by multiplying with the appropriate Lagrange multipliers $\boldsymbol{\mu}$, δ, $\boldsymbol{\lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}^1$ and $\boldsymbol{\gamma}^2$. We let LSRP$(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ denote the Lagrangean subproblem

$$
\begin{aligned}
&\text{LSRP}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2): \\
&\quad \max \sum_{t\in\mathcal{T}} w_t + \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \mu_{kt}\Big(\sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} q_{irt} - d_k\Big) + \delta\Big(B - \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{ir}p_{ir}\Big) \\
&\qquad + \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} \left\{ \lambda_{irt}\Big(Mq_{irt} - \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst}\Big) + \beta_{irt}\Big(Mq_{irt} - \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt}\Big) \right\} \\
&\qquad + \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} \left\{ \gamma^1_{irt}(Mq_{irt} - a_{irt}) + \gamma^2_{irt}(a_{irt} - w_t + M(1 - q_{irt})) \right\} \qquad (5.16) \\
&\quad \text{s.t.} \;\; (5.2) - (5.4), (5.8), (5.11), (5.14), (5.15).
\end{aligned}
$$

Observe that the new objective function (5.16) is not only a function of the original variables $(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q})$ but also a function of the multipliers $(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$. In order to reflect this, we let $f(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ denote the objective function (5.16). We also let $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ denote the optimum value of the LSRP$(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ for given values of the multipliers $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$. Suppose $\mathcal{M}$ denotes the set of all feasible solutions of the Lagrangean subproblem LSRP$(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1,$

$\boldsymbol{\gamma}^2$). In other words, each $(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q})$ variable set satisfying constraints (5.2)-(5.4), (5.8), (5.11), (5.14) and (5.15) is a member of $\mathcal{M}$. Upon this formalization, we can write the equality

$$\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) = max\{f(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) : (\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}) \in \mathcal{M}\}. \tag{5.17}$$

Lagrangean subproblem LSRP$(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ can be decomposed into two subproblems since each constraint is either a function of only continuous variables or only binary variables. Similarly, we can also separate the objective function into two parts so that the terms in the first one contain only continuous variables and the terms in the second one have only binary variables. Terms of the objective function including only the continuous variables together with the constraints with only continuous variables form a linear programming (LP) subproblem which is easy to solve, while the rest of the objective function and the rest of the constraints constituting a binary integer programming (BIP) subproblem. We name the LP subproblem as LSRP$_1(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, which includes the multipliers $\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$, and the BIP subproblem as LSRP$_2(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, which includes all multipliers. There also remains some terms of the objective function which do not include original variables but only multipliers, they are considered as constants for given multiplier values. The subproblems are formulated as,

LSRP$_1(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$:

$$\max \sum_{t\in\mathcal{T}}(1 - \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\gamma^2_{irt})w_t$$
$$- \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}}\left(\lambda_{irt}\sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} + \beta_{irt}\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + (\gamma^1_{irt} - \gamma^2_{irt})a_{irt}\right) \tag{5.18}$$

s.t.

$$\sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} + h_r a_{irt} = \sum_{\ell\in\mathcal{N}_{ir}} y_{ir\ell t} + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} x_{irjst} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.19}$$

$$\sum_{t\in\mathcal{T}}\left(c^s a_{irt} + c^r \sum_{s\in\mathcal{R}}\sum_{j:i\in\mathcal{S}_{js}} x_{jsirt} \right.$$

$$\left. + \sum_{s\in\mathcal{R}}\sum_{j\in\mathcal{S}_{ir}} c^t_{ij} x_{irjst} + \sum_{\ell\in\mathcal{N}_{ir}} c^t_{i\ell} y_{ir\ell t}\right) \le E_r \qquad i\in\mathcal{S}, r\in\mathcal{R} \qquad (5.20)$$

$$\sum_{r\in\mathcal{R}}\sum_{i:\ell\in\mathcal{N}_{ir}} y_{ir\ell t} \le M z_\ell \qquad \ell\in\mathcal{N}, t\in\mathcal{T} \qquad (5.21)$$

$$a_{irt} \le w_t \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.22)$$

$$w_t, a_{irt}, x_{irjst}, y_{ir\ell t} \ge 0 \qquad i,j\in\mathcal{S}, r,s\in\mathcal{R}, \ell\in\mathcal{N}, t\in\mathcal{T}, \qquad (5.23)$$

and

$\mathrm{LSRP}_2(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$:

$$\max \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \mu_{kt} \sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} q_{irt} - \delta \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} b_{ir} p_{ir}$$

$$+ M \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} \left(\lambda_{irt} + \beta_{irt} + \gamma^1_{irt} - \gamma^2_{irt}\right) q_{irt} \qquad (5.24)$$

s.t.

$$q_{irt} \le p_{ir} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T} \qquad (5.25)$$

$$p_{ir}, q_{irt} \in \{0,1\} \qquad i\in\mathcal{S}, r\in\mathcal{R}, t\in\mathcal{T}. \qquad (5.26)$$

We let $\mathcal{Z}_1(\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ denote the optimum value of $\mathrm{LSRP}_1(\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ for given values of the multipliers $\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2$. Similarly, $\mathcal{Z}_2(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ denotes the optimum value of $\mathrm{LSRP}_2(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ for given values of the multipliers $\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2$. As can be observed, it is possible to further decompose $\mathrm{LSRP}_2(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ into smaller subproblems $\mathrm{LSRP}_{2ir}(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$ for each $i\in\mathcal{S}, r\in\mathcal{R}$ as

$\mathrm{LSRP}_{2ir}(\boldsymbol{\mu},\delta,\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{\gamma}^1,\boldsymbol{\gamma}^2)$:

$$\max \sum_{t\in\mathcal{T}} \alpha_{irt} q_{irt} - \delta b_{ir} p_{ir} \qquad (5.27)$$

s.t.

$$q_{irt} \le p_{ir} \qquad t\in\mathcal{T} \qquad (5.28)$$

$$p_{ir}, q_{irt} \in \{0,1\} \qquad\qquad t \in \mathcal{T}. \qquad\qquad (5.29)$$

Here, $\alpha_{irt} = M(\lambda_{irt} + \beta_{irt} + \gamma^1_{irt} - \gamma^2_{irt}) + \sum_{k \in \mathcal{K}_{ir}} \mu_{kt}$. We observe that these small subproblems can be solved optimally by inspection. Now suppose that $\mathcal{Z}_{2ir}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ is the optimum value of the subproblem $\text{LSRP}_{2ir}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$. Moreover, let $\mathcal{Z}_3(\boldsymbol{\mu}, \delta, \boldsymbol{\gamma}^2)$ represent the constant part of the objective function, which is equal to $\delta B - \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} \mu_{kt} d_k + M \sum_{i \in \mathcal{S}} \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} \gamma^2_{irt}$. Then, the optimum value of $\text{LSRP}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ can be found by summing up the optimal values of the subproblems $\text{LSRP}_1(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ and $\text{LSRP}_2(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ with the constant part. In other words, $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) = \mathcal{Z}_1(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) + \mathcal{Z}_2(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) + \mathcal{Z}_3(\boldsymbol{\mu}, \delta, \boldsymbol{\gamma^2})$ with $\mathcal{Z}_2(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) = \sum_{i \in \mathcal{S}} \sum_{r \in \mathcal{R}} \mathcal{Z}_{2ir}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$. As a final remark, one can easily see that the optimum value of $\text{LSRP}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, namely $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, always forms an upper bound for the optimum value of LSRP for any given values of the multipliers $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1$ and $\boldsymbol{\gamma}^2$. In the next section, we explain the approach we use in order to determine optimal multiplier values, namely to solve the Lagrangean dual problem.

5.2.2.2. The Determination of the Best Values of the Lagrange Multipliers. Optimal values of the multipliers are obtained by solving the Lagrangean Dual problem (LD)

$$\text{LD}: \qquad \min\{\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) : \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2 \geq 0\}. \qquad (5.30)$$

After replacing $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ with $max\{f(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) : (\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}) \in \mathcal{M}\})$, we can re-express LD as

$$\min\{\max\{f(\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) : (\mathbf{w}, \mathbf{a}, \mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}) \in \mathcal{M}\}$$
$$: \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2 \geq 0\}. \qquad (5.31)$$

Now, suppose $I_{\mathcal{M}}$ is the index set for which for all $n \in I_{\mathcal{M}}$, $(\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)}) \in \mathcal{M}$. Then, LD becomes

$$\min\{\max\{f(\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) : n \in I_M\}$$

$$: \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2 \geq 0\}, \tag{5.32}$$

from which, the LP, which we call as Master Problem (MP)

$$\begin{aligned}
\text{MP} : Z_{MP} = & \min u_0 \\
& \text{s.t.} \\
& u_0 \in \Re & (5.33)\\
& \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2 \geq 0 & (5.34)\\
& u_0 \geq f(\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) \quad n \in I_{\mathcal{M}} & (5.35)
\end{aligned}$$

follows. Here, Z_{MP} denotes its optimum value.

There can be an enormous number of feasible solutions in $\mathcal{M}$ implying that the number of constraints of type (5.35) can be prohibitively large. Moreover, it can be impossible to comprise all solutions from $\mathcal{M}$ in a reasonable amount of computation time. Therefore, instead of taking all constraints of type (5.35), one can start MP with a subset of them. For instance, we can construct constraints of type (5.35) not for the whole set $I_{\mathcal{M}}$ but for a subset of it, say $I_{\mathcal{RM}}$. By doing this we end up with a mathematical optimization formulation that is of the same form with MP, but with less constraints; it is named as Restricted Master Problem (RMP). Observe that the optimum value of RMP, say $\mathcal{Z}_{RMP}$, is less than or equal to $\mathcal{Z}_{MP}$ since RMP is a relaxation of MP. On the other hand, optimum value of LSRP($\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$) is an upper bound for $\mathcal{Z}_{MP}$ for any value of the multipliers $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$, i.e., $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2) \geq \mathcal{Z}_{MP}$. Therefore, we have a mechanism which generates lower and upper bounds for the minimum possible Lagrangean dual value, $\mathcal{Z}_{MP}$.

In order to improve the lower and upper bounds of $\mathcal{Z}_{MP}$, which are respectively $\mathcal{Z}_{RMP}$ and $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, we adopt the weighted Dantzig-Wolfe column generation method introduced by Wentges (1997) and successfully employed by Klose and Görtz (2007) and by Görtz and Klose (2009) for the capacitated facility location prob-

lem. Each step of the procedure consists of two parts. In the first one we solve RMP to obtain current multiplier values, take a convex combination of the current multipliers with the best known multiplier values, and pass the resulting values to the subproblems to be used as input parameters. In the second part, subproblems are solved to optimality with the resulting multiplier values, optimal values are summed up together with the constant part and if the resulting $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ value exceeds $\mathcal{Z}_{RMP}$ value, a new constraint of the form $u_0 \geq f(\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)}, \boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ is added to RMP. In other words, we let $I_{\mathcal{RM}} = I_{\mathcal{RM}} \cup \{n\}$ where $(\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)})$ values come from an optimal solution of the subproblems at the n^{th} iteration. An observation that is worth mentioning here is that the lower bound values, $\mathcal{Z}_{RMP}$, always increases by the increase in the number of constraints since the optimum value cannot decrease by an additional constraint in a minimization problem. Therefore, $\mathcal{Z}_{RMP}$ always reflects the best lower bound value at each step. On the other hand, although it generally decreases, the value of $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ may fluctuate depending on the values of the multipliers. Hence, the best upper bound is the lowest $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$ value obtained and is represented by $\mathcal{Z}_{BEST}$. Similarly, best multipliers are selected as the ones leading to $\mathcal{Z}_{BEST}$. In order to speed up the convergence of the algorithm, the values of the multipliers are not directly taken from the solutions of RMP but a convex combination of the current and the best ones is used. Coefficients of the current and best multipliers are respectively selected as $\theta = 1/min\{50, 0.5 \times (n + NI)\}$ and $1 - \theta$ as suggested in Wentges (1997) while n is the number of iterations of the weighted Dantzig-Wolfe decomposition algorithm and NI is the number of improvements of the best upper bound. By increasing the coefficient of the best multipliers with the iteration number and the number of updates of the best multipliers, we force the best multiplier values to converge. However, in order to let the effect of the current multiplier values to be larger than a threshold, we choose $1/50$ as the coefficient of the current multipliers instead of $1/0.5 \times (n + NI)$ after $0.5 \times (n + NI)$ becomes too much, i.e., $0.5 \times (n + NI) > 50$.

At the initial step, Lagrange multipliers are set to the optimal dual values of the corresponding constraints of the linear relaxation of LSRP. As the procedure continues between RMP and the subproblems, lower and upper bounds get closer and closer until

they finally converge to each other. At that point the algorithm terminates. In other words, iterations continue until $\mathcal{Z}_{BEST}$ value corresponding to the newly calculated multipliers is less than or equal to optimal u_0 value (i.e., $\mathcal{Z}_{RMP}$) of the RMP. Another approach is to stop the algorithm after a determined number of iterations is reached, or when the difference between $\mathcal{Z}_{BEST}$ and $\mathcal{Z}_{RMP}$ becomes less than or equal to a specified epsilon value. At the final stage of the iterations, we obtain the value of the minimum Lagrangean dual value, or report the $\mathcal{Z}_{RMP}$ together with the best multiplier values. This procedure is summarized as Figure 5.2.

5.2.2.3. Obtaining Feasible Solutions. At each iteration of LH explained in the previous subsection, subproblems are solved to optimality and sensor places (i.e., $\mathbf{p}$) and activity schedules of the sensors (i.e., $\mathbf{q}$) are obtained. Values of these variables do not have to possess a feasible solution together with the sink locations (i.e., $\mathbf{z}$) for LSRP model since subproblems do not force them to obey some of the constraints of LSRP. In particular, $\mathbf{p}$ and $\mathbf{q}$ values can violate coverage constraints (5.6) and budget restriction (5.7). Moreover, even if they satisfy (5.6) and (5.7), active sensor set at some period can be disconnected and there may be some sensors which are not able to send their data to one of the sinks. In order to determine a feasible solution by making use of the calculated $\mathbf{p}$ and $\mathbf{q}$ values, we construct a new mathematical optimization model in which we minimize the total number of changes in the values of $\mathbf{p}$ and $\mathbf{q}$ variables while forcing the budget and coverage requirements. We use four different variable set in this model, which are fp_{ir} denoting the location of the sensors, cp_{ir} showing whether there is a change in the value of p_{ir}, fq_{irt} responsible for the activity of sensor (i, r) at period t and cq_{irt} reflects the change in the value of q_{irt}. We name the model as the Feasibility Model (FM) and give its formulation as

FM:

$$\min \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} cp_{ir} + \sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} cq_{irt} \tag{5.36}$$

s.t.

$$\sum_{r\in\mathcal{R}}\sum_{i:k\in\mathcal{K}_{ir}} fq_{irt} \geq d_k \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{5.37}$$

$$\sum_{i\in\mathcal{S}}\sum_{r\in\mathcal{R}} f_{ir} fp_{ir} \leq B \tag{5.38}$$

$$fq_{irt} \leq fp_{ir} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.39}$$

$$cp_{ir} \geq p_{ir} - fp_{ir} \qquad i \in \mathcal{S}, r \in \mathcal{R} \tag{5.40}$$

$$cp_{ir} \geq fp_{ir} - p_{ir} \qquad i \in \mathcal{S}, r \in \mathcal{R} \tag{5.41}$$

$$cq_{irt} \geq q_{irt} - fq_{irt} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.42}$$

$$cq_{irt} \geq fq_{irt} - q_{irt} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.43}$$

$$cp_{ir}, cq_{irt} \geq 0 \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.44}$$

$$fp_{ir}, fq_{irt} \in \{0,1\} \qquad i \in \mathcal{S}, r \in \mathcal{R}, t \in \mathcal{T} \tag{5.45}$$

Note that p_{ir} and q_{irt} values are calculated by solving the subproblems and considered as input parameters in the formulation FM. In order to make p_{ir} and q_{irt} variables feasible with respect to the budget and coverage restrictions, we add constraints (5.37) and (5.38) to FM. Constraint (5.39) ensures that no sensor can be active at a period without being located at the first hand. Constraint (5.40) and (5.41) are responsible from the definition of variable cp_{ir}. If there is a change in the value of p_{ir}, then either the difference $p_{ir} - fp_{ir}$ or $fp_{ir} - p_{ir}$ take value 1. Hence, the change in the value of p_{ir}, which is cp_{ir}, should be 1. We force cp_{ir} to be larger than both of the differences $p_{ir} - fp_{ir}$ and $fp_{ir} - p_{ir}$ by constraints (5.40) and (5.41) implying that any change in the value of p_{ir} will cause cp_{ir} to be at least 1. In such a case, the value of cp_{ir} will equate to 1 in order not to increase the objective function value (5.36). Similarly, the definition of cq_{irt} is provided by constraints (5.42) and (5.43) and explanations of these constraints are similar. In order to find the solution of FM, we allow Gurobi 4.0 (2010) to run for a limited amount of time but we observe that Gurobi is able to find an optimal solution of FM at all steps of the algorithm for any one of the test instances. Resulting fp_{ir} and fq_{irt} variable values are then set as the values of p_{ir} and q_{irt}.

Sensor places and sensor schedules obtained from an optimal solution of FM together with the sink locations simulated annealing search determines do not necessarily posses a feasible network. In other words, fixing $\mathbf{z}$ values coming from simulated an-

Initiate multipliers $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$, initiate ϵ

Let $I_{\mathcal{RM}} = \{\emptyset\}$, $\mathcal{Z}_{RMP} = -\infty$, $\mathcal{Z}_{BEST} = \infty$, $NI = 1$, $n = 1$

Solve subproblems to obtain $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, and $\mathbf{w}^{(n)}, \mathbf{a}^{(n)}, \mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{p}^{(n)}, \mathbf{q}^{(n)}$

Let $\boldsymbol{\mu}_{BEST}, \delta_{BEST}, \boldsymbol{\lambda}_{BEST}, \boldsymbol{\beta}_{BEST}, \boldsymbol{\gamma}^1_{BEST}, \boldsymbol{\gamma}^2_{BEST}$ be respectively equal to $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$

$\mathcal{Z}_{BEST} = \mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$

$NI = NI + 1$, $\theta = 1/min\{50, 0.5 \times (n + NI)\}$

while $(\mathcal{Z}_{RMP} + \epsilon < \mathcal{Z}_{BEST})$ AND (time limit is not exceeded) **do**

Let $I_{\mathcal{RM}} = I_{\mathcal{RM}} \cup \{n\}$

Solve RMP for $I_{\mathcal{RM}}$ to obtain $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$

Update the multipliers as follows;

$\boldsymbol{\mu} = \theta \times \boldsymbol{\mu} + (1 - \theta) \times \boldsymbol{\mu}_{BEST}$, $\delta = \theta \times \delta + (1 - \theta) \times \delta_{BEST}$, $\boldsymbol{\lambda} = \theta \times \boldsymbol{\lambda} + (1 - \theta) \times \boldsymbol{\lambda}_{BEST}$, $\boldsymbol{\beta} = \theta \times \boldsymbol{\beta} + (1 - \theta) \times \boldsymbol{\beta}_{BEST}$, $\boldsymbol{\gamma}^1 = \theta \times \boldsymbol{\gamma}^1 + (1 - \theta) \times \boldsymbol{\gamma}^1_{BEST}$, $\boldsymbol{\gamma}^2 = \theta \times \boldsymbol{\gamma}^2 + (1 - \theta) \times \boldsymbol{\gamma}^2_{BEST}$

Solve subproblems to obtain $\mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$, and $\mathbf{w}^{(n)}$,$\mathbf{a}^{(n)}$,$\mathbf{x}^{(n)}$,$\mathbf{y}^{(n)}$,$\mathbf{p}^{(n)}$,$\mathbf{q}^{(n)}$

if $(\mathcal{Z}_{BEST} > \mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2))$ **then**

$\mathcal{Z}_{BEST} = \mathcal{Z}(\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2)$

Let $\boldsymbol{\mu}_{BEST}, \delta_{BEST}, \boldsymbol{\lambda}_{BEST}, \boldsymbol{\beta}_{BEST}, \boldsymbol{\gamma}^1_{BEST}, \boldsymbol{\gamma}^2_{BEST}$ be respectively equal to $\boldsymbol{\mu}, \delta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$

$NI = NI + 1$

end if

$\theta = 1/min\{50, 0.5 \times (n + NI)\}$

$n \leftarrow n + 1$

end while

Figure 5.2. Lagrangean heuristic.

nealing search, **p** and **q** values obtained using LH and solving FM can still be infeasible. If the resulting network after fixing **z**, **p** and **q** is infeasible, that means there is at least one active sensor at some period which cannot reach to one of the sinks. This is because the flow balance constraints (5.2) implicitly enforce a connected network in the sense that each sensor should be able to send its data to at least one of the sinks. In such a situation, another MILP model similar to LSRP is constructed in which already located sensor places, i.e., p_{ir} variables that are set to 1, active sensor set, i.e., q_{irt} that are set to 1, and sink locations, i.e., z_ℓ variables that are set to 1, are fixed and the remaining **p** and **q** values are considered as variables. Connectivity constraints are in the form of flow balance constraints together with the energy constraints while the network lifetime is maximized in the objective function. The model is run for a limited time and if no feasible solution is obtained, then we end up without any feasible solution of LSRP for the current iteration of LH. However, if a feasible solution is obtained, it is also feasible for LSRP since it satisfies the coverage, budget and connectivity requirements.

6. COMPUTATIONAL RESULTS

In this chapter, we first explain how we determine the model parameter values and the steps of problem instance generation. We generate two different sensor field topologies. In the first one, locations of the sensors are randomly determined on the two dimensional plane so that the resulting network is connected. In the latter one, candidate sensor locations are determined in a way that they form a grid. The first topology is used for analyzing the effect of consideration of nonzero sink travel times in UHM, LHM and MSM models. Empirical evaluation of integration of the major WSN design isuses such as CP, SLP, SRP, DRP and ASP are also accomplished by comparing objective function values of MSLRP, MCSLRP, LSRP and MSLRP on the problem instances generated based on the first topology. Besides, success of the PIH and SAH heuristics developed for the solution of MSLRP, and the effectiveness of HH implemented for the solution of LSRP are demonstrated on the problem instances that are generated based on the second sensor field topology.

On the other hand, note that the performance of the heuristics are evaluated by comparing the best network lifetimes achieved by the heuristics and by the state-of-the-art MILP solver such as Cplex 11.0 (2007) and Gurobi 4.0 (2010) within a given upper bound on the running time. We report results for only three-hour upper bound since we observe that the quality of the solutions is not affected by larger values. We refer to Cplex 11.0 (2007) for measuring the performance of TCH, TIH and while dealing with the solution of MSM and we use Gurobi 4.0 (2010) as the MILP solver for all other computations. We implemented all the codes for MILP models and for all heuristics in a Visual Studio environment with C# language and all of the experiments are carried out by using a single core of a Dell work station having four Intel Xeon 5460 dual core and 28 GB of RAM operating within Windows Server Edition 2003.

6.1. Test Bed

Four different parameter sets are employed during the experimental analysis. Most of the parameters are common in all sets. Therefore, we give the values of the common parameters before underlining different points. First of all, the unit data production rate of the sensors, which is h, for the UHM, LHM and MSM is taken as 4096 bits/hour. Two types of sensors are employed in MSLRP, MCSLRP, LSRP and MLSRP. The data generation rates of the sensors are taken identical and equal to 4096 bits/hour independent of the type (i.e., $h_r = 4096$ bits/hour for all $r \in R$).

We assume that both type of sensors employed in MSLRP, MCSLRP, LSRP and MLSRP posses identical energy usage characteristics and they are able to match the transmission power according to the transmission distance. Therefore, the transmission energy used by the sensor is a function of the transmission distance. A well known energy model of that type is due to Heinzelman *et al.* (2000). They divide the transmission energy into two parts one of which is directly proportional with the square of the transmission distance (if the path loss index is 2) while the latter part is considered as constant. In other words, the transmission energy used by a sensor located at point i for a transmission to a sensor located at point j is $c^t_{ij} = \kappa_1 + \kappa_2 d^2_{ij}$ in which d_{ij} is the distance between the points i and j, κ_1 and κ_2 are two coefficients respectively selected as 50 μJ per bit and 100 nJ per bit $\times$ meter square. The energy used by the receiver sensor is constant and independent from the transmission distance. We set $c^r = 50$ μJ per bit in our experiments. On the other hand, since the shortest path data propagation protocol is employed for UHM, LHM and MSM, the energy model requires the calculation of c_{ki} as follows. Suppose the shortest path from sensor k to sink visit point i passes through sensor r just after sensor k. This implies that the data will be transferred from sensor k to sensor r at the first hand while the sink is at visit point i if sensor k is served from visit point i. Let the distance between sensors k and r to be represented by d_{kr}. Then, $c^t_{ki} = \kappa_1 + \kappa_2 d^2_{kr}$. Values of these parameters are employed as explained above and advised by Heinzelman *et al.* (2000) and Türkoğulları *et al.* (2009).

Parameters that share common values and interpretations in all models are explained in detail above. Some parameters also have the same interpretation but with different values in all or most of the models. These parameters are summarized in Table 6.1. The first parameter set is used for the models with nonzero sink traveling times, which are UHM, LHM and MSM. As can be understood from the table, initial sensor battery energies, E_k's, are selected as 86400 Joules for each $k \in \mathcal{N}$ as done in Türkoğulları *et al.* (2010a). Moreover, we assume that the transmission range of the sensors is 100 meters.

The second parameter set is used for empirical analysis of the effect of integration of major WSN design issues. In other words, comparison and analysis of the objective function values of the models MSLRP, MCSLRP, LSRP and MLSRP are made by referring to the parameter values given by set 2. On the other hand, accuracy and efficiency of PIH and SAH algorithms developed for the solution of MLSRP and HH algorithm developed for the solution of LSRP are accomplished based on the problem instances with parameter values given respectively by sets 3 and 4. For sets 2, 3 and 4 we employ two types of sensors. Their battery energies are respectively chosen as $E_1 = 43200$ Joules, $E_2 = 86400$ Joules in set 2, as $E_1 = 10000$ Joules, $E_2 = 20000$ Joules in set 3, and finally as $E_1 = 5000$ Joules, $E_2 = 10000$ Joules in set 4. Last energy related parameter of the sensor relates to the amount of energy used for a unit time by an active sensor, which is the energy used for sensing and processing the data. It is set as $c^s = 62$ J/h for all sensor types in parameter set 2 and as $c^s = 204.8$ mJ/h for all sensor types in parameter sets 3 and 4. Note that this energy related parameter is not required in UHM, LHM and MSM models. This is due to the fact that these models do not incorporate the activity scheduling of the sensors. Therefore, each of the sensor must be awake throughout the network lifetime and must use the same amount of sensing energy which can be considered as constant and reduced from the initial battery energies before starting the analysis. Hence, our analysis remains valid under this consideration.

Sensing ranges of type 1 and type 2 sensors are employed as 66.67 and 100 meters respectively, in set 2 while they are respectively taken as 15 and 22 meters in sets 3 and

4. Communication ranges of the sensors of type 1 and type 2 are respectively given as 100 and 150 meters for set 2, and 50 and 80 meters for sets 3 and 4.

Table 6.1. Common parameters with different values.

Parameter Sets	E_1	E_2	c^s	r_1^s	r_2^s	r_1^c	r_2^c
1	86400 J	-	-	-	-	100 m	-
2	43200 J	86400 J	62 J/h	66.67 m	100 m	100 m	150 m
3	10000 J	20000 J	204.8 mJ/h	15 m	22 m	50 m	80 m
4	5000 J	10000 J	204.8 mJ/h	15 m	22 m	50 m	80 m

Finally, the deployment cost over the candidate locations $i \in \mathcal{S}$ is generated according to a uniform distribution. Namely, the deployment cost of type 1 sensor is uniformly distributed between 1 and 10, e.g. $f_{i1} \sim U(1, 10)$ whereas deploying a type 2 sensor is more expensive and the difference between type 1 and type 2 deployment costs is also uniformly distributed between 0 and 5, e.g. $f_{i2} = f_{i1} + U(0, 5)$. Moreover, the total budget is taken as a weighted sum of type 1 and type 2 deployment costs. These weights have distinct values in different numerical experiments. For instance, while empirically evaluating the integration of major design issues by comparing best lifetimes found for models MSLRP, MCSLRP, LSRP and MLSRP, we assume three budget levels for sensor deployment: lower budget level $B_l = \sum_{i \in \mathcal{S}} (0.25 f_{i1} + 0.25 f_{i2})$, moderate budget level $B_m = \sum_{i \in \mathcal{S}} (0.125 f_{i1} + 0.375 f_{i2})$ and high budget level $B_h = \sum_{i \in \mathcal{S}} (0.50 f_{i1} + 0.50 f_{i2})$. Therefore, we perform an analysis for the sensitivity of the results to the deployment budget as well. Besides, we employ a single budget level while evaluating the accuracy and efficiency performances of PIH and SAH algorithms developed for the solution of MLSRP and it is set to $B = 0.5 \sum_{i \in \mathcal{S}} (f_{i1} + f_{i2})$. Hence, the weights of type 1 and type 2 deployment costs are assumed to have equal impacts on the final total deployment budget. On the contrary, the weights of type 1 and type 2 deployment costs are respectively taken as 0.125 and 0.375 while assessing the accuracy and efficiency of the HH algorithm devised for the solution of LSRP.

On the other hand, there are some parameters which are application specific. For instance, sink velocities are relevant only for the UHM, LHM and MSM models which consider nonzero sink travel times. Moreover, as mentioned at the beginning of this chapter, the problem instances, on which we carry all of the numerical experiments, are generated in two different topologies. In the first one, (candidate) sensor locations are uniformly distributed over a two dimensional plane so that the resulting network is a connected one. In the second one, (candidate) sensor locations are assumed to possess a grid structure. We explain the details of these topologies below and also specify the numerical experiments employing them. We also give explanations for the application specific parameters in that context as well.

6.1.1. Random deployment

In this part, we explain the generation of the sensor field in which the (candidate) sensor locations are determined randomly in two dimensional plane. Note that heuristics developed for the solutions of the models with nonzero sink travel times and the empirical evaluation of integrating major design issues are made based on the problem instances generated according to the random deployment.

We generate 5 different instances for each of 20 different problem sets, 10 of which have a hop limit of 2 and the rest have a hop limit of 3. The problem instances include 20, 30, 40, 50, 60, 100, 125, 150, 175 and 200 sensors that are located randomly so that the sensor network is connected. The number of candidate sink visit points is equal to the one forth of the number of sensor nodes and they are randomly deployed in the network area. Later, additional sink visit points are determined randomly within the transmission range of at least one uncovered sensor until each of the sensors are covered by at least one sink visit point. Therefore, we ensure that no sensor is left unserved. We also ensure that a satisfactory number of alternative sink visit points is generated. All problem instances are generated considering a hop limit; but they can be used for testing the other models which are UHM, MSM, MSLRP, MCSLRP, LSRP and MLSRP. This is because the hop limit only forces to determine a sufficient number of sink visit points so that all sensors are able to be served by at least one point. If

the restriction on the hop limit is removed, nothing prevents using the predetermined sink visit points for the unlimited hop case as well.

For assessing the accuracy and efficiency of TIH designed for the solution of UHM and LHM and for analyzing the results of MSM, we refer to the problem instances generated based on this topology. Other parameters employed are as follows. First of all, the value of parameter R which determines the number of tours during the network lifetime is set to 10 while testing the accuracy of the proposed heuristic approach. On the other hand, during the comparison with the distance-independent models found in the literature, we test the sensitivity of our models to parameter R by setting it to 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 75 and 100. The sink velocity v is 5 km/hour for all problem instances for LHM and UHM. Similarly, all the sink velocities are equal to 5 km/hour, i.e., $v_s = 5$ km/hour for all $s \in \mathcal{M}$ in MSM. If the WSN is deployed in the wild habitat conditions, a 5 km/hour velocity for the sink is a rational choice. We let $tmin_i$ and $tmax_i$ be equal to respectively 1 hour and 8760 hours (1 year) for all $i \in \mathcal{S}$. Finally, we limit the sink tour length by D, which is set to 10000 meters for all problem instances.

On the other hand, we again refer to random deployment topology while assessing the best results found by Gurobi for the models MSLRP, MCSLRP, LSRP and MLSRP. For these instances, we assume that the number of sinks to be located in each period is equal to 3, 5 and 7. Hence, we can have an idea on the sensitivity of the effect of the integration approach to the number of sinks. The sensor field is also represented by a set of discrete points each of which may have a different coverage requirement. This is because the level of significance among the points may be different or there may be critical points requiring more careful surveillance. We assume that the set of points to be covered coincides with the set of sink visit points and we randomly assign the coverage requirements at each point to 1 or 2. One can easily generate the set of coverage points in such a way that they do not coincide with the set of sink visit points but this does not restrict or affect our analysis at all.

6.1.2. Grid Structure

Second sensor field topology assume that the candidate sensor and sink locations expose a grid structure. Note that the success of PIH and SAH heuristics for MLSRP and HH algorithm for LSRP are tested on the problem instances produced from this sensor field topology.

We generate 14 different problem sets each of which has 5 instances. They have respectively 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200 and 300 candidate sensor locations. A grid structure is employed in which candidate sensor locations are determined as the intersection points. Number of sensors in each dimension of the grid is determined as the closest integers whose multiplication gives the number of candidate sensor locations. For instance, for 90 and 175 candidate sensor locations, grid structures are respectively generated as 9×10 and 7×25. We assume that the difference between the neighboring grid points are set as 15 meters. This grid structure created for candidate sensor locations is called as sensor grid in the sequel.

On the other hand, we generate a grid structure for candidate sink locations as well, which we call sink grid. The number of candidate sink locations are taken as the half of the number of candidate sensor locations if the number of candidate sensor locations is divisible by 2. If it is not divisible by 2 than the number of sink locations are found by halving the five more of number of candidate sensor locations. For instance, there are 50 and 65 candidate sink locations for the networks having respectively 100 and 125 candidate number of sensors as $50 = 100/2$ and $65 = (125 + 5)/2$. The number of sink visit points in horizontal and vertical dimensions of the sink grid is determined in a similar fashion that is performed for sensor grid. In other words, they are selected as the closest integers which produce the number of candidate sink locations when multiplied by each other and the smaller one of the two integers is selected as the number of sink visit points through the horizontal dimension. For instance, for networks with 125 sensor locations, there are 65 candidate sink locations and the sink grid is taken as 5×13. Sensor and sink grids are assumed to posses a nested structure in the sense that they both lay on the same sensor field. Four

outmost sink grid points are determined as the middle points of the four outmost squares of the sensor grid. The middle point of the outmost southwest square of the sensor grid is $(7.5, 7.5)$ for all test problems. Therefore, the outmost southwest sink point is $(7.5, 7.5)$ for each sink grid. On the other hand, all other three sink grid points may differ among the networks depending on the location of the middle points of the outmost squares of the sensor grid. For instance, sensor grid has 5×25 dimensions for 125 number of candidate sensor locations, then the middle point of the outmost southeast square of the sensor grid is $(52.5, 7.5)$, which is selected as the outmost southeast sink grid point. Similarly, outmost northwest and outmost northeast corner locations of the sink grid is the middle points of the northwest and northeast squares of the sensor grid which is $(7.5, 352.5)$ and $(52.5, 352.5)$, respectively. Horizontal and vertical distances between neighboring sink grid points can be respectively calculated as $(52.5 - 7.5)/4 = 11.25$ and $(352.5 - 7.5)/24 = 14.375$ meters. In general, one can calculate the horizontal and vertical distance between neighboring sink locations respectively as $(15 * (n_1 - 2))/(m_1 - 1)$ and $(15 * (n_2 - 2))/(m_2 - 1)$ given that the sensor and sink grids have $n_1 \times n_2$ and $m_1 \times m_2$ dimensions, respectively. The number of actual sinks that are located among the sink grid points at each period is taken as three, five and seven, i.e., $P = 3$, 5 and 7.

We assume that the coverage requirements of the network is characterized and represented by some set of discrete points which are called as coverage points. As some parts of the field can be more critical and require more careful surveillance, we conjecture a differentiated coverage setting as it is done in the random deployment topology. In order to reflect the differentiated surveillance requirements of the field, varying level of demand coverages are assigned to the coverage points. These coverage points are taken identical to the sensor grid points this time and the coverage requirements of these points are assigned randomly again as 1 or 2 in our experimental setting. One may take a completely different set of coverage points and differentiate the set of coverage points from the set of sensor points but this does not affect our analysis at all.

6.2. Results for Nonzero Traveling Times

In this section, we provide results related to the models with nonzero sink traveling times. First of all, we verify the mathematical optimization models. Then, we demonstrate the efficiency of the proposed heuristics and finally give observations about the average packet delay performances.

6.2.1. Verification of the Proposed Mathematical Optimization Models

For the purpose of model verification we convert UHM into a distance independent model by setting all d_{ij} values to zero as is assumed in the literature. The resulting model deviates from the one of Basagni *et al.* (2008) in two points. First of all, our model includes a constraint on the tour length. Second, the model given in Basagni *et al.* (2008) has a parameter responsible for the cost involved in the setting and release of the tours. Therefore, if we manage to verify that UHM is able to yield longer network lifetimes in comparison with the model obtained by setting all distance values to zero on the generated problem instances, then we show that the consideration of the traveling times results in better models. From now on, "conventional model" represents the model obtained from UHM by setting all d_{ij} values to zero.

In a solution of the conventional model, the sink visit points with positive sojourn times are provided along with their sequence in the tour. This gives a chance for evaluating the quality of the obtained tour by setting the values of the x_{ij} variables in UHM and solving the remaining RLP optimally. In other words, the quality of a tour is evaluated by the objective value of the RLP corresponding to the tour. UHM and the conventional model are solved using Cplex 11.0 (2007) and the best feasible solutions found within three hours are reported. Finally, the percent deviation of the network lifetime given by the best solution of the conventional model from the lifetime of the best solution provided by UHM is computed for all instances. In mathematical words, let the lifetime of the best solution of the conventional model and that of the UHM be given as z_{CM} and z_{UHM}, respectively. Then, the relative percent deviation between the qualities of the best solutions of the conventional model and UHM is given by

$100 \times (z_{UHM} - z_{CM})/z_{UHM}$. Moreover, if the tour provided by the conventional model turns out to be infeasible for UHM, then the percent deviation is assumed to be 100%.

Since there are five instances for each problem set, the average of these five instances are calculated. We consider 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 75 and 100 values sequentially for R, and we choose 20, 30, 40, 50, 60, 100, 125, 150, 175 and 200 as the number of sensors. Since, the model is unlimited in terms of hops, we use only the problem instances generated by considering 2-hops. Hence, the problem instances with 3-hops are not taken into consideration for these numerical comparisons.

As the summary of the results, we can say that the quality of the tour obtained from the conventional model depends on the network size and the number of sink tours. This result is expected because as the network size and the number of sink tours during the network lifetime increase, the impact of the sink travel time on the final tour becomes more prominent. Figure 6.1 demonstrates the average percent deviations in the lifetime values of the best tours computed by the conventional model and UHM. For instance, when there are 100 sensors, one can observe 55.89% and 66.32% deviations for 50 and 75 sink tours, respectively.

As can be noticed, Figure 6.1 verifies our insight about the accuracy of the tours of the conventional model in the literature. With the exception of the instance with 40 sensors and 100 sink tours, the results of the conventional model are as good as the results of UHM when the number of sensors is less than or equal to 60. However, as the number of the sensors increases to more realistic network sizes, the quality of the conventional tours decreases rapidly. Namely, the percent deviation from the optimal lifetime value generally increases with the number of sensors. One can observe the same kind of dependency between the relative percent deviation and the number of sink tours as well. In other words, as the number of sink tours increases, the percent deviation from the optimal value increases for most of the problem instances, especially for large ones. Moreover, this relationship is monotonic implying that the quality of the conventional tour cannot become better with an increase in the number of sink tours.

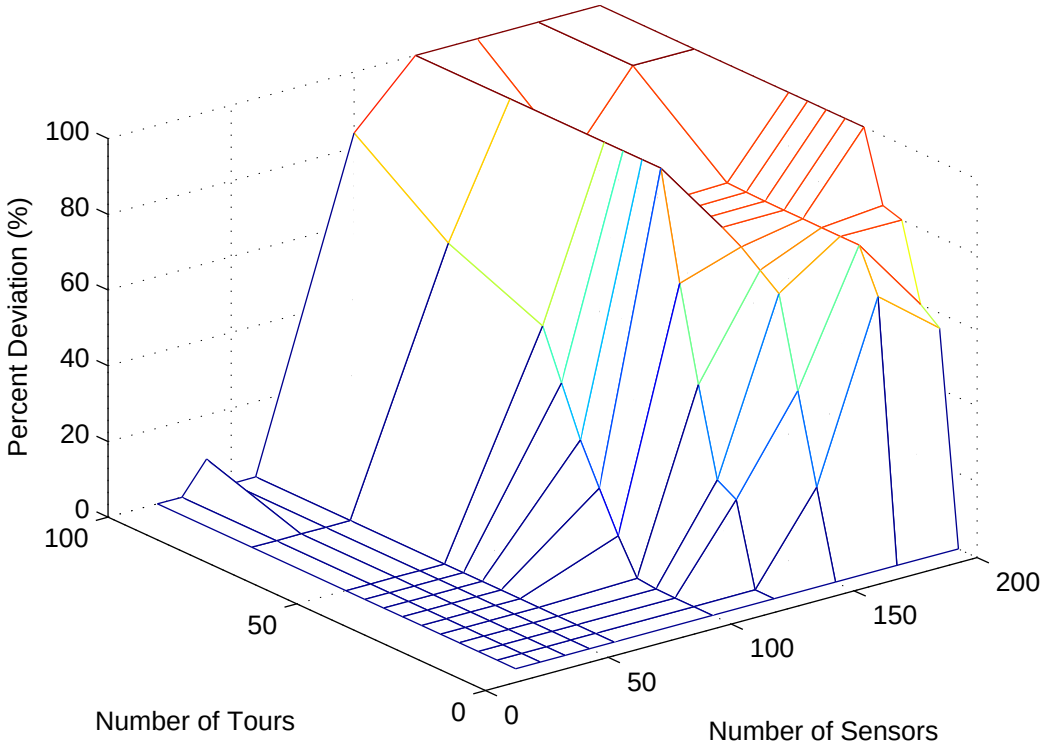

Figure 6.1. The effect of nonzero travel times on the network lifetime.

Since we observe that the quality of the results computed by the conventional model depends on the network size and the number of sink tours, one can increase the network lifetime by taking the sink travel times into consideration especially for large networks and number of tours.

On the other hand, we convert MSM into a distance independent model by setting all d_{ij} values to zero as we do for UHM. The resulting model is very similar to the models of Luo and Hubaux (2010) with minor differences. In order to justify the consideration of nonzero sink travel times, MSM should yield longer network lifetimes in comparison with the model obtained by setting all distance values to zero on the generated problem instances. We repeat the same procedure on MSM that is applied for UHM. This time the conventional model represents the model obtained by setting d_{ij} values in the MSM formulation. MSM and the conventional model are solved using Cplex 11.0 (2007) and the best feasible solutions found within three hours are reported. Quality of the solution obtained from the conventional model can be evaluated by fixing the values

corresponding to the sink location variables $z_{\ell ts}$ for $\ell \in \mathcal{N}, t \in \mathcal{T}, s \in \mathcal{M}_s$ in the MSM formulation and solving the remaining linear model to optimality. Suppose the resulting objective function value is represented by z_{CM} and the objective function value of the best feasible solution found for MSM is given as z_{MSM}. These objective values can be compared in order to reflect the quality of the conventional model solution. The relative percent deviation between the qualities of the best solutions of the conventional model and MSM is again given by $100 \times (z_{MSM} - z_{CM})/z_{MSM}$. We run MSM and the conventional model over the same problem instances mentioned above and report the percent deviations between the qualities of the MSM and conventional model solutions. We use three sinks, and take 5 km/hour for all sinks and observed that all percent deviations are less than 1% for even the largest network. Therefore, we conclude that the sink travel times can be neglected under the presence of multiple mobile sinks.

6.2.2. Performance of the heuristic

In order to evaluate the performance of TIH, we compare the network lifetime found by the heuristic with the one obtained by Cplex within three hours. The plot of the percent deviations for the LHM is illustrated in Figure 6.2. A value of zero for the percent deviation means that the tour computed by TIH is as good as the tour computed by Cplex within three hours. On the other hand, a negative percent deviation implies that the tour of TIH is better than the tour given by Cplex. We keep the number of sink tours R at 10, and consider the problem instances with 20, 30, 40, 50 and 60 sensors. Since Cplex is not able to find a feasible solution for any of the instances with more than 60 sensors, we do not include them in the figure. We remark that in computing the percent deviations we use the average of five instances for each problem set. The CPU times required by the problem instances are plotted in Figure 6.3 where Cplex-2hop, Cplex-3hop, TIH-2hop and TIH-3hop correspond to the CPU times obtained by Cplex and TIH for 2-hop and 3-hop instances, respectively.

A quick look at Figure 6.2 reveals that the performance of the heuristic is quite good. First of all, the maximum deviation is less than 8% and 1% for 2-hop and 3-hop instances, respectively (the largest deviation being 7.62% for the 2-hop instance with

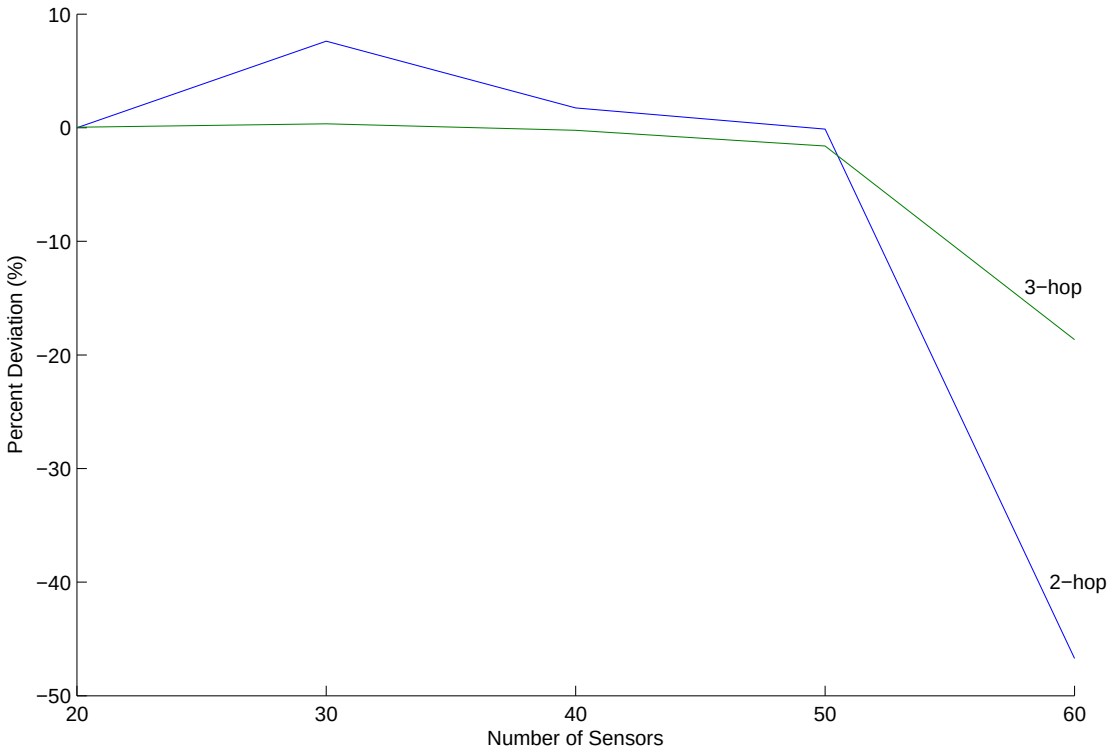

Figure 6.2. Performance of TIH: accuracy.

30 sensors, and 0.34% for the 3-hop instance with 30 sensors). If the 2-hop and 30-sensor case is excluded, the largest deviation decreases to 1.74% for 2-hop instances, and all other deviations are below 1%. In one of the problem instances of the 2-hop and 30-sensor case, the heuristic is trapped on a local optimum which deviates 34.14% from the best result found by Cplex. This high deviation increases the average of five instances to 7.62%. As a matter of fact, for only one out of 25 instances a percent deviation value higher than 5% is observed. Moreover, one can also observe that the results of the heuristic become more accurate as the number of sensors increases. For instance, for the 60-sensor case, the quality of the tour obtained by TIH is even better than the best tour found by Cplex within 3 hours for both 2-hop and 3-hop instances. If the number of sensors becomes larger, then Cplex cannot find a feasible tour within this time limit, which is an indication of the higher accuracy of TIH with the increasing network size. As a consequence, we can say that it is a practical heuristic and is capable of producing better solutions than commercial solvers for WSNs with realistic sizes. On the other hand, Cplex is able to find an optimal solution for all problem instances

within three hours for UHM. Although TIH is still successful for the solution of UHM, it is not beneficial since a commercial solver accomplishes the same task. Therefore, we do not report any results computed with TIH for UHM in this paper. However, a performance analysis of TIH over much larger networks such as the ones with more than 200 sensors can be carried out to emphasize its efficiency.

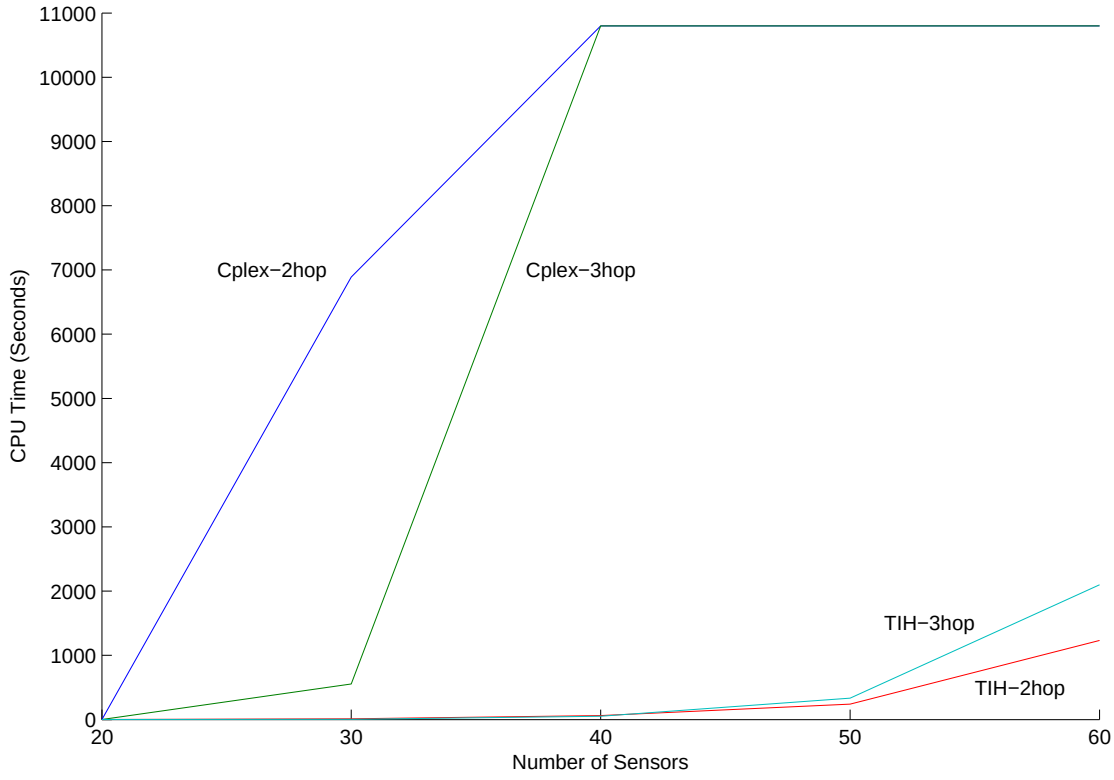

Figure 6.3. Performance of TIH: efficiency.

TIH does not only produce high network lifetime values but also demands less computational time as can be observed in Figure 6.3. For instance, it is seen in this figure that the CPU time of Cplex and TIH for 3-hop and 30-sensor instance are 553 and 3 seconds, respectively. For most of the instances, the results of TIH are obtained within a few seconds. For instances with 50 sensors, the CPU times are in minutes and for instances with 60 sensors results are found within almost half an hour for the 3-hop case and 15 minutes for the 2-hop case. On the other hand, Cplex is stopped by the 3 hour running time limitation during the solution of the most of the instances and the best tour it computes is not significantly better. Thus, we can say that TIH

is not only accurate but also efficient.

6.2.3. On The Average Packet Delay

In LHM, each sensor is served by the sink in each tour as dictated by constraint (3.16). After a feasible tour and the corresponding sojourn times of the sink are obtained using TIH, it is possible to determine the total sleep mode or inactive time of each sensor in each tour. For instance, if a sensor is served from all the sink visit points, then that sensor's inactive time during a tour is equal to the duration of the tour, which is actually a lower limit. On the other hand, if a sensor is not served by at least one sink visit point, the sensor is also inactive during the sojourn time of the sink at that visit point. Without loss of generality, we assume that the total sojourn time at a single visit point is distributed evenly among the tours. In other words, the sojourn times at a visit point are equally divided among the tours. This assumption enables us to estimate the average inactive time of each sensor per tour, which is equal to the sum of the travel times of the sink in a single tour and the ratio of the sum of the sojourn times spent by the sink at the visit points from which the sensor is not served to the number of tours. For example, suppose that the number of sink tours is 10 and a sensor, say sensor k, is not served by sink visit points 1, 3, and 5, and the sink travel time for a single tour is 0.5 hour. Then the inactive time of sensor k for each tour can be found as $0.5 + (t_1 + t_3 + t_5)/10$. By employing this approach, we can determine the average inactive time of each sensor in each tour corresponding to the solutions of TIH for all problem instances.

Furthermore, the average inactive times of the sensors can be converted into the average packet delays for a given data generation period of the sensor. For example, suppose that a sensor's inactive time in a tour is 2.38 hours for a 1-hour data generation period. We assume that the length of the inactive time uniformly divides the data generation period such that on the average the sensor has sent its previous packet just half an hour before the inactive time begins. Hence, the next packet of the sensor is to be generated 0.5 hour later, which has to be delayed at the end of the inactive time. So the delay for that packet is $2.38 - 0.5 = 1.88$ hours. Moreover, a second packet

is also produced 1.5 hours after the inactive time begins and delayed for 0.88 hours. Thus, the average delay per packet for that sensor is simply $(1.88 + 0.88)/2 = 1.38$ hours. Therefore, after calculating the average delay per packet for each sensor, an overall average delay per packet can be computed simply by taking the average of the packet delays over the sensors. Although the values we calculate here are not the exact packet delays, they are close estimates because we take the average through many problem instances. The computational results obtained when the hop limit is set to 3 are summarized in Figure 6.4 and Figure 6.5. We would like to mention that a similar behavior was also observed for different hop limits.

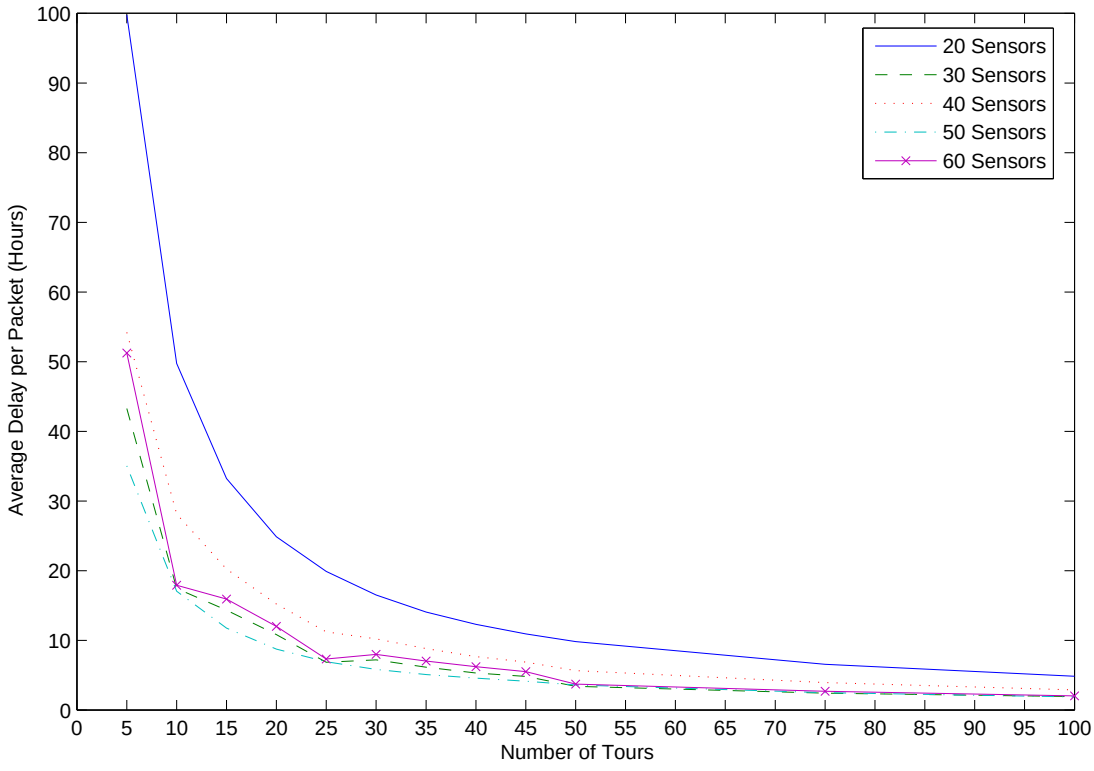

Figure 6.4. Average packet delay: the effect of the number of sink tours.

Each curve in Figure 6.4 depicts the change in the average packet delay with respect to the number of sink tours for a different number of sensors (20–60 sensors in increments of 10). As can be observed in the figures, the average delay decreases as the number of sink tours increases. It is possible to claim that it becomes asymptotically equal to the sink travel time for a single tour, which is an expected outcome. This also shows the benefit of multiple sink tours when the number of hops is limited.

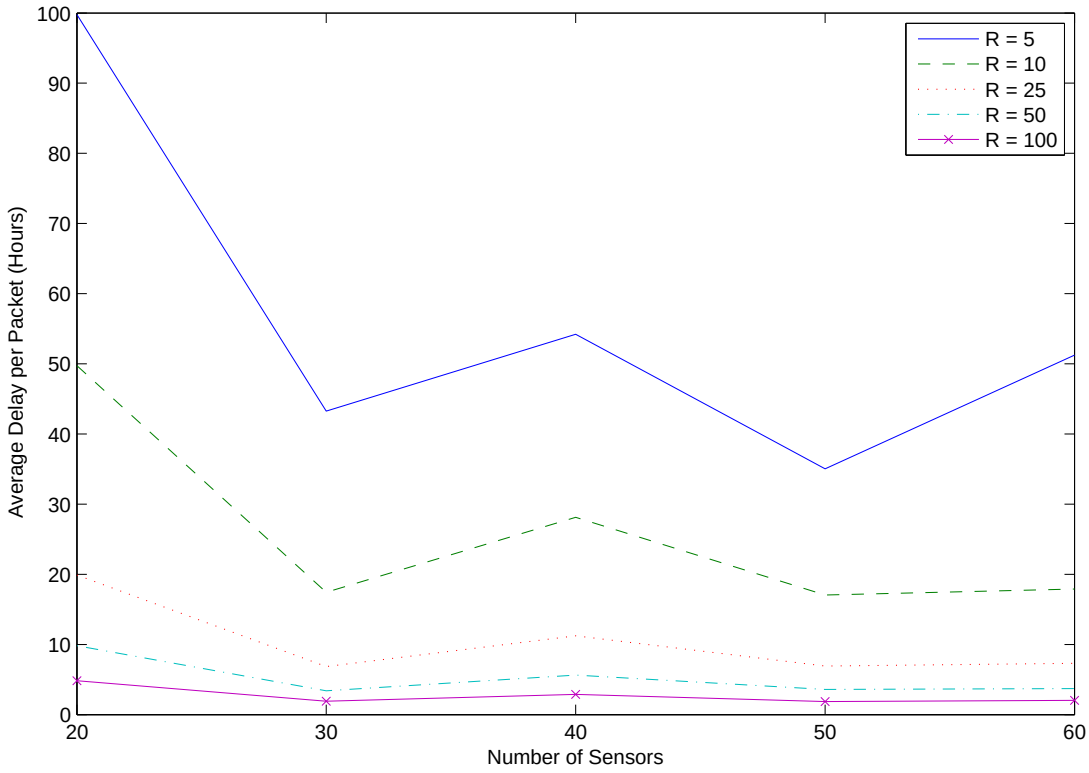

Figure 6.5. Average packet delay: the effect of the number of sensors.

Figure 6.5 illustrates the change in the average packet delay with respect to the number of sensors in the WSN. Each curve in the figure is for a different number of sink tour. They are, starting from the highest to the lowest, for 100, 50, 25, 10 and 5 tours. It is possible to say that, the average delay per packet among WSNs of different sizes have more variation when the number of sink tours are less. Therefore, the inclusion of multiple sink tours contribute to the robustness of the network as well by reducing the variability of the average packet delay.

6.3. Results for Mobile Sinks with Zero Traveling Times

In this section, we first analyze the effect of integration of major WSN design issues. Then we demonstrate the success of PIH and SAH algorithms developed for the solution of MLSRP model by comparing the best network lifetimes found by them in a given computation time with the ones obtained by Gurobi in the same computation time.

6.3.1. Evaluation of the Models

Before beginning to evaluate the effect of integrating major WSN design issues, we provide a summary of the models with zero sink traveling times in Table 6.2. First column identifies the models. Second column includes their brief descriptions, while third column presents the major design issues resolved optimally using the corresponding model. Finally, the last two columns provide information about their complexity, i.e., the order of variables (i.e., binary, continuous) and constraints.

Here, we demonstrate the effect of integration of major design issues and effect of mobility by comparing the results of the summarized models on some randomly generated instances. Initially, the effect of the integration of sensor placement is analyzed by comparing results between MSLRP and MCSLRP. Later on, the simultaneous effect of sensor placement and activity scheduling of the sensors are shown by a comparison between the results of MSLRP and MLSRP. Finally, the effect of the sink mobility is evaluated by interactively analyzing the results of LSRP and MLSRP. For each of the models MSLRP, MCSLRP, LSRP and MSLRP, we run the state-of-the-art MILP solver Gurobi 4.0 (2010) for at most three hours for each instance. Then, we calculate the percent deviation between the objective values of the best solution found for MSLRP and MCSLRP, MSLRP and MLSRP and LSRP and MLSRP. For instance, if the objective function values of the best solutions found by Gurobi for MSLRP and MCSLRP for a problem instance are z_{MSLRP} and z_{MCSLRP}, we report the percent deviation as $100 \times (\frac{z_{MCSLRP} - z_{MSLRP}}{z_{MSLRP}})$. We present the network lifetimes of the MSLRP and percent deviations between the results of MSLRP and MCSLRP, MSLRP and MLSRP, and LSRP and MLSRP in Table 6.3 for 3 sinks and medium budget setting. The first (sixth) column stands for the number of candidate sensor and sink locations while the second (seventh) column includes the lifetime (in hours) corresponding to the best solution found for MSLRP in 3 hours. Moreover, for most of the MSLRP instances, especially for the small sized ones, Gurobi is able to compute the optimal network lifetimes; they are given in bold in the table. The third, fourth and fifth (eighth, ninth and tenth) columns contain respectively the percent deviations between the objective values of the MSLRP and MCSLRP, MSLRP and MLSRP, LSRP and

Table 6.2. A summary of models and involved design decisions.

Model Formulation	**Description**	**Design Decisions**	**Num. of Vars. Binary Continuous**	**Num. of Consts.**
MSLRP (4.3)-(4.9)	Finds optimal mobile sink trajectories and optimal amount of data on each network connection	Sink routes Data routes Flow amounts	$\mathcal{O}(\|N\|\|T\|)$ $\mathcal{O}(\|S\|\|R\|\|T\|(\|S\|\|R\|+\|N\|))$	$\mathcal{O}((\|S\|\|R\|+\|N\|)\|T\|)$
MCSLRP (4.10)-(4.21)	Finds optimal locations of the sensensors, optimal mobile sink trajectories and optimal amount of data on each network connection	Sensor locations Sink routes Data routes Flow amounts	$\mathcal{O}(\|S\|\|R\|+\|N\|\|\|T\|)$ $\mathcal{O}(\|S\|\|R\|\|T\|(\|S\|\|R\|+\|N\|))$	$\mathcal{O}((\|S\|\|R\|+\|N\|)\|T\|+\|K\|)$
LSRP (5.1)-(5.15)	Finds optimal locations and activity schedules of the sensors, optimal stationary sink locations and optimal amount of data on each network connection	Sensor locations Activity schedules Sink locations Data routes Flow amounts	$\mathcal{O}(\|S\|\|R\|\|T\|+\|N\|)$ $\mathcal{O}(\|S\|\|R\|\|T\|(\|S\|\|R\|+\|N\|))$	$\mathcal{O}((\|S\|\|R\|+\|N\|+\|K\|)\|T\|)$
MLSRP (4.22)-(4.36)	Finds optimal locations and activity schedules of the sensors, optimal trajectories of the mobile sinks, and optimal amount of data on each network connection	Sensor locations Activity schedules Sink routes Data routes Flow amounts	$\mathcal{O}(\|S\|\|R\|\|T\|+\|N\|\|T\|)$ $\mathcal{O}(\|S\|\|R\|\|T\|(\|S\|\|R\|+\|N\|))$	$\mathcal{O}((\|S\|\|R\|+\|N\|+\|K\|)\|T\|)$

MLSRP. Therefore, these columns are responsible for the Effect of Sensor Placement, Sensor Placement and Scheduling, and Mobility respectively. To ease the notation we use SPE, SPSE and ME as headings for these columns.

As can be observed from the table, the level of integration can effect the design decisions in terms of the network lifetime. In order to obtain a clearer understanding of this fact, we take the average of the five instances for each problem set and continue to the analysis over the averages. Finally, percent deviations between the results of the models are plotted versus the number of candidate sensor locations for varying number of sinks. For instance, in order to evaluate and quantify the effect of imposing only sensor deployment decision onto the MSLRP, we give the percent deviation plots between MSLRP and MCSLRP in Figure 6.6 for 3, 5 and 7 sinks.

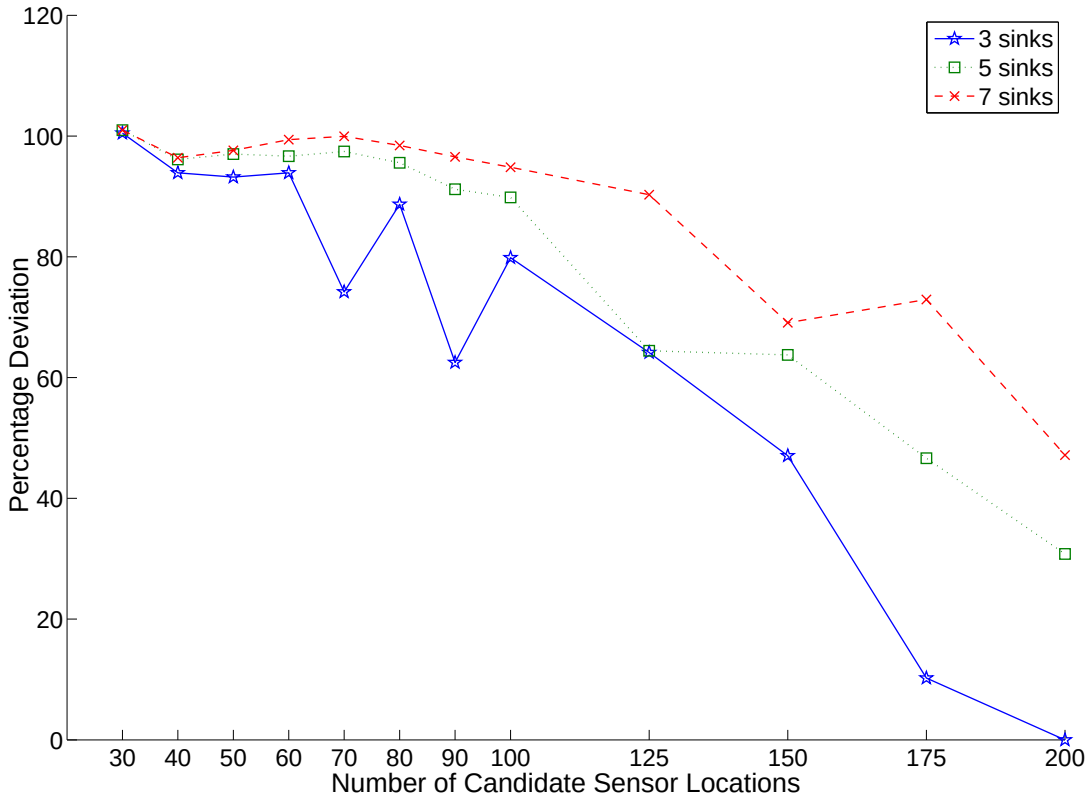

Figure 6.6. The effect of sensor deployment in terms of percent deviations.

We try to expose the favorable effect of the integration of CP with SRP and DRP

Table 6.3. Lifetime values for MSLRP and the effect of integration.

$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{LHM}}$	**SPE**	**SPSE**	**ME**	$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{LHM}}$	**SPE**	**SPSE**	**ME**
(30,8)	**690.99**	95.94	147.87	0.00	(90, 28)	648.46	79.12	109.47	0.44
(30,10)	**680.71**	100.99	302.27	1.03	(90, 28)	638.50	74.02	105.19	12.31
(30,13)	**671.54**	100.65	103.03	0.79	(90, 29)	650.65	75.99	101.24	2.14
(30,8)	**687.19**	98.34	395.70	0.20	(90, 27)	670.84	83.37	148.45	1.87
(30,10)	**656.24**	106.41	212.79	0.00	(90, 27)	**668.25**	0.00	81.92	10.61
(40,13)	**694.48**	92.71	291.31	0.10	(100,29)	665.26	80.20	122.37	0.89
(40,16)	**680.54**	86.39	101.19	5.84	(100,32)	**650.70**	76.20	111.80	4.99
(40,11)	**683.72**	98.24	346.41	0.14	(100,31)	673.50	77.09	105.30	0.00
(40,13)	**671.46**	96.26	103.10	0.00	(100,32)	**643.65**	91.09	112.35	9.28
(40,12)	**686.91**	95.95	249.87	0.01	(100,31)	**651.25**	74.64	110.72	0.98
(50,14)	**679.51**	98.67	103.81	0.00	(125,39)	623.44	65.93	72.74	0.75
(50,16)	**685.68**	88.85	98.67	0.00	(125,37)	628.51	53.97	77.51	0.00
(50,14)	**672.42**	95.46	204.19	0.00	(125,37)	638.22	59.64	91.04	4.17
(50,16)	**682.36**	94.15	151.72	0.00	(125,38)	614.98	62.61	86.02	7.81
(50,15)	**693.90**	88.95	252.57	10.84	(125,38)	677.78	78.55	101.87	3.64
(60,18)	677.15	95.42	274.98	5.34	(150,44)	**623.25**	62.46	90.38	5.35
(60,20)	631.41	100.45	116.17	2.73	(150,44)	639.17	51.41	75.99	4.19
(60,19)	**687.38**	88.14	230.91	10.43	(150,45)	640.55	51.08	61.58	3.10
(60,17)	**692.21**	88.43	99.00	0.00	(150,43)	645.14	13.42	83.87	17.50
(60,19)	**660.26**	97.06	209.52	0.19	(150,43)	**651.49**	56.86	88.10	0.00
(70, 22)	**634.95**	9.38	155.69	9.41	(175,52)	674.28	0.00	53.56	11.70
(70, 20)	**669.49**	85.23	105.99	0.86	(175,49)	624.09	26.50	86.48	1.63
(70, 22)	643.07	90.81	112.34	0.00	(175,53)	623.10	0.00	8.19	11.26
(70, 22)	671.13	90.48	106.86	0.00	(175,54)	**683.59**	0.00	42.68	28.94
(70, 20)	**679.74**	95.06	241.93	1.09	(175,31)	616.77	24.72	66.09	20.39
(80, 24)	666.61	94.19	196.25	2.59	(200,61)	590.52	0.00	12.31	11.77
(80, 24)	679.65	87.63	151.42	5.27	(200,62)	**605.38**	0.00	35.64	30.33
(80, 25)	**681.65**	88.93	101.72	0.00	(200,57)	637.61	0.00	47.45	68.40
(80, 25)	**655.60**	94.98	109.95	0.00	(200,60)	621.27	0.00	10.66	20.53
(80, 26)	**672.22**	77.79	142.23	3.23	(200,57)	**647.27**	0.00	2.44	2.80

in Figure 6.6 by plotting the percentage deviations between lifetime values obtained by MSLRP and MCSLRP within three hours. We can say that such an integration results in a considerable improvement, which decreases with increasing network sizes (i.e., the number of candidate sensor locations). For instance, for 3 mobile sinks, the improvement is around 100% for the networks having 30 candidate sensor points, around 90% for the networks with more than 30 and less than 70 candidate sensor locations and around 75% for networks with 70 candidate sensor locations. Similarly, improvements about 90%, 60% and 80% are observed for networks having 80, 90 and 100 candidate sensor locations. The effect is still important as it is approximately 65% and 50% for large networks such as the networks with 125 and 150 candidate sensor locations, respectively. Finally, a 10% improvement is observed with the networks of 175 candidate sensor locations, and the effect reduces to 0% for the networks with 200 candidate sensor locations.

The effect of integration decreases with increasing network sizes mainly due two two reasons. First, as the candidate number of sensors increases, the problem turns out to be too difficult for Gurobi to solve exactly and improve the results of the previous models. Therefore, for the problems with large networks, Gurobi returns the best feasible solution it finds in three hours which may turn out to be relatively a suboptimal one for large instances. Another reason is that as the size gets larger, the network becomes too crowded so that there exists several critical sensors whose loss leads to a coverage loss. In other words, as the network becomes crowded, the number of critical sensors, whose loss determines the network lifetime, increases. A possible remedy that is worth considering for this phenomenon is the increase in the number of mobile sinks. Increasing the number of sinks may help critical sensors in these cases. Therefore, we also plot the percent deviations between the objective values obtained by solving MSLRP and MCSLRP for 5 and 7 sinks and observe that the positive effect of sensor placement is more influential with larger number of sinks although the reduction in the effect of sensor deployment still persists by increasing the size of the network. Moreover, we also witness that increasing the number of sinks pays back especially for large networks as expected. For instance, the average effect of the sensor deployment is respectively 47.16% and 30.79% for networks with 200 candidate sensor locations with

7 sinks and 5 sinks, while it is 0% for the same network with 3 sinks.

On the other hand, the effect of adding CP and activity scheduling to the integration of SRP and DRP can be seen in Figure 6.7, which includes the plots obtained by solving MSLRP and MLSRP. Notice that the effect of simultaneous consideration of sensor deployment and scheduling decisions is much stronger. For 3 sinks when the network of size is less than 50 candidate sensor locations, the improvement is about 225%. Similarly, the improvement becomes about 160%, 185%, 145% and 140% for networks with 50, 60 70 and 80 candidate sensor locations, respectively. They reduce respectively down to around 110%, 110%, 85% and 80% for networks of 90, 100, 125 and 150 candidate sensor locations. Finally, around 50% and 20% improvements are observed for networks having respectively 175 and 200 candidate sensor locations.

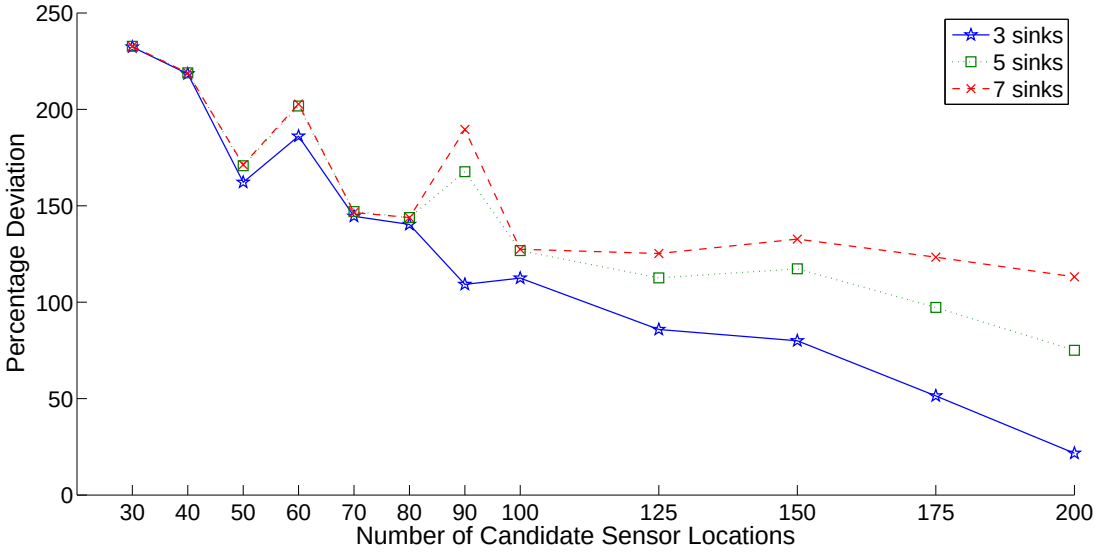

Figure 6.7. The simultaneous effect of sensor placement and scheduling.

Furthermore, the positive effect of simultaneous integration of sensor deployment and scheduling decisions with the MSLRP also decreases with increasing network sizes and we manage to make the reduction curve smoother by increasing the number of sinks. Hence, the comments we make about the response of positive effect of sensor deployment decision integration to the number of sinks is also valid for this case as well. In short, the network designer should consider an increase in the number of sinks for

relatively larger networks in order to get the maximum contribution from an integrated approach.

We plot the percent deviations between the average objective values calculated by solving the formulations LSRP and MLSRP in Figure 6.8 in order to visualize the effect of mobility of the sinks on the network lifetime. As can be seen, for 3 sinks the effect of mobile sink(s) seems to be negligible for small networks. The effect is less than 1% for networks of 20 and 30 candidate locations and around 1% for networks with 40 candidate locations. Moreover, the improvement introduced by the mobility is around 2% for the networks (except the networks with 60% candidate locations is around 2.5%) with 50 - 80 (inclusive) candidate locations. However, the effect becomes larger as the network size increases, which is expected. For instance, the positive effect of mobility becomes around 5.5% on 90 candidate sensor locations, again decreases down to around 3% for 100 and 125 candidate locations. Then, the effect rapidly increases to about 6% for 150, to 15% for 175 candidate sensor points and it finally reaches to around 25% for the networks with 200 candidate sensor locations. Therefore, as the network size gets larger, the mobility of the sinks become more important and causes higher improvements on the network lifetime. This result also coincides with our intuition that an increase in the number of mobile sinks in larger networks may improve the network lifetime as indicated in the previous paragraph. However, an increase in the number of sinks leads to a decrease in the positive effect of mobility, which seems meaningful since the worth of the marginal unit of any material decreases by the increase in the total quantity as a general principle.

Besides, in Figure 6.9, we provide a plot of the lifetimes obtained by Gurobi for the solution of MLSRP against the number of candidate sensor locations for three different budget levels, namely for low, medium and high budgets. We perform this experiment for 3 sinks. As can be seen, the behavior of the lifetimes are similar for all three budget levels. Moreover, we also observe a tendency that the effect of increase in the budget is more influential in large networks. This can be explained by the increase in the number of critical sensors whose loss determine the network lifetime by an increase in the network size. An increase in the budget creates more opportunity to

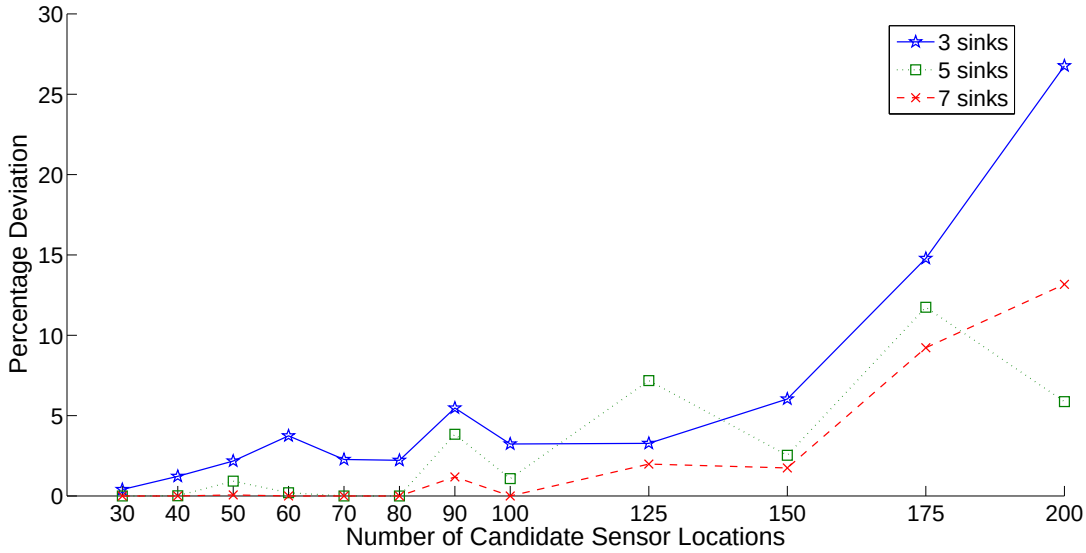

Figure 6.8. The effect of mobility of the sink(s).

help to such critical sensors by either activating sensors of different types at the same point, or activating a sensor near-by the critical sensor. Hence, it is intuitive that the higher budget levels are more effective for larger networks in that sense.

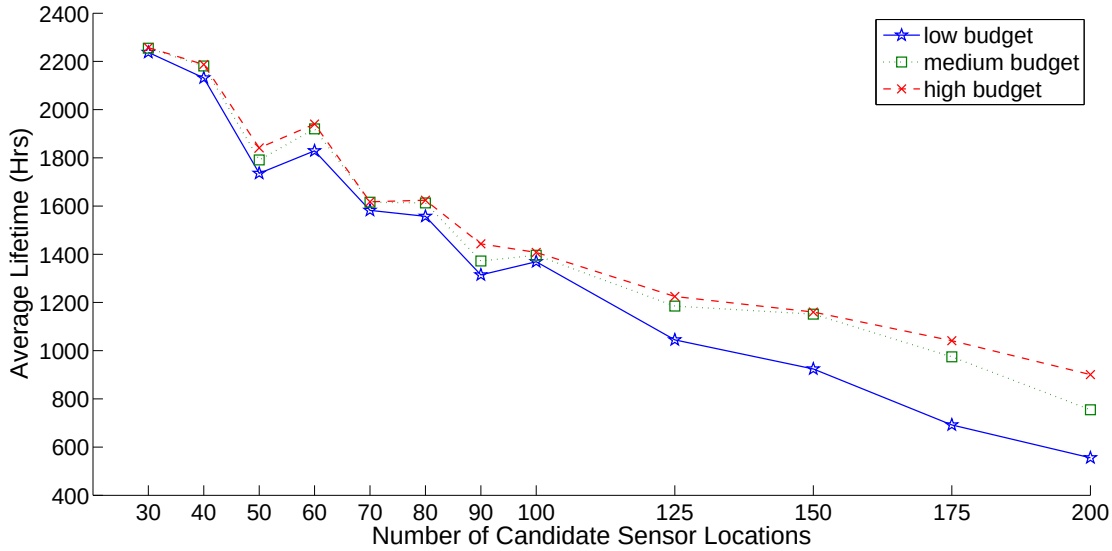

Figure 6.9. The effect of budget: lifetime vs. size.

Although it is possible to claim that the budget level affects the improvement based on the integration given in Figure 6.9, the mechanism behind the direct relation between the budget level and the network lifetime is not clear. Hence, we also provide

Figure 6.10, in which the average lifetime of the networks with 200 candidate sensor locations are plotted against six different budget levels. These levels from the lowest to the highest are $B_1 = \sum_{i \in \mathcal{S}} (0.125 f_{i1} + 0.25 f_{i2})$, $B_2 = \sum_{i \in \mathcal{S}} (0.25 f_{i1} + 0.25 f_{i2})$, $B_3 = \sum_{i \in \mathcal{S}} (0.125 f_{i1} + 0.375 f_{i2})$, $B_4 = \sum_{i \in \mathcal{S}} (0.25 f_{i1} + 0.375 f_{i2})$, $B_5 = \sum_{i \in \mathcal{S}} (0.375 f_{i1} + 0.375 f_{i2})$ and $B_6 = \sum_{i \in \mathcal{S}} (0.50 f_{i1} + 0.50 f_{i2})$. Clearly, an increase in the budget considerably improves the network lifetime. However, its effect seems to be decreasing as the budget level increases. This is due to the fact that the budget constraint becomes redundant as the budget is increased more than a certain threshold for some of the instances implying that a further increase in the budget level has no effect on the lifetime. Hence, the effect of budget rises seems to be lower on the average for high budgets.

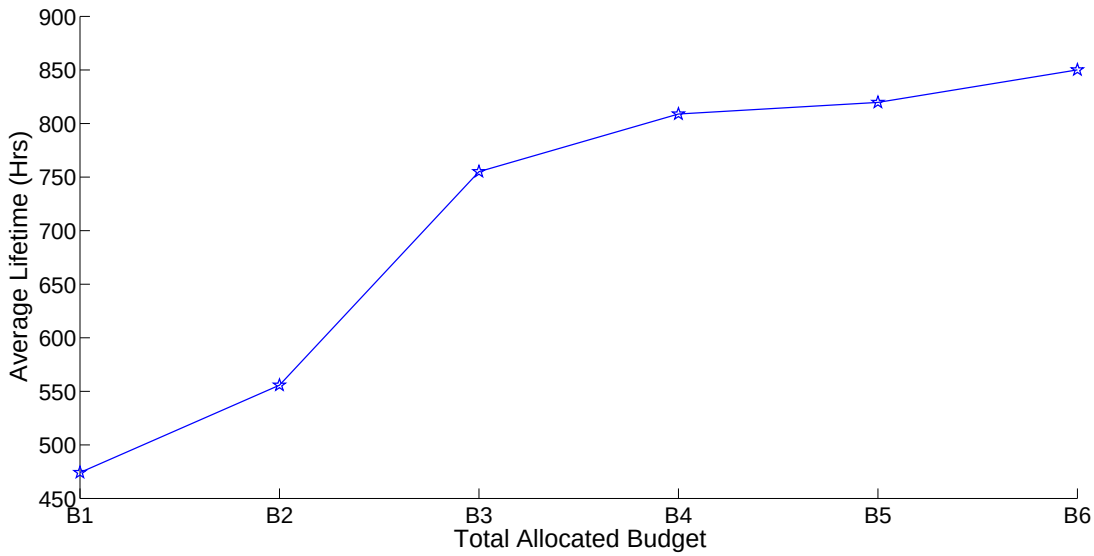

Figure 6.10. The effect of budget: lifetime vs. budget level.

One possible criticism against our approach depends on the fact that whilst computing the percent deviations between model results, we rely on the best possible network lifetime values found by Gurobi in 3 hours which are not necessarily optimal. However, we have an intuitive answer to that question. It is a well known fact in the optimization field that, due to the NP-hard nature of the most of the integer programming problems, performance of the MILP solvers like Gurobi for the smaller models like MSLRP are relatively superior than it is for large models like MCSLRP and MLSRP. This implies that results we report about the positive effect of the decision integration

idea is even conservative and the improvement that can be achieved by integrating sensor placement, sensor scheduling and sink mobility decisions to MSLRP model can even be larger than the values reported in this study. Moreover as can be observed from the network lifetimes reported in Table 6.3 for 3 sink MSLRP instances, Gurobi is able to find an optimal solution for most of the instances especially for the small sized ones meaning that pozitive effect of decision integration is at least as much as the reported values for those instances. On the other hand, for large instances we can expect Gurobi to be more successful for MSLRP than for MLSRP and MCSLRP simply because MLSRP and MCSLRP instances are at least as hard as MSLRP instances. Hence, our results are conservative for large instances as well.

6.3.2. Performance of Period Iteration and Sequential Assignment Heuristics

In this section, we try to assess the performance of the heuristics compared to the state-of-the-art MILP solver Gurobi on the randomly generated problem instances. For each of the test instances, we let Gurobi, PIH and SAH to run at most three hours and report the maximum network lifetime obtained for $P = 3$ sinks in Table 6.4. The first (sixth) column stands for the number of candidate sensor and sink locations while the second, third, fourth and fifth (seventh, eighth, ninth and tenth) columns include the maximum network lifetimes obtained by Gurobi, PIH and SAH, respectively for the solution of MLSRP during 3 hours.

To better assess the success of the heuristics, we compute the average network lifetimes of five instances corresponding to each problem set with a different number of candidate sensor locations, and plot them against the number of candidate sensor locations in Figure 6.11, Figure 6.13 and Figure 6.14 for 3, 5 and 7 sinks, respectively. We also compute the percent deviation between the maximum network lifetimes obtained by Gurobi and the heuristics as follows: $100 \times (\frac{z_{\text{PIH}} - z_{\text{G}}}{z_{\text{PIH}}})$ and $100 \times (\frac{z_{\text{SAH}} - z_{\text{G}}}{z_{\text{SAH}}})$. The average percent deviations for each problem set are respectively plotted in Figure 6.12, Figure 6.15 and Figure 6.16 for 3, 5 and 7 sinks.

Table 6.4. Maximum lifetime values obtained for MLSRP by Gurobi, PIH and SAH.

$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{G}}$	$z_{\mathbf{PIH}}$	$z_{\mathbf{SAH}}$	$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{G}}$	$z_{\mathbf{PIH}}$	$z_{\mathbf{SAH}}$
(20, 10)	97018.00	94081.00	78538.27	(90, 45)	28602.55	31506.69	41532.86
(20, 10)	87790.97	78108.45	78105.78	(90, 45)	25265.33	31135.02	40879.14
(20, 10)	92985.17	88334.74	78009.41	(90, 45)	15241.33	26606.89	37870.41
(20, 10)	98228.52	104145.26	89646.22	(90, 45)	21265.98	28508.82	36386.79
(20, 10)	88446.30	77986.82	68372.78	(90, 45)	20575.32	31192.99	31613.76
(30, 15)	69231.61	75029.88	64738.55	(100, 50)	18684.52	27989.04	36241.61
(30, 15)	71420.74	71084.02	65835.68	(100, 50)	22897.74	25744.00	37530.37
(30, 15)	86438.23	75500.32	75140.46	(100, 50)	27762.38	27266.92	36543.09
(30, 15)	72617.78	75670.49	62614.08	(100, 50)	21788.25	22920.31	28932.71
(30, 15)	96140.15	77217.00	88251.99	(100, 50)	19929.85	24676.80	33872.28
(40, 20)	77198.41	62580.27	74335.68	(125, 65)	15090.45	21415.34	29704.73
(40, 20)	65868.96	58223.84	63137.11	(125, 65)	21702.53	22064.35	31380.45
(40, 20)	56943.71	55167.81	55120.87	(125, 65)	14287.30	23051.30	25529.78
(40, 20)	58931.98	58695.51	64084.63	(125, 65)	12388.17	21945.20	29848.70
(40, 20)	60197.12	58397.50	58596.77	(125, 65)	2829.31	21258.03	30980.44
(50, 25)	50216.42	47919.91	57533.67	(150, 75)	2099.22	17428.48	22396.16
(50, 25)	53289.69	59464.82	59512.29	(150, 75)	11908.02	16443.39	28069.42
(50, 25)	44220.33	47078.78	52225.93	(150, 75)	0.00	20527.92	25765.42
(50, 25)	49595.42	50039.37	49755.15	(150, 75)	11984.73	13474.58	28409.26
(50, 25)	44158.40	43470.75	45684.85	(150, 75)	2142.61	18486.94	28353.19
(60, 30)	30747.31	38356.31	43330.73	(175, 90)	10680.24	11485.92	25691.75
(60, 30)	44036.71	42073.22	47422.20	(175, 90)	0.00	14556.05	21569.37
(60, 30)	37409.13	42594.30	44225.46	(175, 90)	0.00	9367.47	25932.04
(60, 30)	46193.67	42197.39	44466.13	(175, 90)	11034.08	12202.47	24582.01
(60, 30)	40713.51	44764.76	42074.65	(175, 90)	0.00	14672.68	22968.62
(70, 35)	37026.57	36292.45	45123.31	(200, 100)	0.00	10339.99	27875.19
(70, 35)	27553.40	37133.35	41757.57	(200, 100)	0.00	10011.74	21205.11
(70, 35)	27769.43	35018.46	47391.83	(200, 100)	0.00	11331.97	27615.42
(70, 35)	40704.09	37382.15	41740.55	(200, 100)	0.00	9586.48	24046.19
(70, 35)	27611.60	30424.57	42482.00	(200, 100)	0.00	13569.71	21157.06
(80, 40)	35869.73	27526.33	35055.70	(300, 150)	0.00	3773.94	16758.08
(80, 40)	37391.14	35791.49	36915.53	(300, 150)	0.00	6940.45	13957.22
(80, 40)	27818.06	33266.36	39569.12	(300, 150)	0.00	5768.77	15458.48
(80, 40)	27630.02	38166.60	41591.44	(300, 150)	0.00	6901.57	17234.99
(80, 40)	36937.75	34099.83	32584.53	(300, 150)	0.00	6307.61	19585.93

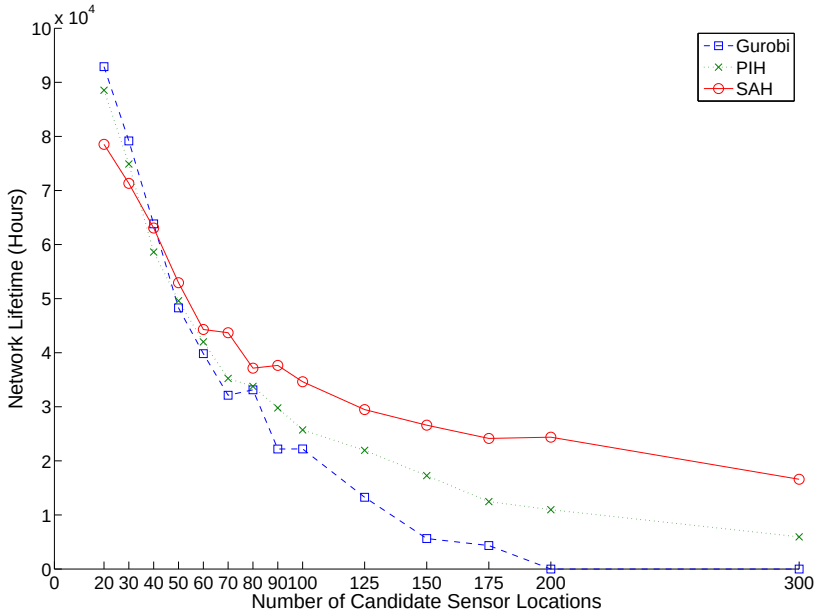

Figure 6.11. Maximum network lifetimes by Gurobi, PIH and SAH for 3 sinks.

As can be seen from Table 6.4, Figure 6.11 and Figure 6.12 which are prepared for 3 sinks, the maximum network lifetimes obtained by Gurobi are longer than those found by PIH and SAH for networks with 20, 30, and 40 candidate sensor locations within the three-hour time limit. However, both heuristics produce longer lifetimes than Gurobi does for most of the instances having 50-80 (inclusive) candidate sensor locations, and for all instances with larger than 80 candidate sensor locations. Moreover, the success of the heuristics becomes more apparent for larger networks such as the ones having more than 100 candidate sensor locations. For example, for the instances with 150 candidate sensor locations, percent deviations reach 63.00 % for PIH and 79.69% for SAH. These percent deviations imply that the average network lifetimes found by PIH and SAH are respectively 2.7 and 4.92 times longer than the average network lifetime found by Gurobi. Percent deviations for instances with 175 candidate sensor locations are very close to the percent deviations observed for instances with 150 candidate sensor locations. As the network size increases to 200 and 300 candidate sensor locations, percent deviations reach 100 % due to the fact that Gurobi is unable to find a feasible solution for any instances of the networks with 200 and 300 candidate sensor locations. On the other hand, PIH is more successful than SAH only for the networks with 20 and 30 candidate sensor locations. PIH and SAH can be considered as equally successful

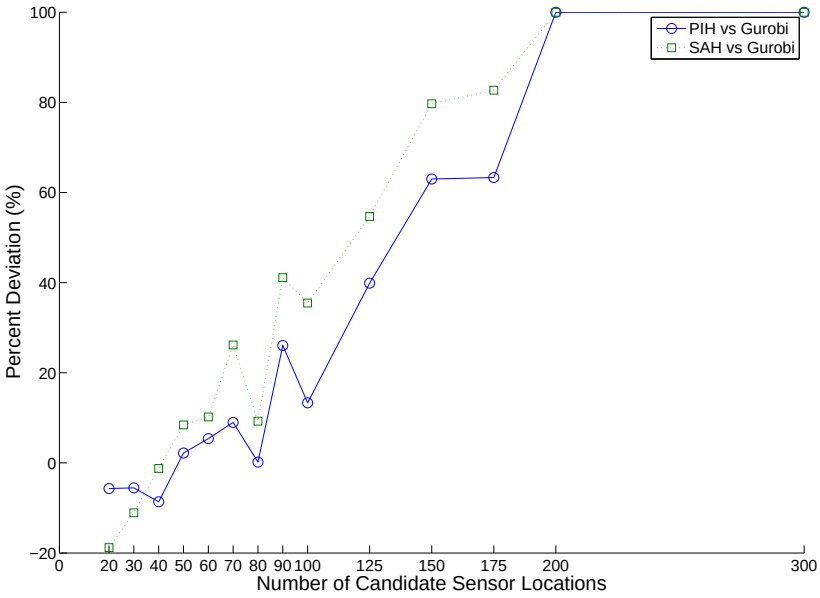

Figure 6.12. PIH and SAH vs. Gurobi: percent deviations for 3 sinks.

for networks having less than 90 candidate sensor locations, but SAH dominates PIH for large networks. As a summary of the results, one can claim that both of PIH and SAH are more successful than Gurobi almost for all instances of MLSRP, except very small ones.

In addition, one might like to see that PIH and SAH are robust in the sense that they produce networks of comparable quality for varying network parameters. Therefore, we also experiment with the same network instances changing only the number of sinks. Average network lifetimes obtained for 5 and 7 sinks are respectively plotted at Figure 6.13 and Figure 6.14 while Figure 6.15 and Figure 6.16 include average percent deviations for each problem set. A quick look at these figures reveals that the relative performances of the heuristics and Gurobi persist for increasing number of sinks. Hence, the conclusions derived from the results for 3 sinks are still valid for 5 and 7 sinks and both heuristics are robust in this sense. On the other hand, it seems that Gurobi is negatively affected from the increase in the number of sinks for especially large networks due to the resulting growth in the complexity of the formulation, which makes PIH and SAH promising tools for the design of large WSNs.

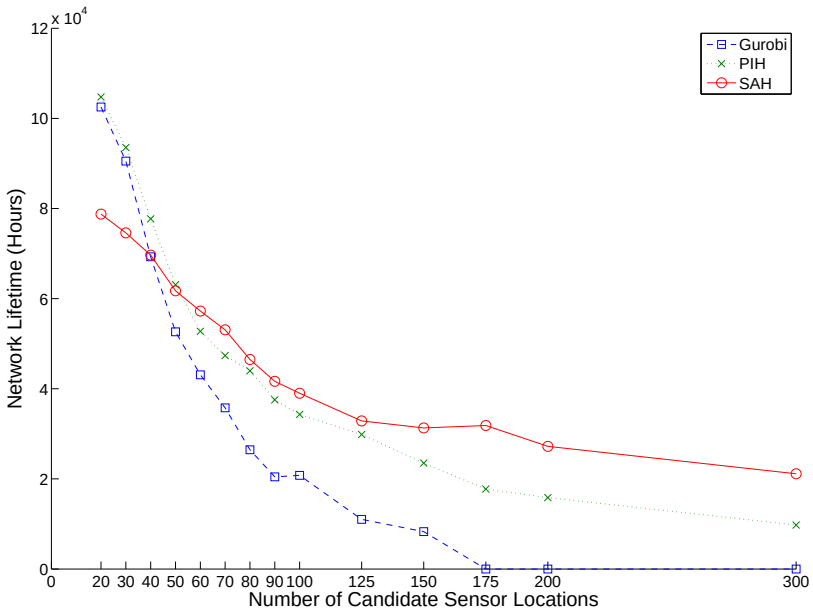

Figure 6.13. Maximum network lifetimes by Gurobi, PIH and SAH for 5 sinks.

One can also observe from Figure 6.11, Figure 6.13 and Figure 6.14 that the network lifetime decreases with the increasing number of candidate sensor locations for all of the three solution procedures (i.e., Gurobi, PIH, SAH). There may be three reasons behind it. First, as the number of candidate sensor locations increases, the problem turns out to be computationally difficult for Gurobi and the heuristics to solve optimally. Therefore, for the problems with large networks, all three solution methods return the best feasible solution they find in three hours which may turn out to be suboptimal for large instances. Another reason is that as the size increases, the network becomes too large so that there exists several critical sensors whose death leads to a serious loss in the coverage. In other words, as network becomes larger, the number of critical sensors, whose loss determines the network lifetime, increases. A possible remedy that is worth considering is to increase the number of mobile sinks. A comparison between Figure 6.11, Figure 6.13 and Figure 6.14 reveals that the graph becomes flatter as the number of sinks increase which is an indication of the positive effect of the number of sinks on network lifetime. That effect seems to be stronger for the networks of moderate size since there is more need for more mobile sinks in moderate sized networks compared to the small networks and at the same time the solution methodologies are able to produce near optimal solutions. It can be expected

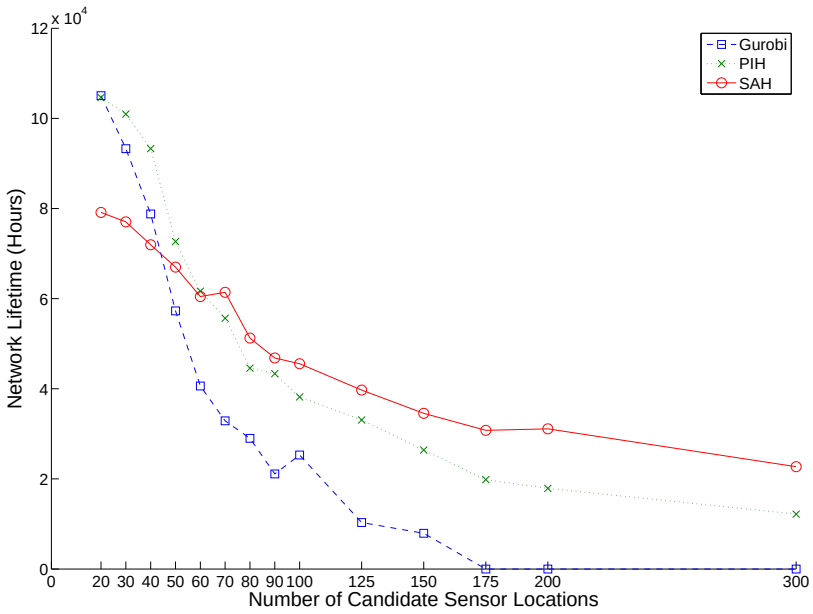

Figure 6.14. Maximum network lifetimes by Gurobi, PIH and SAH for 7 sinks.

that the favorable effect (i.e., longer lifetimes) of increasing the number of sinks becomes even higher for larger networks. Unfortunately, this is not clearly observable in practice since it is not possible to solve large problems optimally in a reasonable running time. A final reason for the decrease in the network lifetime by the increase in the number of candidate sensor locations is that as the networks becomes larger, more sensors send their data via multiple hops which increases the load of the relaying nodes. This may wear out these sensors sooner which can result in lower network lifetimes.

Recall that we allot three hours CPU times for Gurobi, PIH and SAH for each of the problem instances but if Gurobi finds an optimal solution or PIH or/and SAH converges to a solution before three hours they immediately report the best feasible solution they found and start processing for the next instance without waiting until the end of the three hours. The average computation times required by Gurobi, PIH and SAH are plotted in Figure 6.17 against the number of candidate sensor locations for 3 sinks. We observe that Gurobi uses all 3 hours for all of the instances. For example, average computation time used by Gurobi for 300 candidate sensor locations is 61703.28 seconds. We exclude it from Figure 6.17 since it is out of scale. Both of the heuristics, however, require much less computation time than Gurobi. As only a

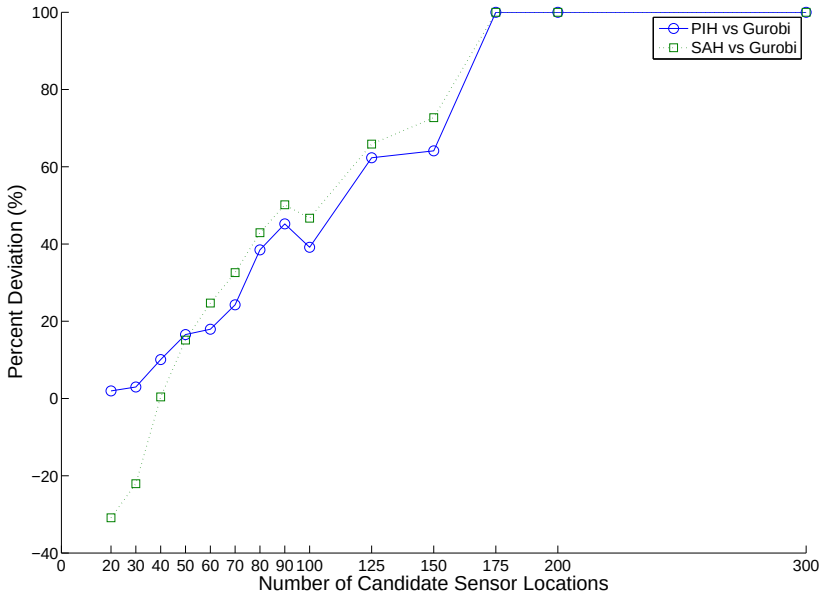

Figure 6.15. PIH and SAH vs. Gurobi: percent deviations for 5 sinks.

small amount of time (we implement it as 100 seconds) is allotted for any round of PIH (for any value of ϕ), Gurobi is unable to improve the solution from the previous restricted model after several iterations and PIH quickly converges. Hence, the amount of total time consumed by PIH is relatively lower. Similarly, we devote 100 seconds for the solution of each subproblem of SAH. Nevertheless, as the convergence of the inner loop of SAH takes several steps for any value of ϕ, the total time required by SAH is larger compared to PIH for the most of the instances. To be more specific, the computation times of PIH and SAH are nearly the same for networks with 20 and 200 candidate sensor locations, but for all other instances SAH requires between 1.71–4.01 times more CPU than PIH does. Therefore, if a network designer seeks for a heuristic producing good networks around 10 minutes he/she can prefer PIH but if he/she bears around 2.5 times more CPU time, then SAH should be used.

We also plot the average computation times used by Gurobi, PIH and SAH for 5 and 7 sinks respectively in Figure 6.18 and Figure 6.19. It can be observed that the performance of PIH and SAH with respect to the required computation times are quite alike to their performances with 3 number of sinks. Besides, Gurobi has even more difficulty in finding good solution in the allotted computation times and consumes

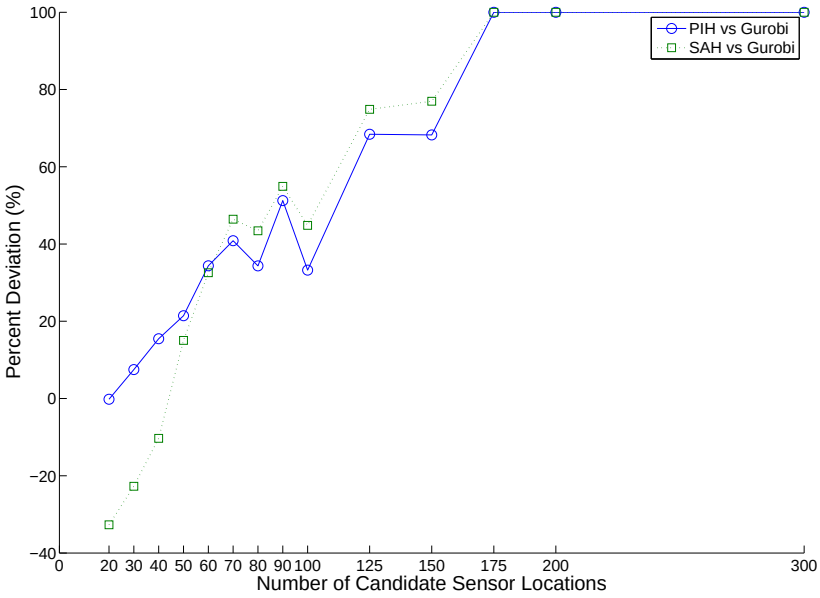

Figure 6.16. PIH and SAH vs. Gurobi: percent deviations for 7 sinks.

even more time for large networks. Hence, the quality of PIH and SAH are even more dominant for large networks.

We finally provide plots of average network lifetimes and average percent deviations for each problem set with data production rates $h_r = 2048$ bits/hour in Figure 6.20 and Figure 6.21, and for $h_r = 8192$ bits/hour in Figure 6.22 and Figure 6.23 while there are 3 sinks. Notice that doubling or halving the data production rate is equivalent to halving or doubling the sensor battery energies. We observe patterns both in average lifetime and in average percent deviations similar to those presented for $h_r = 4048$ bits/hour. We observe more fluctuations in PIH's percent deviation performance compared to those of SAH. Depending on these observations, it is possible to assert that PIH and SAH are robust to changes in the input parameters and even more successful if the input changes increase the size and complexity of the formulation.

On the other hand, if there are small changes in the value of parameters after finding the solution for a default set of parameters, it is possible to use the previous solution as a starting point for all three solution procedures (Gurobi, PIH, and SAH). However, depending on the reported performances, we can claim that repairing infeasi-

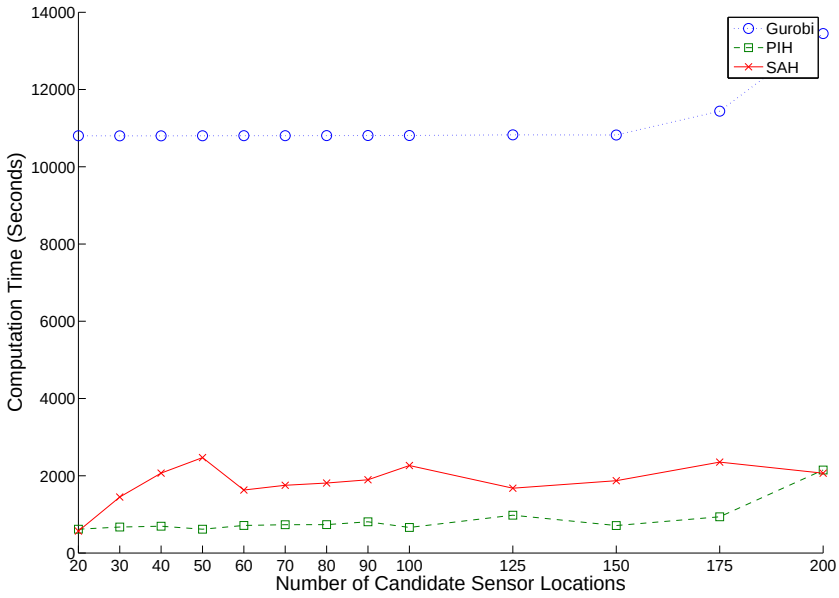

Figure 6.17. Average computation times for 3 sinks.

bilities (if the changes make the previous solution infeasible) or improving the previous solution can be done more efficiently with PIH or SAH than with Gurobi.

6.4. Results for Stationary Sinks

In this section, we assess the performance of the new hybrid heuristic HH by comparing the results with Gurobi, on the described test bed. We let Gurobi and HH run for each instance for at most 3 hours and report the best feasible solution they find. Note that if Gurobi finds an optimal solution within three hours, or HH converges before 3 hours, both method immediately report their solutions and start running for the next instance without waiting until the end of the three-hour-period. Maximum network lifetimes obtained by Gurobi and HH, and percent deviations between them are reported in Table 6.5. The first and fifth columns with header ($|\mathcal{S}|$, $|\mathcal{N}|$) include the number of candidate sensor and sink locations while the second, third, sixth and seventh columns with headers z_G and z_{HH} include respectively the maximum network lifetimes obtained by Gurobi and HH for LSRP in at most 3 hours, respectively. Finally, the values given in the fourth and eighth columns under the heading $\%_D$ are the percent deviation between Gurobi and HH results; they are calculated as $100 \times (\frac{z_{HH}-z_G}{z_{HH}})$.

Table 6.5. Gurobi vs. HH: maximum lifetime and percent deviations.

$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{G}}$	$z_{\mathbf{HH}}$	$\%_{\mathbf{D}}$	$(\lvert\mathcal{S}\rvert, \lvert\mathcal{N}\rvert)$	$z_{\mathbf{G}}$	$z_{\mathbf{HH}}$	$\%_{\mathbf{D}}$
(20, 10)	31673.68	24902.64	-21.38	(90, 45)	7827.15	9700.51	23.93
(20, 10)	26548.47	20412.13	-23.11	(90, 45)	0.00	10089.56	100.00
(20, 10)	31207.74	21945.23	-29.68	(90, 45)	0.00	10327.10	100.00
(20, 10)	28512.77	22810.22	-20.00	(90, 45)	0.00	9844.97	100.00
(20, 10)	28700.08	21945.23	-23.54	(90, 45)	0.00	9626.67	100.00
(30, 15)	22754.74	19676.53	-13.53	(100, 50)	0.00	9152.87	100.00
(30, 15)	20070.06	21945.23	9.34	(100, 50)	0.00	8787.37	100.00
(30, 15)	24471.60	20441.49	-16.47	(100, 50)	0.00	9093.19	100.00
(30, 15)	21332.73	20418.13	-4.29	(100, 50)	0.00	8769.78	100.00
(30, 15)	32280.79	21945.23	-32.02	(100, 50)	0.00	8677.54	100.00
(40, 20)	20671.24	19771.15	-4.35	(125, 65)	0.00	8421.56	100.00
(40, 20)	14790.42	17348.74	17.30	(125, 65)	0.00	8888.34	100.00
(40, 20)	14683.91	15264.96	3.96	(125, 65)	0.00	8499.20	100.00
(40, 20)	18811.16	14859.34	-21.01	(125, 65)	0.00	8418.64	100.00
(40, 20)	21881.87	16409.01	-25.01	(125, 65)	0.00	8721.65	100.00
(50, 25)	14632.38	15773.47	7.80	(150, 75)	0.00	5982.60	100.00
(50, 25)	18147.63	17285.11	-4.75	(150, 75)	0.00	6845.89	100.00
(50, 25)	9964.92	14860.34	49.13	(150, 75)	0.00	6409.58	100.00
(50, 25)	17297.98	15287.89	-11.62	(150, 75)	0.00	7017.86	100.00
(50, 25)	7889.01	14241.52	80.52	(150, 75)	0.00	6782.92	100.00
(60, 30)	11460.14	14240.83	24.26	(175, 90)	0.00	6161.06	100.00
(60, 30)	9269.97	15119.92	63.11	(175, 90)	0.00	6638.70	100.00
(60, 30)	15069.11	14074.29	-6.60	(175, 90)	0.00	6332.38	100.00
(60, 30)	12206.23	15258.13	25.00	(175, 90)	0.00	6238.45	100.00
(60, 30)	12908.33	14332.66	11.03	(175, 90)	0.00	5977.80	100.00
(70, 35)	8660.97	12665.14	46.23	(200, 100)	0.00	5786.38	100.00
(70, 35)	12680.03	11964.01	-5.65	(200, 100)	0.00	6205.23	100.00
(70, 35)	8366.05	13226.87	58.10	(200, 100)	0.00	7105.24	100.00
(70, 35)	12038.93	12650.81	5.08	(200, 100)	0.00	6141.99	100.00
(70, 35)	7604.81	11880.53	56.22	(200, 100)	0.00	6503.09	100.00
(80, 40)	10575.04	10365.54	-1.98	(300,150)	0.00	5659.57	100.00
(80, 40)	11717.39	9904.30	-15.47	(300,150)	0.00	5576.10	100.00
(80, 40)	9964.92	10734.55	7.72	(300,150)	0.00	5388.16	100.00
(80, 40)	0.00	12558.07	100.00	(300,150)	0.00	5165.94	100.00
(80, 40)	0.00	10226.38	100.00	(300,150)	0.00	5354.12	100.00

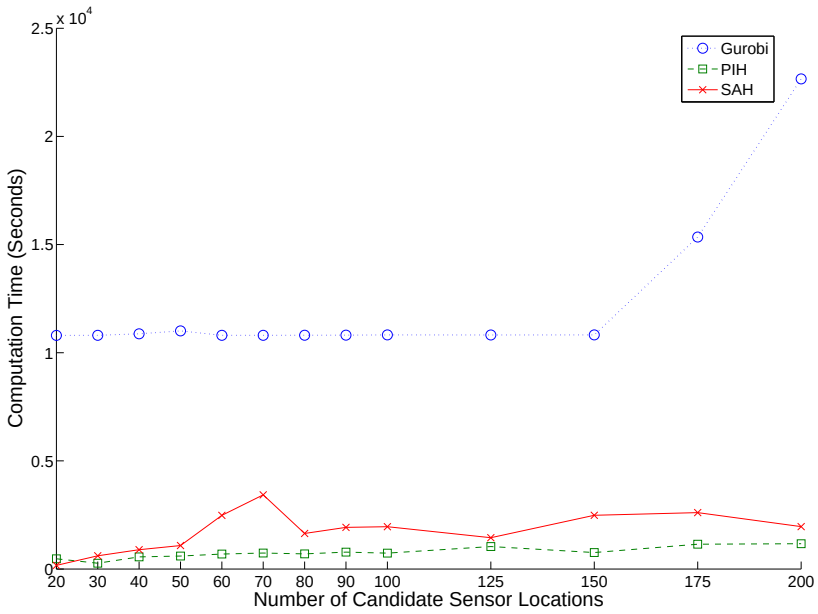

Figure 6.18. Average computation times for 5 sinks.

As can be observed from Table 6.5, Gurobi is able to obtain longer network lifetimes for most of the small instances, for networks having less than or equal to 40 candidate sensor locations. However, as the network size increases HH becomes more successful than Gurobi since network lifetimes obtained by Gurobi are longer than the ones calculated by HH in only 2 instances of the networks with 50 and 80 candidate sensor locations, and only in 1 instance of the networks with 60 and 70 candidate sensor locations. Moreover, as the network size becomes even larger, the success of HH becomes much clearer. Network lifetimes obtained by HH is longer than the ones obtained by Gurobi for all instances of the networks with 90 or more candidate sensor locations. In addition, Gurobi is unable to find a network with positive lifetime respectively for two, four and all instances of the networks with 80, 90 and more than 90 candidate sensor locations. For these instances, the percent deviation between the network lifetimes obtained by Gurobi and HH are assumed to be 100%.

We compute the averages of the network lifetimes found by Gurobi and HH for five instances corresponding to each of the different problem set with different number of candidate sensor locations and plot these averages against the number of candidate sensor locations in Figure 6.24 in order to be able to visualize the performance of HH.

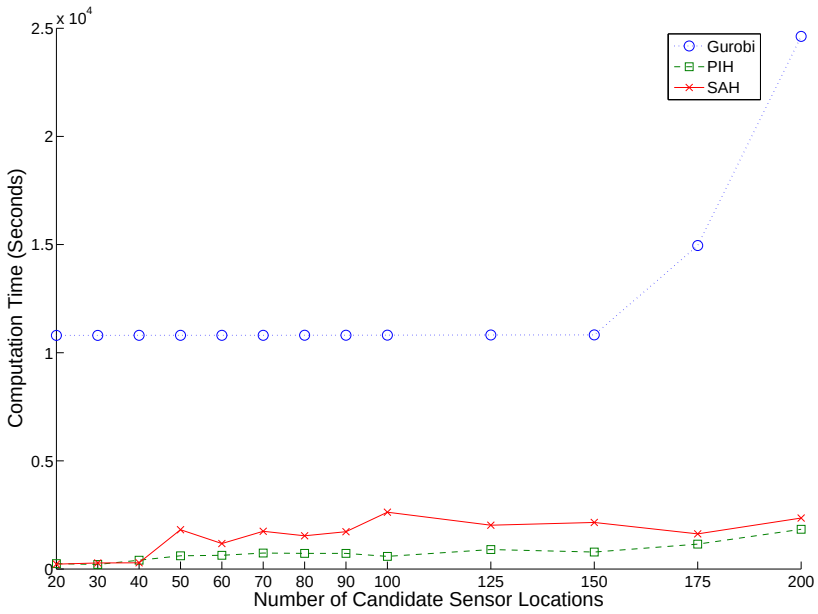

Figure 6.19. Average computation times for 7 sinks.

As can be observed from Figure 6.24 that, the network lifetimes obtained by Gurobi decrease rapidly as the network size increases and HH produces networks with longer lifetimes for 50 or more candidate sensor locations. Figure 6.24 also reveals that the network lifetimes decreases with the number of candidate sensor locations for both of the Gurobi and HH results. In the previous section, in which performance of PIH and SAH algorithms are elaborated, a similar observation for the lifetime decrease is already explained with the reasons behind it. These explanations are also valid here.

We also compute the averages of the percent deviations between the maximum network lifetimes obtained by Gurobi and the HH for each of the different problem set and plot them against the number of candidate sensor locations in Figure 6.25. A quick look at Figure 6.25 reveals that the network designer should refer to heuristic solution strategies like HH for the networks having 50 or more candidate sensor locations. Therefore, it makes more sense to use Gurobi for small networks, i.e., the ones with no more than 40 candidate sensor locations.

Besides, both Gurobi and HH reach the three-hour running time limit for most of the instances of the small networks and for all instances of the moderate and large

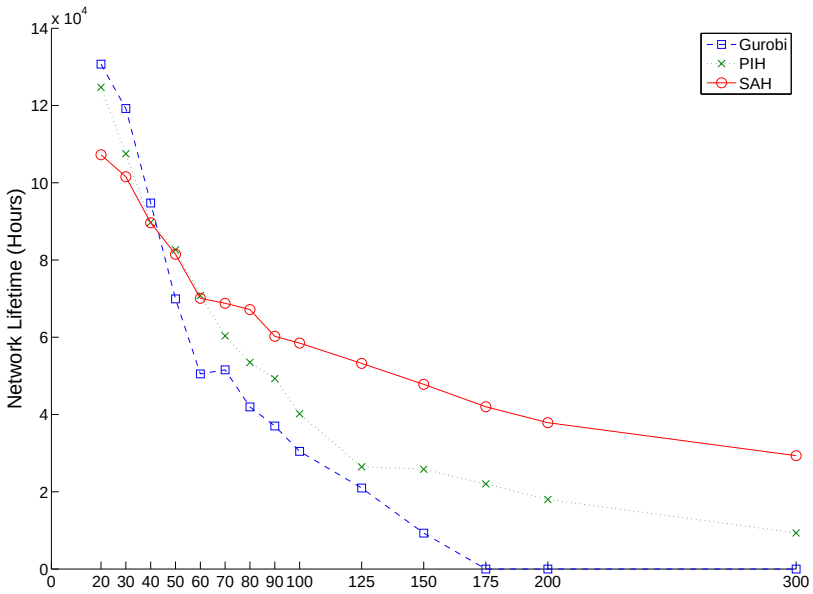

Figure 6.20. Maximum network lifetimes by Gurobi, PIH and SAH for $h_r = 2048$ bits/hour.

networks. When it comes to the quality of the produced networks, HH's performance dominates for most of the instances, as reported.

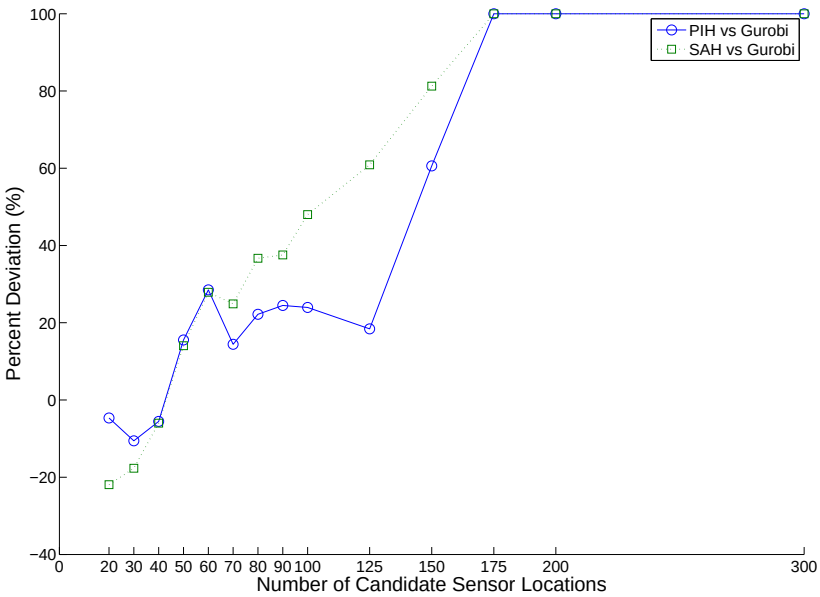

Figure 6.21. PIH and SAH vs. Gurobi: percent deviations for $h_r = 2048$ bits/hour.

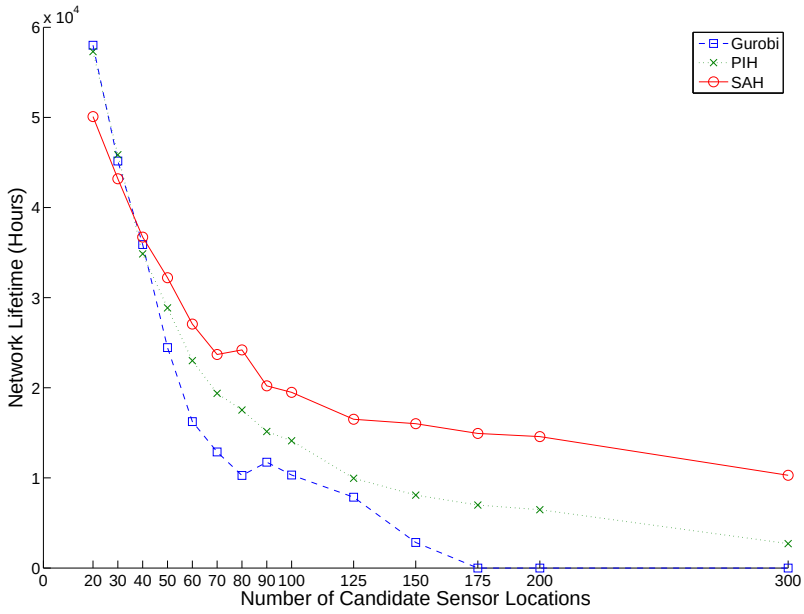

Figure 6.22. Maximum network lifetimes by Gurobi, PIH and SAH for $h_r = 8192$ bits/hour.

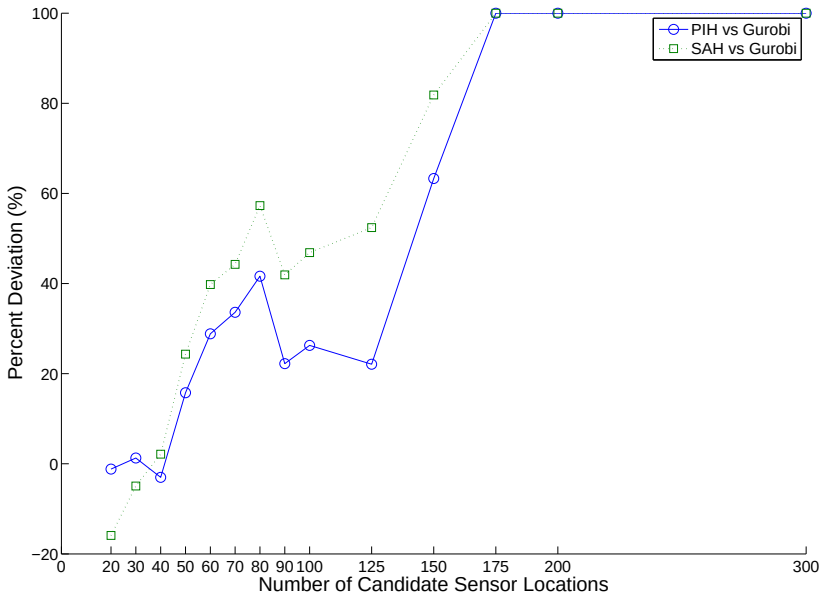

Figure 6.23. PIH and SAH vs. Gurobi: percent deviations for $h_r = 8192$ bits/hour.

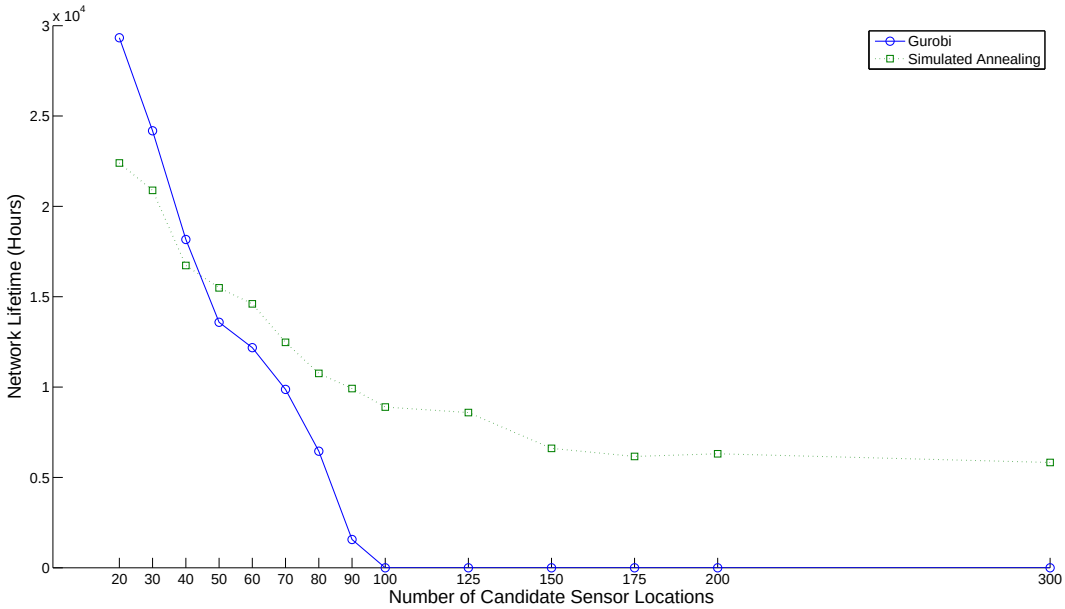

Figure 6.24. Gurobi vs. HH: maximum network lifetimes.

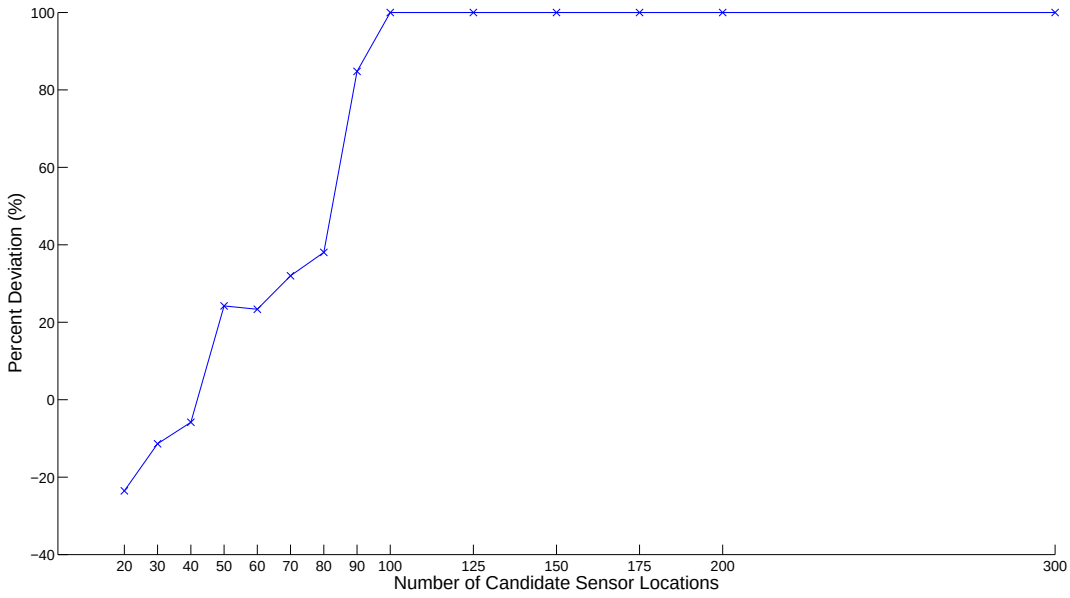

Figure 6.25. Gurobi vs. HH: percent deviations.

7. CONCLUSION

In this thesis, we first develop two MILP formulations, UHM and LHM, in which a single mobile sink with nonzero traveling time is considered. In both models, the sink is assumed to repeat the same tour throughout the network lifetime and the aim of the models are to find the sink tour maximizing the network lifetime. In the first model, each sensor is able to send its data to the sink either through direct connection to the sink if sink is nearby or through probably multiple hops over other sensors. In the latter model, we limit the number of hops that data pass through from the origin sensor to the sink. Therefore, sensors are allowed to send their data only if the sink is nearby so that they can transmit their data in the dictated hop limit. In both models, total sink travel times are considered as part of the network lifetime and the data accumulated during the sink travel times is taken into consideration in data flow balance and energy constraints. Both models turn out to be hard problems which led us to the heuristic solution strategies. We develop a tour construction heuristic (TCH) to initiate a tour and a tour improvement heuristic (TIH) to improve it. We run the heuristics together with the MILP solver Cplex over randomly generated problem instances for a fixed amount of time. The comparison of the network lifetimes found and the computation times required by the heuristics and the solver shows that they are both accurate and efficient. Moreover, we also find the percent improvement in the network lifetime achieved because of the consideration of the nonzero sink travel times. It is made by comparing the real network lifetimes corresponding to the tours found by TIH considering a nonzero sink travel time and the tour found assuming instantaneous sink jumps with zero travel times. We manage to show that taking nonzero sink traveling times into account considerably improves the network lifetime especially for the realistic sized networks and if the sink is to repeat high number of tours. Besides, we extend the model UHM to consider multiple mobile sinks with nonzero sink travel times and we show that the consideration of the sink travel times improves the network lifetimes only if the velocity of the sinks are very slow. Hence, we conclude that the sink travel times can be neglected under the presence of multiple mobile sinks.

Next, we develop several MILP models with increasing depth of the integration. First MILP model, MSLRP, deeply integrates the data routing (DRP) and sink routing (SRP) problems for multiple mobile sinks. Second one, MCSLRP, extends MSLRP by imposing the coverage problem (CP) on it. We also improve the model of Türkoğulları *et al.* (2010a), LSRP, by integrating DRP, activity scheduling problem (ASP) with sink location problem (SLP) with multiple stationary sinks. We manage to decrease the number of decision variables of the models of Türkoğulları *et al.* (2010a) considerably by letting the period lengths to be determined optimally by the model. The final model we present, MLSRP, achieves the highest level of of integration. It extends MCSLRP by imposing ASP on it, and extends LSRP by imposing SRP with multiple mobile sinks on it. Then, we show the positive effect of integrating major design issues in MILP formulations by comparing the objective values of the MSLRP, MCSLRP, LSRP and MLSRP. For instance, the comparison between the objective values of MSLRP and MCSLRP points the importance of considering the sensor deployment issue in the WSN design process. Similarly, we observe remarkable improvement in the network lifetime by considering sensor deployment and activity scheduling of the sensors in the WSN design process. Finally, by comparing the network lifetimes found by LSRP and MLSRP, we measure the improvement made by controlled sink mobility.

We finally provide heuristic solution procedures for MLSRP and LSRP models and a theoretical exact solution method for MLSRP. Period iteration heuristic (PIH) and sequential assignment heuristic (SAH) developed for the solution of MLSRP are able to produce much longer network lifetimes than the MILP solver Gurobi in much less computation time. Both heuristics depend on the idea of reducing the number of variables gradually in an iterative manner and fixing the values of the previously found variables in the following iterations. Besides, although branch-and-price is a promising widely used exact MILP solution methodology, its application for the solution of MLSRP turns out to be inefficient. On the other hand, HH algorithm developed for the solution of LSRP, employ a weighted Dantzig-Wolfe column generation procedure in every iteration for finding the optimum Lagrangean dual value for fixed sink locations while a simulated annealing algorithm perturbs the sink locations at the outer part. Numerical experiments show that HH algorithm can be preferred to MILP solvers

especially for networks with realistic sizes.

In summary, this thesis contributes to the literature in several aspects; they can be summarized as follows:

- Development of the MILP models for single and multiple mobile sink(s) in which nonzero sink travel times and the data accumulated during the sink travel times are taken into consideration. They are flexible enough to let unlimited and limited hop data routing protocols.
- Showing that considering nonzero sink travel times are very important for single sink case in which the sink repeats the same tour many times and the network is in realistic sizes.
- Showing that the sink travel times can be neglected for the case of multiple mobile sinks.
- Formulation of several MILP models with increasing breadth of integration of major WSN design problems and showing the improvement made by further integrations.
- Development of practical and intuitional heuristic procedures for MLSRP model which are shown to be accurate and efficient
- Development of a theoretical exact solution procedure for MLSRP
- Development of an efficient heuristic solution procedure for the solution of LSRP model of Türkoğulları *et al.* (2010a)

The perspective presented in this thesis can be extended in several directions. First of all, performance metrics of the WSNs created according to the proposed models and the heuristics can be simulated by some real life WSN simulators such as NS-2 (The Network Simulator, 2012), TOSSIM (Levis *et al.*, 2003), OMNeT++ (Varga and Hornig, 2008). By doing this, one can easily evaluate some other performance metrics other than network lifetime such as message latency and loss, throughput rates. This type of analysis helps also to assess whether the assumptions and the parameters of the models comply with the real conditions. Secondly, although very successful heuristic solution strategies and an exact solution approach are presented, this thesis lacks effi-

cient exact solution procedures. An attempt to make branch-and-price more efficient by reformulating the model or by referring to more efficient methods combining Dantzig-Wolfe and subgradient algorithms for the solution of Lagrangean dual will eventually make optimally solving realistic sized WSN design problems possible. Alternative linearization methods such as the one proposed by Sherali and Adams (1998) can be considered for increasing the efficiency of the branch-and-price algorithm. It is because the linear relaxation and/or Lagrangean bounds would be significantly improved if the linearization is accomplished without using the M's. As a third extension, one may try employing a new network lifetime definition in the formulations. For instance, defining the network lifetime as the time that passes until a specific percentage of the coverage is lost, and formulating appropriate MILP models with partial coverage constraints would be another research direction.

REFERENCES

Akyıldız, I. F., T. Melodia and K. R. Chowdhury, 2007, "A survey on wireless multimedia sensor networks", *Computer Networks*, Vol. 51, No. 4, pp. 921-960.

Akyıldız, I. F., W. Su, Y. Sankarasubramaniam and E. Çayırcı, 2002a, "A survey on sensor networks", *Communications Magazine*, Vol. 40, No. 8, pp. 102-114.

Akyıldız, I. F., W. Su, Y. Sankarasubramaniam and E. Çayırcı, 2002b, "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, No. 4, pp. 393-422.

Al-Karaki, J. N. and A. E. Kamal, 2004, "Routing techniques in wireless sensor networks: a survey", *Wireless Communications*, Vol. 11, No. 6, pp. 6-28.

Alsalih, W., S. Akl and H. Hassanein, 2007, "Placement of multiple mobile base stations in wireless sensor networks", *IEEE International Symposium on Signal Processing and Information Technology*, pp. 229-233.

Altınel, İ. K., N. Aras, E. Güney and C. Ersoy, 2008, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks", *Computer Networks*, Vol. 52, No. 12, pp. 2419-2431.

Applegate, D., R. Bixby, V. Chvatal and W. Cook, 2010, "Concorde TSP solver", http://www.tsp.gatech.edu/concorde, accessed at November 2010.

Azad, A. and A. Chockalingam, 2006, "Mobile base stations placement and energy aware routing in wireless sensor networks", *IEEE Wireless Communications and Networking Conference*, pp. 264-269.

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh and P. H. Vance, 1998, "Branch-and-price: column generation for solving huge integer programs", *Operations Research*, Vol. 46, No. 3, pp. 316-329.

Basagni, S., A. Carosi, E. Melachrinoudis, C. Petrioli and Z. M. Wang, 2008, "Controlled sink mobility for prolonging wireless sensor networks lifetime", *Wireless Networks*, Vol. 14, No. 6, pp. 831-858.

Basagni, S., A. Carosi, C. Petrioli and C. A. Phillips, 2011, "Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks", *Wireless Networks*, Vol. 17, No. 3, pp. 759-778.

Behdani, B., J. C. Smith and Y. Xia, 2013, "The Lifetime Maximization Problem in Wireless Sensor Networks with a Mobile Sink: MIP Formulations and Algorithms", *IIE Transactions*, Vol. 45, No. 10, pp. 1094-1113.

Behdani, B., Y. Yun, J. C. Smith and Y. Xia, 2012, "Decomposition algorithms for maximizing the lifetime of wireless sensor networks with mobile sinks", *Computers and Operations Research*, Vol. 39, No. 5, pp. 1054-1061.

Burrell, J., T. Brooke and R. Beckwith, 2004, "Vineyard computing: sensor networks in agricultural production", *Pervasive Computing*, Vol. 3, No. 1, pp. 38-45.

Butler, Z., P. Corke, R. Peterson, and D. Rus, 2004, "Virtual fences for controlling cows", *IEEE Conference on Robotics and Automation*, Vol. 5, pp. 4429-4436.

Callaway, E. H., 2004, "Wireless sensor networks: architectures and protocols", Vol. 3, CRC press, Florida.

Chang, J.-H. and L. Tassiulas, 2004, "Maximum lifetime routing in wireless sensor networks", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 4, pp. 609-619.

Cplex, 2007, "Cplex 11.0 User's Manual", http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, accessed at November 2010.

Culler, D., D. Estrin and M. Srivastava, 2004, "Guest editors' introduction: overview of sensor networks", *Computer*, pp. 41-49.

Dargie, W. and C. Poellabauer, 2010, "Fundamentals of wireless sensor networks: theory and practice", Wiley, Chichester, UK.

Desrochers, M. and G. Laporte, 1991, "Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints", *Operations Research Letters*, Vol. 10, No. 1, pp. 27-36.

Dishongh, T. and M. McGrath, 2009, "Wireless sensor networks for healthcare applications", Artech House, Norwood MA.

Fan, G. and S. Jin, 2010, "Coverage problem in wireless sensor network: a survey", *Journal of Networks*, Vol. 5, No. 9, pp. 1033-1040.

Gandham, S. R., M. Dawande, R. Prakash, and S. Venkatesan, 2003, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations", *IEEE Global Telecommunications Conference*, pp. 377-381.

Gatzianas, M. and L. Georgiadis, 2008, "A distributed algorithm for maximum lifetime routing in sensor networks with mobile sink", *IEEE Transactions on Wireless Communications*, Vol. 7, No. 3, pp. 984-994.

Geoffrion, A. M., 1994, "Structured modeling: survey and future research directions", *ORSA CSTS Newsletter*, Vol. 15, No. 1, pp. 1-20.

Ghosh, A. and S. K. Das, 2008, "Coverage and connectivity issues in wireless sensor networks: a survey", *Pervasive and Mobile Computing*, Vol. 4, No. 3, pp. 303-334.

Görtz, S. and A. Klose, 2009, "A subgradient based branch-and-bound for the capacitated facility location problem", Technical report, Department of Operations Research, University of Aarhus, Working paper no. 2009/1.

Gu, Y., D. Bozdag, E. Ekici, F. Ozguner, C.-G. Lee, 2005, "Partitioning based mobile element scheduling in wireless sensor networks", *IEEE SECON*, pp. 386-395.

Güney, E., N. Aras, İ. K. Altınel and C. Ersoy, 2010, "Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks", *Computer Networks*, Vol. 54, No. 11, pp. 1805-1822.

Güney, E., N. Aras, İ. K. Altınel and C. Ersoy, 2012, "Efficient solution techniques for the integrated coverage, sink location and routing problem in wireless sensor networks", *Computers and Operations Research*, Vol. 39, No. 7, pp. 1530-1539.

Gurobi, 2010, "Gurobi Optimizer 4.0, high-end libraries for math programming", http://www.gurobi.com/, accessed at May 2013.

Hamida, E. B. and G. Chelius, 2008, "Strategies for data dissemination to mobile sinks in wireless sensor networks", *Wireless Communications*, Vol. 15, No. 6, pp. 31-37.

Heinzelman, W. R., A. Chandrakasan and H. Balakrishnan, 2000, "Energy-efficient communication protocol for wireless microsensor networks", *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 10-19.

The Network Simulator, 2010, "The Network Simulator by Information Sciences Institute", http://www.isi.edu/nsam/ns/, accessed at May 2012.

Juang, P., H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, 2002, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet", *ACM Sigplan Notices*, Vol. 37, pp. 96-107.

Kappler, C. and G. Riegel, 2004, "A real-world, simple wireless sensor network for monitoring electrical energy consumption", *Wireless Sensor Networks*, Vol. 2920, pp. 339-352.

Keskin, M. E., İ. K. Altınel, N. Aras and C. Ersoy, 2011, "Lifetime Maximization in Wireless Sensor Networks Using a Mobile Sink with Nonzero Traveling Time", *The Computer Journal*, Vol. 54, No. 12, pp. 1987-1999.

Keskin, M. E., İ. K. Altınel, N. Aras and C. Ersoy, 2013a, "Wireless sensor network design by lifetime maximization: an empirical evaluation of integrating major design issues and sink mobility", *Revision submitted to International Journal of Sensor Networks*.

Keskin, M. E., İ. K. Altınel, N. Aras and C. Ersoy, 2013b, "Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility", *To Appear in Ad Hoc Networks*.

Keskin, M. E., İ. K. Altınel and N. Aras, 2013c, "A hybrid heuristic for optimal deployment, activity scheduling and data routing in wireless sensor networks", Technical report, Institute of Graduate Science and Engineering, Boğaziçi University, İstanbul, Report no. FBE-IE-02/2013-10.

Kirkpatrick, S., Gelatt C.D. and M. P. Vecchi, 1983, "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671-680.

Klose, A. and S. Görtz, 2007, "A branch-and-price algorithm for the capacitated facility location problem", *European Journal of Operational Research*, Vol. 179, No. 3, pp. 1109-1125.

Lee, S. H., S. Lee, H. Song and H. S. Lee, 2009, "Wireless sensor network design for tactical military applications: remote large-scale environments", *IEEE Military Communications Conference*, pp. 1-7.

Levis P., N. Lee, M. Welsh and D. Culler, 2003, "TOSSIM: accurate and scalable simulation of entire TinyOS applications", *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 126-137.

Li, J. and P. Mohapatra, 2007, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks", *Pervasive and Mobile Computing*, Vol. 3, No. 3, pp. 233-254.

Liang, W., J. Luo and X. Xu, 2010, "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks", *IEEE Global Telecommunications Conference*, pp. 1-6.

Luo, J. and J.-P. Hubaux, 2005, "Joint mobility and routing for lifetime elongation in wireless sensor networks", *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 3, pp. 1735-1746.

Luo, J. and J.-P. Hubaux, 2010, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: the case of constrained mobility", *IEEE/ACM Transactions on Networking*, Vol. 18, No. 3, pp. 871-884.

Madan, R. and S. Lall, 2006, "Distributed algorithms for maximum lifetime routing in wireless sensor networks", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 8, pp. 2185-2193.

Marshall, I. W., C. Roadknight, I. Wokoma and L. Sacks, 2003, "Self-organising sensor networks", *1st UK-UbiNet Workshop*.

Martinez, K., R. Ong, J. K. Hart and J. Stefanov, 2004, "GLACSWEB: a sensor web for glaciers", *Proceedings of European Workshop on Wireless Sensor Networks*, pp. 46-49.

Merrill, W. M., F. Newberg, K. Sohrabi, W. Kaiser and G. Pottie, 2003, "Collaborative networking requirements for unattended ground sensor systems", *IEEE Aerospace Conference Proceedings*, Vol. 5, pp. 2153-2165.

Michahelles, F., P. Matter, A. Schmidt and B. Schiele, 2003, "Applying wearable sensors to avalanche rescue", *Computers and Graphics*, Vol. 27, No. 6, pp. 839-847.

Miller, C. E., A. W. Tucker and R. A. Zemlin, 1960, "Integer programming formulation of traveling salesman problems", *Journal of the ACM*, Vol. 7, No. 4, pp. 326-329.

Nesamony, S., M. K. Vairamuthu, M. Orlowska and S. Sadiq, 2006, "On optimal route computation of mobile sink in a wireless sensor network", Technical report, School of information technology and Electrical Engineering, The university of Queensland, Report no. 465.

Öncan, T., İ. K. Altınel and G. Laporte, 2009, "A comparative analysis of several asymmetric traveling salesman problem formulations", *Computers and Operations Research*, Vol. 36, No. 3, pp. 637-654.

Papadimitriou, I. and L. Georgiadis, 2005, "Maximum lifetime routing to mobile sink in wireless sensor networks", *Proceedings of the 13th IEEE SoftCom*, pp. 1-5.

Polastre, J., R. Szewczyk, A. Mainwaring, S. Culler and J. Anderson, 2004, "Analysis of wireless sensor networks for habitat monitoring", *Wireless Sensor Networks*, pp. 399-423.

Popa, L., A. Rostamizadeh, R. Karp, C. Papadimitriou and I. Stoica, 2007, "Balancing traffic load in wireless networks with curveball routing", *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing*, Vol. 9, No. 14, pp. 170-179.

Rappaport, T. S., 1996, "Wireless communications: principles and practice", Vol. 2, Prentice Hall, PTR New Jersey.

Riem-Vis, R., 2004, "Cold chain management using an ultra low power wireless sensor network", *WAMES 2004*, pp. 21-23.

Sherali, H. D. and W. P. Adams, 1998, "A reformulation-linearization technique for solving discrete and continuous nonconvex problems", Springer, Dordrecht, Netherlands.

Sherali, H. D. and P. J. Driscoll, 2002, "On tightening the relaxations of Miller-Tucker-Zemlin formulations for asymmetric traveling salesman problems", *Operations Research*, Vol. 50, No. 4, pp. 656-669.

Shi, Y. and Y. T. Hou, 2008, "Theoretical results on base station movement problem for sensor network", *The 27th IEEE Conference on Computer Communications*, pp. 1-5.

Somasundara, A. A., A. Ramamoorthy and M. B. Srivastava, 2007, "Mobile element scheduling with dynamic deadlines", *IEEE Transactions on Mobile Computing*, Vol. 6, No. 4, pp. 395-410.

Soytürk, M. and T. Altılar, 2007, "A routing algorithm for mobile multiple sinks in large-scale wireless sensor networks", *IEEE Symposium on Wireless Pervasive Computing*, pp. 65-70.

Türkoğulları, Y. B., N. Aras and İ. K. Altınel, 2009, "Optimal placement, scheduling, and routing to maximize lifetime in sensor networks", *Journal of the Operational Research Society*, Vol. 61, No. 6, pp. 1000-1012.

Türkoğulları, Y. B., N. Aras, İ. K. Altınel and C. Ersoy, 2010a, "An efficient heuristic for placement, scheduling and routing in wireless sensor networks", *Ad Hoc Networks*, Vol. 8, No. 6, pp. 654-667.

Türkoğulları, Y. B., N. Aras, İ. K. Altınel and C. Ersoy, 2010b, "A column generation based heuristic for sensor placement, activity scheduling and data routing in wireless sensor networks", *European Journal of Operational Research*, Vol. 207, No. 2, pp. 1014-1026.

Valle, C. A., A. S. da Cunha, G. R. Mateus and W. M. Aioffi, 2009, "Optimization and Simulation in Wireless Sensor Networks with multiple mobile sinks", Unpublished manuscript.

Varga, A. and R. Hornig, 2008, "An overview of the OMNeT++ simulation environment", *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops*, pp. 60-69.

Vincze, Z., K. Fodor, R. Vida and A. Vidács, 2006, "Electrostatic modelling of multiple mobile sinks in wireless sensor networks", *Proceedings of the IFIP Networking Workshop on Performance Control in Wireless Sensor Networks*, pp. 30-37.

Wang, J. and N. Zhong, 2006, "Efficient point coverage in wireless sensor networks", *Journal of Combinatorial Optimization*, Vol. 11, No. 3, pp. 291-304.

Wang, Z. M., S. Basagni, E. Melachrinoudis and C. Petrioli, 2005, "Exploiting sink mobility for maximizing sensor networks lifetime", *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 287-295.

Wentges, P., 1997, "Weighted Dantzig–Wolfe Decomposition for Linear Mixed-integer Programming", *International Transactions in Operational Research*, Vol. 4, No. 2, pp. 151-162.

Werner-Allen, G., K. Lorincz, J. Johnson, J. Lees, and M. Welsh, 2006, "Fidelity and yield in a volcano monitoring sensor network", *Proceedings of the 7th symposium on Operating systems design and implementation*, pp. 381-396.

Winkler, M. K.-D. Tuchs, K. Hughes and G. Barclay, 2008, "Theoretical and practical aspects of military wireless sensor networks", *Journal of Telecommunications and Information Technology*, Vol. 2, pp. 37-45.

Wu, X., G. Chen and S. K. Das, 2008, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 5, pp. 710-720.

Yick, J., B. Mukherjee and D. Ghosal, 2008, "Wireless sensor network survey", *Computer Networks*, Vol. 52, No. 12, pp. 2292-2330.

Younis, M. and K. Akkaya, 2008, "Strategies and techniques for node placement in wireless sensor networks: a survey", *Ad Hoc Networks*, Vol. 6, No. 4, pp. 621-655.

Yun, Y. and Y. Xia, 2010, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications", *IEEE Transactions on Mobile Computing*, Vol. 9, No. 9, pp. 1308-1318.

Yun, Y., Y. Xia, B. Behdani and J.C. Smith, 2010, "Distributed algorithm for lifetime maximization in delay-tolerant wireless sensor network with mobile sink", *49th IEEE Conference on Decision and Control*, pp. 370-375.

Zuniga, M. and B Krishnamachari, 2004, "Analyzing the transitional region in low power wireless links", *First Annual Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 517-526.