Generating Optimal Scheduling for Wireless Sensor Networks by Using Optimization Modulo Theories Solvers

Gergely Kovásznai, Csaba Biró and Balázs Erdélyi

IoT Research Institute
Eszterhazy Karoly University
Eger, Hungary
iot.uni-eszterhazy.hu/en

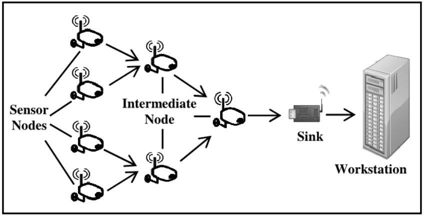
SMT 2017
July 22, 2017
Heidelberg, Germany

IoT applications, WSNs

Internet of Things (IoT) includes the use of small, inexpensive, self-powered devices that can sense their environment. Typically in

- agriculture,
- industry,
- security,
- environmental and habitat monitoring,
- traffic monitoring,
- military,
- etc.

Typically, they communicate wirelessly.



Outline

- Security and dependability constraints: coverage, evasive and moving target
- Lifetime maximization
- SMT-based approaches
- OMT problem formalization
- WSN simulation
- Benchmarks
- Experiments
- Conclusions and future work

Security and dependability constraints: coverage

How well the sensors observe the physical environment? Two main types:

- *Area coverage* to cover a given area of interest;
- *Point coverage* to cover a set of target points.

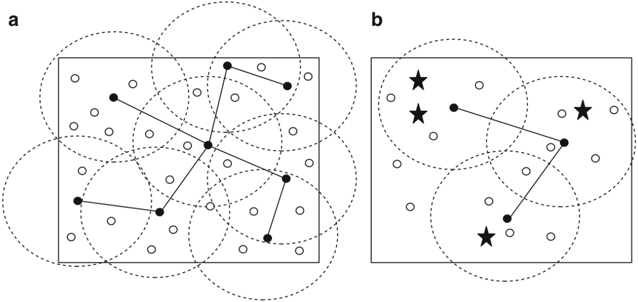


Fig. 1 Coverage types. (**a**) Area coverage. (**b**) Point coverage

[M. Cardei. Coverage problems in sensor networks. Handbook of Combinatorial Optimization, 2013.]

Point coverage can be used to simulate area coverage by using points that approximate an area.

Security and dependability constraints: evasive and moving target constraints

Additional security requirements in critical systems and in military applications, in order to protect sensor nodes to be damaged, detected or attacked.

Evasive constraint. To prohibit the sensor nodes to be active for too long.

Moving target constraint. Not to cover critical target points by the same sensor node for too long.

Lifetime maximization

The aim is to maximize the WSN's lifetime.

Why?

- Sensor nodes are self-powered and have limited power supply.

How?

- By sending certain sensor nodes into sleep mode and waking them up later on, in a synchronized way.

Let's generate a sleep/wake-up scheduling which does not violate the constraints at any time and provides a maximal lifetime for the WSN!

Heuristic optimization approaches for coverage solving

Most of previous works deal WSN lifetime maximization as an optimization problem by applying heuristics.
[M. Cardei, D. Ding-Zhu. Improving wireless sensor network lifetime through power aware organization. Wireless Networks, 2005.]
[D. Tian, N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. WSNA, 2002.]

- They scale up to a few hundred sensor nodes and tens of target points.
- They sometimes sacrifice 100% precise coverage.
- They focus on the *coverage problem*, without giving attention to other security/dependability constraints (e.g. evasive, moving target).

SMT-based approaches

A few previous works apply SMT solving to generate a sleep/wake-up scheduling that respects WSN constraints.

[K. Weiqiang et al. An SMT-based accurate algorithm for the K-coverage problem in sensor network. UBICOMM, 2014.]
Focuses only on coverage. Deals only with homogeneous nodes. Reports experiments with Z3.

[Q. Duan et al. Provable configuration planning for wireless sensor networks. CNSM, 2012.]
Addresses several constraints. Deals only with homogeneous nodes. Reports experiments with YICES. Scales up to hundreds of nodes.

None of them addresses lifetime maximization.
Shall we try to apply OMT solvers?

SMT-based approaches

A few previous works apply SMT solving to generate a sleep/wake-up scheduling that respects WSN constraints.

[K. Weiqiang et al. An SMT-based accurate algorithm for the K-coverage problem in sensor network. UBICOMM, 2014.]
Focuses only on coverage. Deals only with homogeneous nodes. Reports experiments with Z3.

[Q. Duan et al. Provable configuration planning for wireless sensor networks. CNSM, 2012.]
Addresses several constraints. Deals only with homogeneous nodes. Reports experiments with YICES. Scales up to hundreds of nodes.

None of them addresses lifetime maximization.
Shall we try to apply OMT solvers?

SMT-based approaches

A few previous works apply SMT solving to generate a sleep/wake-up scheduling that respects WSN constraints.

[K. Weiqiang et al. An SMT-based accurate algorithm for the K-coverage problem in sensor network. UBICOMM, 2014.]
Focuses only on coverage. Deals only with homogeneous nodes. Reports experiments with Z3.

[Q. Duan et al. Provable configuration planning for wireless sensor networks. CNSM, 2012.]
Addresses several constraints. Deals only with homogeneous nodes. Reports experiments with YICES. Scales up to hundreds of nodes.

None of them addresses lifetime maximization.
Shall we try to apply OMT solvers?

SMT-based approaches

A few previous works apply SMT solving to generate a sleep/wake-up scheduling that respects WSN constraints.

[K. Weiqiang et al. An SMT-based accurate algorithm for the K-coverage problem in sensor network. UBICOMM, 2014.]
Focuses only on coverage. Deals only with homogeneous nodes. Reports experiments with Z3.

[Q. Duan et al. Provable configuration planning for wireless sensor networks. CNSM, 2012.]
Addresses several constraints. Deals only with homogeneous nodes. Reports experiments with YICES. Scales up to hundreds of nodes.

None of them addresses lifetime maximization.
Shall we try to apply OMT solvers?

What is the plan?

1. Let us introduce an OMT formalization of the lifetime maximization problem for WSNs where
   - all of the coverage, evasive and moving target constraints are addressed,
   - sensor nodes are heterogeneous (i.e., they have different sensing ranges).
2. Let us perform experiments with existing OMT solvers: OptiMathSAT, Z3, Symba.
   [R. Sebastiani, P. Trentin. OptiMathSAT: A tool for optimization modulo theories. CAV, 2015.]
   [N. Bjørner et al. μZ - An optimizing SMT solver. TACAS, 2015.]
   [Y. Li et al. Symbolic optimization with SMT solvers. POPL, 2014.]
3. Let us provide new and practical OMT benchmarks for the SMT community.

OMT formalization

Let us introduce the following notations:

$n \geq 1$: number of sensor nodes

$m \geq 1$: number of target points

r_i: the sensing range of the i^{th} node

L_i: the lifetime of the i^{th} node

$d_{i,j}$: the distance between the i^{th} node and the j^{th} point

T: the WSN's lifetime

- This is what we want to maximize.

$w_{i,t}$: Boolean variable that denotes if the i^{th} node is awake at the t^{th} time interval

- We are looking for a satisfying assigment to the variables.

OMT formalization

Let us introduce the following notations:

- $n \geq 1$: number of sensor nodes
- $m \geq 1$: number of target points
- r_i: the sensing range of the i^{th} node
- L_i: the lifetime of the i^{th} node
- $d_{i,j}$: the distance between the i^{th} node and the j^{th} point
- T: the WSN's lifetime
  - This is what we want to maximize.
- $w_{i,t}$: Boolean variable that denotes if the i^{th} node is awake at the t^{th} time interval
  - We are looking for a satisfying assigment to the variables.

OMT formalization

Let us introduce the following notations:

- $n \geq 1$: number of sensor nodes
- $m \geq 1$: number of target points
- r_i: the sensing range of the i^{th} node
- L_i: the lifetime of the i^{th} node
- $d_{i,j}$: the distance between the i^{th} node and the j^{th} point

- T: the WSN's lifetime
  - This is what we want to maximize.

- $w_{i,t}$: Boolean variable that denotes if the i^{th} node is awake at the t^{th} time interval
  - We are looking for a satisfying assigment to the variables.

OMT formalization

Let us introduce the following notations:

$n \geq 1$: number of sensor nodes

$m \geq 1$: number of target points

r_i: the sensing range of the i^{th} node

L_i: the lifetime of the i^{th} node

$d_{i,j}$: the distance between the i^{th} node and the j^{th} point

T: the WSN's lifetime

- This is what we want to maximize.

$w_{i,t}$: Boolean variable that denotes if the i^{th} node is awake at the t^{th} time interval

- We are looking for a satisfying assigment to the variables.

OMT formalization

Let us introduce the following notations:

$n \geq 1$: number of sensor nodes

$m \geq 1$: number of target points

r_i: the sensing range of the i^{th} node

L_i: the lifetime of the i^{th} node

$d_{i,j}$: the distance between the i^{th} node and the j^{th} point

T: the WSN's lifetime

- This is what we want to maximize.

$w_{i,t}$: Boolean variable that denotes if the i^{th} node is awake at the t^{th} time interval

- We are looking for a satisfying assigment to the variables.

OMT formalization – Lifetime constraint

For each node, the number of time intervals at which the node is awake must not exceed the node's lifetime.

$$\forall i \ (1 \leq i \leq n). \ \sum_{t=1}^{T} w_{i,t} \leq L_i$$

SMT-LIB formalization:

```
(<=
    (+  (boolToInt (w i 0))
        (boolToInt (w i 1))
        ... (boolToInt (w i T)) )
    (L i)  )
```

OMT formalization – Lifetime constraint

For each node, the number of time intervals at which the node is awake must not exceed the node's lifetime.

$$\forall i\ (1 \leq i \leq n).\ \sum_{t=1}^{T} w_{i,t} \leq L_i$$

SMT-LIB formalization:

```
(<=
    (+  (boolToInt (w i 0))
        (boolToInt (w i 1))
       ... (boolToInt (w i T)) )
    (L i)  )
```

OMT formalization – K-coverage constraint

Each point is covered by at least $K \geq 1$ sensor nodes.

$$\forall j, t \; (1 \leq j \leq m, \; 1 \leq t \leq T). \; \sum_{i \in S_j} w_{i,t} \geq K$$

where $S_j = \{i \mid d_{i,j} \leq r_i\}$ is the set of nodes which are able to cover the j^{th} point.

SMT-LIB formalization:

```
(>=
    (+  (boolToInt (covers0jAt t))
        (boolToInt (covers1jAt t))
        ... (boolToInt (coversnjAt t)) )
    K )
```

OMT formalization – K-coverage constraint

Each point is covered by at least $K \geq 1$ sensor nodes.

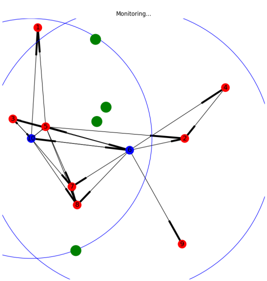
$$\forall j, t \; (1 \leq j \leq m, \; 1 \leq t \leq T). \; \sum_{i \in S_j} w_{i,t} \geq K$$

where $S_j = \{i \mid d_{i,j} \leq r_i\}$ is the set of nodes which are able to cover the j^{th} point.

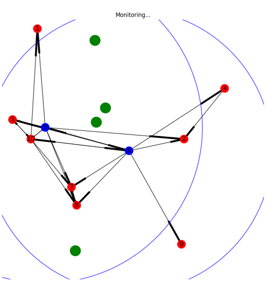
SMT-LIB formalization:

```
(>=
    (+  (boolToInt (covers0jAt t))
        (boolToInt (covers1jAt t))
        ... (boolToInt (coversnjAt t)) )
    K )
```

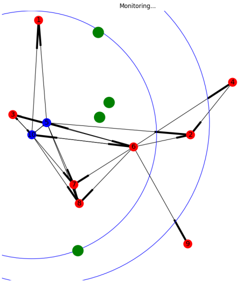
Illustration – K-coverage constraint



(a) $t = 0$



(b) $t = 1$



(c) $t = 2$

Figure: Sleep/wake-up scheduling of sensor nodes for 2-coverage and evasive constraint with $E = 2$. The active nodes (blue dots) are monitoring the target points (green dots).

OMT formalization – Evasive constraint

A node must not stay awake for more than $E \geq 1$ consecutive time intervals.

$$\forall i, t \ (1 \leq i \leq n, \ 1 \leq t \leq T - E). \sum_{t'=t}^{t+E} w_{i,t'} \leq E$$

SMT-LIB formalization:

```
(<=
    (+  (boolToInt (w i t))
        (boolToInt (w i t+1))
        ... (boolToInt (w i t+E)) )
    E )
```

OMT formalization – Evasive constraint

A node must not stay awake for more than $E \geq 1$ consecutive time intervals.

$$\forall i, t \ (1 \leq i \leq n, \ 1 \leq t \leq T - E). \ \sum_{t'=t}^{t+E} w_{i,t'} \leq E$$

SMT-LIB formalization:

```
(<=
    (+  (boolToInt (w i t))
        (boolToInt (w i t+1))
        ... (boolToInt (w i t+E)) )
    E )
```

OMT formalization – Moving target constraint

Some critical points may require not to be covered by the same sensor node for more than $M \geq 1$ consecutive time intervals.

$$\forall j \in CR,\ \forall i \in S_j,\ \forall t\ (1 \leq t \leq T - M).\ \sum_{t'=t}^{t+M} w_{i,t'} \leq M$$

where $CR \subseteq \{j \mid 1 \leq j \leq m\}$ is the set of critical points.

SMT-LIB formalization: Similar to the one for the evasive constraint.

OMT formalization – Objective function

The aim is to maximize the WNS's lifetime.

$$\max : T$$

SMT-LIB formalization:

- For Z3:

```
(maximize T)
```

- For OptiMathSAT:

```
(maximize T :local-lb 0 :local-ub T̂)
```

where $\widehat{T} = \sum_{i=1}^{n} L_i$ works as a time horizon for the WSN.

- For Symba:

```
(=>  $constraints (<= T TOpt)  )
```

OMT benchmarks – WSN simulation

For simulating WSNs, we chose an IEEE 802.15.4 compatible sensor node that is able to communicate wirelessly and has common parameters such as a $3V$ power supply.
A good example is the commonly used sensor node MICAz.

Such sensor nodes are provided with an RF transceiver with an estimated range of $100m$, such as the commonly used CC2420.

OMT benchmarks – WSN simulation

The manufacturer of CC2420 publish data about 8 performance levels. We need to calculate the sensing range for each performance level.

PA_LEVEL	Output Power P (mW)	Power Consumption I (mA)	Estimated Range r (m)
31	1.000	17.4	120
27	0.794	16.5	?
23	0.501	15.2	?
19	0.316	13.9	?
15	0.200	12.5	?
11	0.100	11.2	?
7	0.032	9.9	?
3	0.003	8.5	?

We can calculate the minimum range from the minimum output power:

$$r_{min} = \sqrt{P_{min} r_{max}^2}$$

The estimated range r for the power consumption I can be calculated as follows:

$$r = \frac{r_{max} - r_{min}}{P_{max} - P_{min}}(I - I_{min}) + r_{min}$$

OMT benchmarks – WSN simulation

The manufacturer of CC2420 publish data about 8 performance levels. We need to calculate the sensing range for each performance level.

PA_LEVEL	Output Power P (mW)	Power Consumption I (mA)	Estimated Range r (m)
31	1.000	17.4	120
27	0.794	16.5	109
23	0.501	15.2	92
19	0.316	13.9	75
15	0.200	12.5	58
11	0.100	11.2	41
7	0.032	9.9	25
3	0.003	8.5	6.75

We can calculate the minimum range from the minimum output power:

$$r_{min} = \sqrt{P_{min} r_{max}^2}$$

The estimated range r for the power consumption I can be calculated as follows:

$$r = \frac{r_{max} - r_{min}}{P_{max} - P_{min}} (I - I_{min}) + r_{min}$$

OMT benchmarks

A network grid of 600 meters by 600 meters. Random locations for sensor nodes and target points. Random performance levels.

We generated two benchmark sets:

Harder benchmarks: 10 sensor nodes, 4 target points, 2-coverage constraint, evasive constraint with $E = 3$, and moving target constraint with $M = 2$.

Easier benchmarks: 10 sensor nodes, 2 target points, 1-coverage constraint, evasive constraint with $E = 2$, and moving target constraint with $M = 1$.

Within each benchmark set, we generated

1. 20 benchmarks with all the constraints enabled,
2. 20 benchmarks with only the moving target constraint disabled, and
3. 20 benchmarks with only the evasive constraint disabled.

3 variants of each benchmark instance: for OptiMathSAT, Z3, and Symba, respectively.

OMT benchmarks

A network grid of 600 meters by 600 meters. Random locations for sensor nodes and target points. Random performance levels.

We generated two benchmark sets:

Harder benchmarks: 10 sensor nodes, 4 target points, 2-coverage constraint, evasive constraint with $E = 3$, and moving target constraint with $M = 2$.

Easier benchmarks: 10 sensor nodes, 2 target points, 1-coverage constraint, evasive constraint with $E = 2$, and moving target constraint with $M = 1$.

Within each benchmark set, we generated

1. 20 benchmarks with all the constraints enabled,
2. 20 benchmarks with only the moving target constraint disabled, and
3. 20 benchmarks with only the evasive constraint disabled.

3 variants of each benchmark instance: for OptiMathSAT, Z3, and Symba, respectively.

OMT benchmarks

A network grid of 600 meters by 600 meters. Random locations for sensor nodes and target points. Random performance levels.

We generated two benchmark sets:

Harder benchmarks: 10 sensor nodes, 4 target points, 2-coverage constraint, evasive constraint with $E = 3$, and moving target constraint with $M = 2$.

Easier benchmarks: 10 sensor nodes, 2 target points, 1-coverage constraint, evasive constraint with $E = 2$, and moving target constraint with $M = 1$.

Within each benchmark set, we generated

1. 20 benchmarks with all the constraints enabled,
2. 20 benchmarks with only the moving target constraint disabled, and
3. 20 benchmarks with only the evasive constraint disabled.

3 variants of each benchmark instance: for OptiMathSAT, Z3, and Symba, respectively.

OMT benchmarks

A network grid of 600 meters by 600 meters. Random locations for sensor nodes and target points. Random performance levels.

We generated two benchmark sets:

Harder benchmarks: 10 sensor nodes, 4 target points, 2-coverage constraint, evasive constraint with $E = 3$, and moving target constraint with $M = 2$.

Easier benchmarks: 10 sensor nodes, 2 target points, 1-coverage constraint, evasive constraint with $E = 2$, and moving target constraint with $M = 1$.

Within each benchmark set, we generated

1. 20 benchmarks with all the constraints enabled,
2. 20 benchmarks with only the moving target constraint disabled, and
3. 20 benchmarks with only the evasive constraint disabled.

3 variants of each benchmark instance: for OptiMathSAT, Z3, and Symba, respectively.

Experiments

Experiments were run on 3.60 GHz 8-core CPU with 8 GB memory. Time limit: 600 seconds. Memory limit: 3 GB.

Results for the harder benchmarks over QF_UFLIA:

	Solver	*#SAT/UNS*	*#TO*	*Opt*	*Time*	*Space*	*#Crash*
All constraint on	OptiMathSAT	11/5	4	73	245.5	440.3	
	Z3	2/5	13	72	393.6	449.8	
	Symba	1/5	14	2	423.1	461.9	
Moving target off	OptiMathSAT	10/5	5	74	215.8	314.8	
	Z3	7/5	8	73	258.2	408.3	
	Symba	4/5	11	60	393.5	439.1	
Evasive off	OptiMathSAT	12/4	3	74	149.0	286.5	1
	Z3	7/5	8	72	265.2	468.5	
	Symba	5/5	10	60	355.6	475.0	

Experiments

Experiments were run on 3.60 GHz 8-core CPU with 8 GB memory. Time limit: 600 seconds. Memory limit: 3 GB.

Results for the harder benchmarks over QF_UFLIA:

	Solver	*#SAT/UNS*	*#TO*	*Opt*	*Time*	*Space*	*#Crash*
All constraint on	OptiMathSAT	11/5	4	73	245.5	440.3	
	Z3	2/5	13	72	393.6	449.8	
	Symba	1/5	14	2	423.1	461.9	
Moving target off	OptiMathSAT	10/5	5	74	215.8	314.8	
	Z3	7/5	8	73	258.2	408.3	
	Symba	4/5	11	60	393.5	439.1	
Evasive off	OptiMathSAT	12/4	3	74	149.0	286.5	1
	Z3	7/5	8	72	265.2	468.5	
	Symba	5/5	10	60	355.6	475.0	

Experiments

Results for the easier benchmarks over QF_UFLIA:

	Solver	*#SAT/UNSAT*	*#TO*	*Opt*	*Time*	*Space*
All constraints on	Z3	20/0	0	226	63.1	493.1
	OptiMathSAT	12/0	8	159	277.8	520.7
	Symba	9/0	11	159	484.9	436.2
Moving target off	Z3	20/0	0	226	49.2	333.7
	OptiMathSAT	11/0	9	130	311.3	476.1
	Symba	9/0	11	159	488.8	340.0
Evasive off	Z3	20/0	0	226	50.8	300.0
	Symba	19/0	1	226	324.8	334.1
	OptiMathSAT	12/0	8	159	344.1	488.2

Experiments

Results for the easier benchmarks over QF_UFLRA:

	Solver	*#SAT/UNSAT*	*#TO*	*Opt*	*Time*	*Space*
All constraints on	Z3	19/0	1	226	87.7	497.1
	SYMBA	18/0	2	226	223.4	578.5
	OPTIMATHSAT	12/0	8	159	277.0	507.4
Moving target off	Z3	20/0	0	226	41.4	318.9
	SYMBA	20/0	0	226	172.5	393.0
	OPTIMATHSAT	11/0	9	130	310.6	469.9
Evasive off	Z3	20/0	0	226	36.2	284.4
	SYMBA	20/0	0	226	128.9	340.5
	OPTIMATHSAT	11/0	9	159	345.2	467.5

Conclusion and future work

Conclusions:

- OptiMathSAT provides the most stable performance and scales the best
- Z3 is efficient on fewer target points and lower parameter values
- Symba provides convincing performance over real numbers

Future work:

- To do experiments with different parameter settings for the OMT solvers
- To provide benchmarks with larger network grid and higher number of nodes/points
- To come up with an specialized OMT approach for WSN-like optimization problems
  - Based on the energy-consumption values, we could predict when the WSN runs out of energy
  - Preliminary, convincing results for 40-50 nodes

Conclusion and future work

Conclusions:

- OptiMathSAT provides the most stable performance and scales the best
- Z3 is efficient on fewer target points and lower parameter values
- Symba provides convincing performance over real numbers

Future work:

- To do experiments with different parameter settings for the OMT solvers
- To provide benchmarks with larger network grid and higher number of nodes/points
- To come up with an specialized OMT approach for WSN-like optimization problems
  - Based on the energy-consumption values, we could predict when the WSN runs out of energy
  - Preliminary, convincing results for 40-50 nodes