EFFICIENT MONITORING AND RESOURCE MANAGEMENT WITH SENSOR NETWORKS

Theses of the Ph.D. Dissertation

Kálmán Tornai, M.Sc.

Scientific adviser:
János Levendovszky, D.Sc.
Doctor of the Hungarian Academy of Sciences

Péter Pázmány Catholic University
Faculty of Information Technology and Bionics
Multidisciplinary Technical Sciences Doctoral School

Budapest
2013

1. Introduction and objectives

Due to the recent improvements of communication and sensor technology several applications have been developed which satisfy the requirements of mobility and flexibility. Primarily these applications focus on monitoring complex systems, and detecting abnormal events in these systems.
However the available resources (battery, radio transmission power, radio bandwidth), which are needed for data transmissions, are limited. These limitations impose considerable restrictions on i) rate of transmission; ii) lifespan of the nodes in the network; iii) spectral efficiency. In efficient monitoring systems these limitations have to be taken into consideration. The current research is focusing on such monitoring systems wherein best possible performance is available under the constraints mentioned before. The scope of problem concerns several wireless, mobile technologies where such restrictions are in place. In this dissertation a specific class of applications is going to be detailed, examined and optimized.

I am going to introduce novel algorithms and methods to solve the following problems: 1) optimal media access control; 2) detecting critical events with low error probability analyzing sensed values, in monitoring systems where wireless sensor networks are applied. These algorithms and methods can not only be used in WSN applications. The scheduler which is going to be introduced can be applied in several problems, where equipment with finite capacity has to be scheduled, such as call admission control in telecommunication systems, memory and processor management in general purpose computing systems or in financial computational systems. The techniques and mathematical results of event detection method can be deployed in problems such as processing time series. They are capable of detecting and forecasting anomalies and outlier values, furthermore they can answer to challenges in the theory of decision. The scheme can be adapted to be used in distributed computing systems, and crowdsourcing applications.

1.1 *Introduction – technological motivations and existing results*

In this section the existing results are discussed and some open questions are posed.

1.1.1 The global scheme of monitoring systems

The challenges of WSN based monitoring systems are summarized by Figure 1. The dissertation focuses on the problems of the WSN transmission protocols and the problems of data processing. More precisely my theses concern with (i) resource management and packet scheduling; and (ii) detection of unusual measurements and abnormal events.
In the rest of this section I detail the existing results for the problems described above.

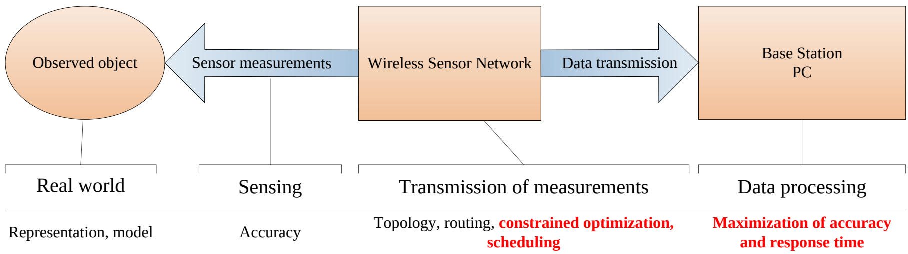

Figure 1. Research areas of WSN based monitoring systems. The investigated areas of the dissertation are highlighted.

1.1.2 Resource management and packet scheduling

The complex data transmissions and computations of a monitoring network require efficient scheduling algorithms, which are able to provide optimal solutions in the case of limited capacity. The monitoring systems I develop in the thesis need optimal scheduling.
With the advent of grid computing, the classical science of scheduling theory has gained a new sphere of applications. Methods which were originally developed for decision making around limited resources in manufacturing and service industries have been adapted in the areas of computer science, telecommunication and other computationally intensive disciplines such as computational biology, chemistry and finance [1], [2]. Generally, the task is the following: we have to find the appropriate schedule of jobs by using a finite resource taking into account the demands, the time constrains, priorities and also the given capacity should never be exceeded.
In the problem which I have investigated each job has a priority and a time constraint within the job has to be finished. The jobs can generally be stopped and resumed at a later point on a different machine which is referred to as preemption in scheduling theory. The serving machines are generally assumed to be identical and there are a

known, constant number of machines available. Nowadays there is no such an algorithm or method which solves the previously defined problem in polynomial time. Sahni [3] presents an $\mathcal{O}(n \log nm)$ algorithm to construct a feasible schedule, one that meets all deadlines, if one exists, for n jobs and m machines. This result has been extended to machines with identical functionality but different processing speed, termed uniform machines, and jobs with both starting times and deadlines [4]. However, the scheduling task becomes more difficult when a feasible schedule does not exist. In this case one possible objective is to minimize the *total tardiness* (TT) or to uniformly distribute the number of jobs over capacity. (Tardiness of an individual job under a given schedule is defined as the amount of time by which the job finishes after its prescribed deadline, and is considered to be zero if the job finishes on or before the deadline.)

In the case of minimizing the maximum tardiness across all jobs, Lawler [5] shows that the problem is solvable in polynomial time, even with some precedence constraints. Martel [4] also used this construction to create an algorithm. However the TT problem even with single machine was proven to be NP-hard [6]. A pseudo polynomial algorithm has been developed by Lawler [7] for this problem, but it does not have good practical runtime characteristics. In practical applications, jobs often have relative priorities associated with them, represented by positive real weights and the objective becomes minimizing the *total weighted tardiness* (TWT). Azizoglu et al. [8] worked on an algorithm to find optimal schedule for the not weighted TT problem without preemption, but their branch and bound algorithm is too slow, in practice, for problems with more than 15 jobs. Rachamadugu et al. [9] provides a solution for TWT problem without preemption.

The data gathering with the use of WSNs in monitoring systems has energy and complexity constrains, therefore the optimization of resource is crucial. The task of resource scheduling in WSNs are the followings: i) the schedule of communication channel; ii) the schedule of the low-power processing unit. In the case of application investigated in this dissertation the critical tasks are: the media access control (MAC) and packet scheduling. There are several classical protocols, however mostly of these protocols cannot be used in WSN, where the energy efficiency is crucial. There are two main groups of MAC protocols: contention-based random access with collision sensing; and scheduling based (TDMA) methods.

In the contention-based scheme, nodes try to access the channel instantly, in order to increase the throughput and to minimize the end-to-end latency. (This task is equivalent to the TT problem.) The great advantage of the TDMA protocols is the energy efficiency, because there is no need for detecting packed collisions, retransmissions and also the nodes only wake up during its own slot time. In case of TDMA the deterministic data traffic and the existence of spanning tree of the network is useful and this requisitions are often satisfied in monitoring systems.

In energy limited environment the S-MAC protocol [11] sends nodes into sleeping mode regularly, and B-MAC [12] uses low power consumption strategy for waking

up nodes. X-MAC [13] is an improved version of B-MAC protocol, which reduces the energy consumptions due to the overhearing.
In the case of using TDMA protocols the end-to-end latency is high, the throughput of the network is low. This latency can be improved by altering the traditional timeslot assignment strategy [14]. The TreeMAC protocol [15] has been designed for WSN taking into consideration the bandwidth demand of nodes. Combining the advantages of contention based and TDMA protocols several solutions have been developed using the cross-layer developing approach. Protocols of Goldsmith [16] and Jurdak [17] provide solutions for both routing, MAC and optimal transmission power.
As detailed above there is no algorithm to solve the TWT problem on real-sized problems. Furthermore it is unknown whether the performance of existing MAC protocols of WSN-s can be improved or not.
In the first thesis, I investigate the optimal resource management and scheduling in the case of uniform load, TT and TWT problem, and in the case of WSNs as well. I introduce methods which solve the problems with near optimal solution. These heuristics provide solutions in polynomial time in order to use in practice, furthermore the algorithms can be executed on many-core architectures.

1.1.3 Detection of outliers and events

Event detection is concerned with the detection of sudden changes in the observed system and giving alarm if such changes occur. In WSN environment not only the precise detection but the energy efficiency is of also importance.
The event detection problem can often be reduced to other data mining tasks such as the task of detection deviating, unusual, outlier values. A real-time executable algorithm with high decision rate especially designed for time series does not exist currently and several researchers are working on such methods.
Based on the information model the existing outlier detection algorithms can be classified as follows: i) Unsupervised methods, where there are no assumption on data; ii) Supervised classification, where a model exists for both regular and outlier values; iii) Semi supervised recognition, where a model exists only for regular measurements. In the case of WSN the unsupervised methods are applicable, because we barely have information about the object have to be measured and a model about the occurring events or erroneous measurements only hardly can be created. There are many parametric, model based methods which can only be used with inaccurate detection rate, and with high detection error probabilities [18], [19], [20]. To replace the classical methods adaptive algorithms have also been developed. Among of the adaptive methods there is a significant amount of approaches based on artificial neural networks, which have good detection rates, can easily be adapted and have relatively low computational complexity. [21], [22], [23].

There are several *SVM based* event detection methods used in WSN. Havinga et al [24], [25] developed the Quarter Sphere SVM method, which uses SVM to classify the measurements of nodes. A quarter-hypersphere is being used in SVM in order to perform classification. Bezdek et al. [26] replaced the quarter-sphere by a hyper-ellipsoid, in order to reach higher detection rate. The *neural network based classifier* is a method with two phases [23], where the two phases are the following: i) each sensor performs locally a classification, which results are sent to the BS where ii) the decisions are fused and classified to detect events. In the case of *voting-based method*s the nodes of WSN build up a decision tree, and the decision of the nodes are processed by BS with a voting system (majority decision, weighted decision). *Rule detection and pattern matching*: the events are a priori defined, or compiled with data mining techniques, and detection algorithm seeks matching measurement to these defined rules. With this method even complex series of events can be detected, however the rules cannot be generated automatically [28]. The pattern matching problem can be solved using fuzzy logics [29], where the probability of the event can also be handled. Using *Feature extraction based method*s the amount of data transmitted through the network can be minimized, by pre-processing them on the sensing node [30]. In that way the energy consumption of the sensor nodes can be significantly decreased.

Many of the previously discussed methods however do not focus on the energy efficiency; furthermore there are only a few of methods which rely on the information of the spatial and temporal correlation between the sensor nodes. Using that information the detection ratio can be improved.

In the second thesis I investigate the monitoring wireless sensor networks. I introduce an algorithm which is able to reach high detection rate and energy efficient by reducing the number of radio transmission in the network. Another advantage of the algorithm is the fast processing capability, which is being solved by executing the algorithm on may-core processing architecture.

1.2 The objectives of the dissertation

Concluding the previous section, the objective of the dissertation is to develop algorithmic and methodic tools, which are able to efficiently detect event and crisis in WSN at high detection rate, in energy efficient way with minimum resources.
The next table summarizes the research areas, the WSN technological and general application areas of my developed methods.

Research area	***WSN application***	***Further application of results***
Resource management, scheduling	MAC protocols, Packet scheduling	Telecommunication, Schedule of computational resources, Finance
Event detection, analysis of time series	Evaluating sensor readings, detecting outlier values and events	Outlier detection in time series; Analysis of spatiotemporal correlated data; Decision systems; Smart Metering and Smart Grid systems; Crowdsourcing

Table 1. Summary of research areas and theses

2. Research methodology

In order to achieve the results presented in the dissertation, the following research methodology have been used:

Modeling: It is essential to model the investigated problem at each thesis. That means i) to apply existing models; ii) and to develop and construct new models and to modify the existing ones to create new better fitting ones.
In my work, the algorithm which solves the packet scheduling for WSN relies on models of the communication channels and models of interference. The topology of the network and the relationship between the nodes are described with tools of graph theory.
During the analysis of the time series I have used the models of random processes and time series.

Combinatorial optimization: My solution for resource scheduling (and packet scheduling) is to transform the original problem into a quadratic optimization problem, which can easily be solved. This method provides the solution in polynomial order of time.

Analytic investigation: Investigating the parameters of the models of the time series I have determined the connection between the model and the optimal free parameters of the developed prediction and clustering based method.

Validation with simulation on synthetic and real data: I have written several software and simulation in Matlab environment in order to validate and test the performance of developed algorithms and methods. I have also compared the result with the result of existing heuristics and solutions. In all cases I took the existing physical and technological models and parameters into consideration, therefore the simulations provide nearly as good testing environment as real implementations and applications. Previous methods are summarized by the following figure:

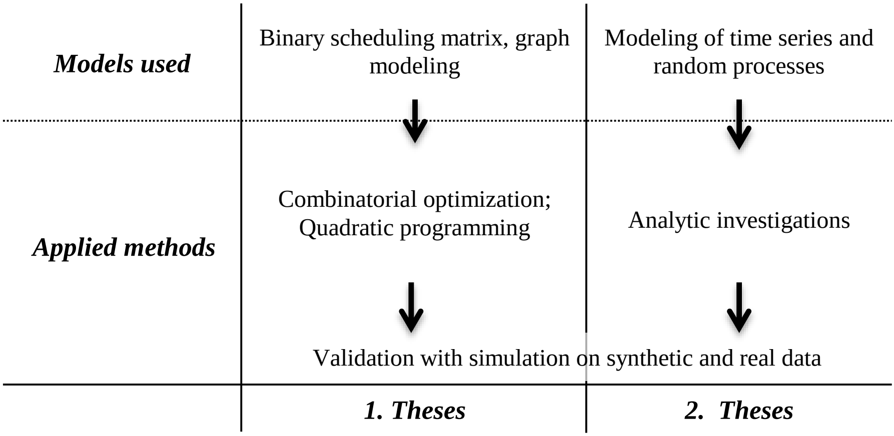

Figure 2. Models and methods used during the research

3. New scientific results

3.1 Scheduling algorithms

1. Thesis. I have developed a new algorithm to solve the resource scheduling problem. This method runs in polynomial order of time, handles the predefined constraints of the task and is able to produce optimal scheduling. The algorithm can be applied to schedule the capacity of a processing unit, where the jobs have priority and deadline. Such task is to schedule the overnight portfolio evaluation and optimization in algorithmic trading systems. I have extended this algorithm to schedule packet transmission in wireless and ad hoc networks, which are used as transmission media for monitoring systems.

(Publications connected to this thesis are: [S1], [S2], [S3], [S5], [S6], [S7].)

Scheduling problems occur in monitoring systems in the following contexts:

- Scheduling the arriving jobs at a processing unit on a node or on the BS. The scheduler has to take the capacity of the processing unit(s) into consideration.
- Packet scheduling in communication systems and the scheduling of the media access.

The model of the problem is as follows: Given J number of jobs, which have to be processed at a processing unit with V capacity. For each j a size and a cutoff time is prescribed. Each job has to be finished before its cutoff time is due. The size of the job is denoted by X_j, the due time is denoted by K_j, where $j = 1 \ldots J$. Each value is discrete integer, therefore the schedule of a single j job can be written as follows:

$$\mathbf{c}_j \in \{0,1\}^L , \tag{1.1}$$

where L denotes the total length of the schedule.
Using the schedule vector a $\mathbf{C}$ binary matrix can be composed with size $J \times L$. This matrix represents the schedule of all jobs. An example is given in (1.2), where the parameters are the followings: $J = 6$, $X_1 = 5$, $X_2 = 5$, $X_3 = 4$, $X_4 = 3$, $X_5 = 3$ $X_6 = 7$, $K_1 = 6$, $K_2 = 8$, $K_3 = 7$, $K_4 = 6$, $K_5 = 4$ and $K_6 = 8$.

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{1.2}$$

In the previously defined matrix, if $c_{j,i}$ value equals 1, then job j is being processed at time slot i by the processing units.
The schedule matrix is valid only, if the prescribed marginal conditions are satisfied. The conditions are the following: in the jth row of the scheduling matrix the number of ones has to be X_j; and none of the columns in the scheduling matrix contains more than V number of ones. Furthermore the finishing time of each j job is before the predefine K_j value. More formally these entail the following equations:

$$\forall j = 1, \ldots, J : \sum_{l=1}^{L} c_{j,l} = X_j , \tag{1.3}$$

$$\forall l = 1 \ldots L : \sum_{j=1}^{J} c_{j,l} \leq V, \tag{1.4}$$

$$\forall j = 1, \ldots, J : \sum_{l=K_j+1}^{L} c_{j,l} = 0. \tag{1.5}$$

Using the previously defined model several optimization tasks can be defined, where using a specific objective function and constraint functions an optimal scheduling matrix has to be determined. The objective function can be either: (i) uniform load, or (ii) minimizing the preemption, or (iii) minimal tardiness, or (iv) minimal weighted tardiness. In the following discussion I am going to detail the solutions I have developed for these problems.

1.1. I have created polynomial-order heuristic schedulers, which can solve the total weighted tardiness problem. These new algorithms (LWPF and PLWPF) outperform the existing solutions. I have verified the results with simulations and tests.

One variant of the scheduling problem is the so called TWT problem, in which the marginal condition described by (1.5) certainly is not going to be satisfied, because the available processing capacity is not enough. So the incoming jobs cannot be completed in due time. This problem requires the extension of the original model; therefore for each j job we introduce a weight as follows:

$$\mathbf{w} = \{ w_1, w_2, w_3, \ldots, w_J \} \in \mathbb{R}^N, \; w_j \geq 0, \forall j = 1, ..., J \tag{1.6}$$

The objective is to minimize the total weighted tardiness (TWT), therefore the objective function is:

$$\mathbf{C}_{opt} := \arg\min_{\mathbf{C}} \sum_{j=1}^{J} w_j T_j \; , \tag{1.7}$$

where $T_j = \sum_{l=K_j}^{L} c_{j,l}$.

In the case of Largest Weighted Process First (LWPF) algorithm, the jobs are ordered according to theirs weight. The largest weighted process is the first and so on. The jobs are scheduled to the first possible position, where the capacity constraint is not exceeded. This method provides significantly better solution than the EDD and WSPT heuristics.

The perturbed LWPF (PLWPF) method is the improvement of previous heuristics. The results of LWPF are not as good as the results of HNN regarding to the TWT. PLWPF method is analogous to the PSHNN method. This heuristics generates a scheduling matrix, which is going to be randomly modified then. This perturbation does not violate the capacity constraints; however the predefined constraints on job sizes might be violated. The matrix must be corrected; the correction algorithm is depicted by Figure 3.

Choosing the best solution of several randomly generated and corrected scheduling matrix, we can get such a solution which is nearly as good as the solution provided by the quadratic optimization. However this method is inflexible against the altering restrictions.

REQUIRE $\mathbf{x}, \mathbf{w}, \mathbf{K}, V, \mathbf{C}$

FOR $l = 1 \rightarrow L$

  WHILE $\sum_{j=1}^{J} c_{j,l} > V$

    Remove 1 from row i, where i is the row in column l with minimal weight that has $C_{j,k}$=1

  END WHILE

END FOR

FOR $k = 1 \rightarrow J$

  WHILE $\sum_{k=1}^{L} c_{l,k} > \mathbf{X}_l$

    Remove 1 from row l from the column i, where i is the righternmost column in row l such that $C_{k,l}$=1

  END WHILE

  WHILE $\sum_{k=1}^{L} c_{l,k} < \mathbf{X}_l$

    Add 1 to row l in column i, where i is the lefternmost column where 1 can be added without violating the capacity constraint V

  END WHILE

END FOR

RETURN C

Figure 3. Correction algorithm

1.2. I have transformed the scheduling problem into a quadratic optimization problem based on the previously defined model. My solution provides uniform load at the resource. Furthermore, it takes the predefined job sizes and the time constraints into consideration too.

The objective function of the task, where the processing units have to be loaded uniformly, can be written as follows:

$$\mathbf{C}_{opt} : \min_{\mathbf{C}} \sum_{l=1}^{L}\sum_{k=1}^{L}\left(\sum_{j=1}^{J} c_{j,l} - \sum_{j=1}^{J} c_{j,k}\right)^2, \tag{1.8}$$

namely, the difference between the weights of two arbitrary chosen column of the scheduling matrix is minimal.

The matrix $\mathbf{C}$ is transformed into a vector row-wise, then the total objective function (supplemented by constraint functions)

$$\min E(\mathbf{C}) = \sum_{j=1}^{J}\left(\sum_{l=1}^{L} c_{j,l} - X_j\right)^2 + \sum_{j=1}^{J}\sum_{l=K_j+1}^{L} c_{j,l}^2 + \sum_{l=1}^{L}\sum_{k=1}^{L}\left(\sum_{j=1}^{J} c_{j,l} - \sum_{j=1}^{J} c_{j,k}\right)^2 \tag{1.9}$$

is transformed such that the minimum value of the following quadratic form equals with the solution to previous objective function:

$$\frac{1}{2}\mathbf{c}^T\mathbf{W}\mathbf{c} + \mathbf{b}^T\mathbf{c}. \tag{1.10}$$

In order to perform the transformation following equations have to be solved. The first is the constraints on the job sizes:

$$\sum_{j=1}^{J}\left(\sum_{l=1}^{L} c_{j,l} - X_j\right)^2 = \frac{1}{2}\mathbf{c}^T\mathbf{W}_A\mathbf{c} + \mathbf{b}_A^T\mathbf{c}. \tag{1.11}$$

Second, third equation defines the time constraints:

$$\sum_{j=1}^{J}\sum_{l=K_j+1}^{L} c_{j,l}^2 = \frac{1}{2}\mathbf{c}^T\mathbf{W}_C\mathbf{c} + \mathbf{b}_C^T\mathbf{c}. \tag{1.12}$$

The third equation is the objective function:

$$\sum_{l=1}^{L}\sum_{k=1}^{L}\left(\sum_{j=1}^{J} c_{j,l} - \sum_{j=1}^{J} c_{j,k}\right)^2 = \frac{1}{2}\mathbf{c}^T\mathbf{W}_D\mathbf{c} + \mathbf{b}_D^T\mathbf{c}. \tag{1.13}$$

After these equations are solved, the following $\mathbf{W}$ matrices and $\mathbf{b}$ vectors can be written. (The third equation has been split into two parts.)

$$\mathbf{b}_A = 2\begin{pmatrix}\mathbf{d}_{A_1} & \mathbf{d}_{A_2} & \cdots & \mathbf{d}_{A_J}\end{pmatrix},\ \mathbf{d}_{A_j} = X_j \cdot \begin{pmatrix}\mathbf{1}_{1\times K_j} & \mathbf{0}_{1\times L-K_j}\end{pmatrix}, \tag{1.14}$$

$$\mathbf{W}_A = 2\begin{pmatrix} \mathbf{1}_{K_1\times K_1} & \mathbf{0}_{K_1\times L-K_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{L-K_1\times K_1} & \mathbf{0}_{L-K_1\times L-K_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{K_2\times K_2} & \mathbf{0}_{K_2\times L-K_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{L-K_2\times K_2} & \mathbf{0}_{L-K_2\times L-K_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{K_J\times K_J} & \mathbf{0}_{K_J\times L-K_J} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}_{L-K_J\times K_J} & \mathbf{0}_{L-K_J\times L-K_J} \end{pmatrix}$$

$$\mathbf{W}_C = 2\begin{pmatrix} \mathbf{D}_{C_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{C_2} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_{C_J} \end{pmatrix} \text{ and } \mathbf{b}_C = \mathbf{0}_{JL\times 1}, \tag{1.15}$$

where $\mathbf{D}_{C_j} = \begin{pmatrix} \mathbf{0}_{K_j\times K_j} & \mathbf{0}_{K_j\times(L-K_j)} \\ \mathbf{0}_{(L-K_j)\times K_j} & \mathbf{I}_{(L-K_j)\times(L-K_j)} \end{pmatrix}$.

$$\mathbf{W}_{D_1} = 2L\begin{pmatrix} \mathbf{I}_{L\times L} & \mathbf{I}_{L\times L} & \cdots & \mathbf{I}_{L\times L} \\ \mathbf{I}_{L\times L} & \mathbf{I}_{L\times L} & \cdots & \mathbf{I}_{L\times L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{L\times L} & \mathbf{I}_{L\times L} & \cdots & \mathbf{I}_{L\times L} \end{pmatrix}, \text{ and } \mathbf{b}_{D_1} = \mathbf{0}_{JL\times 1}, \tag{1.16}$$

$$\mathbf{W}_{D_2} = 2\mathbf{1}_{JL\times JL} \text{ and } \mathbf{b}_{D_2} = \mathbf{0}_{JL\times 1}. \tag{1.17}$$

The parameters of the quadratic form can be computed by summing up the previous matrices and vectors as follows:

$$\begin{aligned} \mathbf{b}_{BC} &= \alpha\mathbf{b}_A + \gamma\mathbf{b}_C + \delta_1\mathbf{b}_{D_1} + \delta_2\mathbf{b}_{D_2}, \\ \mathbf{W}_{BC} &= \alpha\mathbf{W}_A + \gamma\mathbf{W}_C + \delta_1\mathbf{W}_{D_1} + \delta_2\mathbf{W}_{D_2}. \end{aligned} \tag{1.18}$$

Each constraint can be priorized by the α, γ, δ_1, δ_2 heuristic parameters. This method can be generalized by changing the constraint. In this way other, similar scheduling problems can be solved, as follows.

1.3. Based on previous results, I have created new packet scheduling protocols for wireless sensor networks. The extension of these solutions can cope with further constraints arising in WSN environment. Therefore the protocol is able to provide valid packet scheduling matrices for complex WSNs.

I have demonstrated that resource scheduling method introduce in section 1.2. is adaptable to use in wireless sensor networks. I have investigated and adapted the method in two applications, as follows:

1) The physical model of the first, simplified implementation is the following: the nodes of the network are arranged into clusters, and each node communicates with base station via its cluster head node. This layout is illustrated by Figure 4.

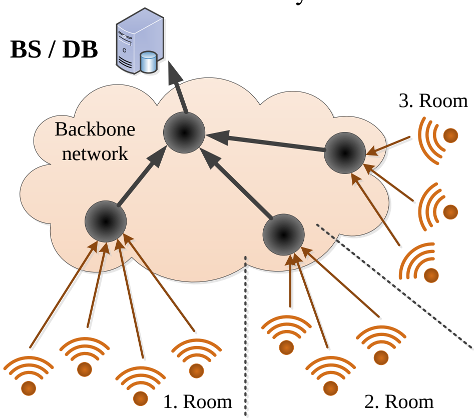

Figure 4. Layout of WSN with cluster heads

In this case the task is to schedule the packets arriving to the cluster heads in order to determine which packets are sent simultaneously to the BS in one timeframe. The number of packets which have to be sent to the BS from one source is known (X_j). The transmissions also have due time, so all transmissions have to be finished before the predefined time is over (K_j). In this model it is supposed that the available capacity is not enough to transmit all packet in due time. Furthermore, the time constraint is more important than the reliability of communication, because the source nodes have limited energies (after K_j time its battery discharges). The objective is to provide uniform packet loss probability:

$$P\left(\text{packet loss at node } i\text{, at timeslot } l\right) = \frac{\sum_{j=1}^{J} c_{j,l} - V}{\sum_{j=1}^{J} c_{j,l}}. \qquad (1.19)$$

This problem can then be easily transformed into the previously introduced scheduling problem.

2) The packet scheduling of an entire WSN has to be determined in the second, extended application. We have the following constraints to guarantee the validity of packet scheduling:
   a. The amount of packets for each leaf node is predefined.
   b. The data traffic at an intermediate node is also defined; the precedencies of the transmissions have to be maintained.
   c. The interference has to be strictly avoided in order to guarantee the arrival of the packets.

   Based on the previous constraints a quadratic optimization task can be constructed, that solves the original problem with the predefined parameters. In order to provide a valid packet scheduling, the weight matrices and vectors of the quadratic optimization are the following: penalty to avoid interference $\mathbf{W}_F$, $\mathbf{b}_F$; matrices and vectors ($\mathbf{W'}_A$, $\mathbf{b'}_A$, $\mathbf{W}_E$, $\mathbf{b}_E$) to ensure the correct traffic in the network; $\mathbf{W}_G$, $\mathbf{b}_G$ weight matrix and vector to avoid unnecessary wake-up of nodes. The objective can be arbitrary, the corresponding weight are denoted by $\mathbf{W}_H$ and $\mathbf{b}_H$. Using previous matrices and vectors the ultimate optimization task is $\mathbf{W} = \alpha'\mathbf{W'}_A + \varepsilon\mathbf{W}_E + \xi\mathbf{W}_F + \zeta\mathbf{W}_G + \mathbf{W}_H$ and $\mathbf{b} = \alpha\mathbf{b}_A + \varepsilon\mathbf{b}_E + \xi\mathbf{b}_F + \zeta\mathbf{b}_G + \mathbf{b}_H$, where α, ε, ξ, ζ parameters are heuristic values in order to prioritize the penalty matrices. I have determined the weight matrices and vectors for traffic coordination as follows:

$$\mathbf{W}'_A = 2\begin{pmatrix} \mathbf{I}_{J\times J} & \mathbf{I}_{J\times J} & \cdots & \mathbf{I}_{J\times J} \\ \mathbf{I}_{J\times J} & \mathbf{I}_{J\times J} & \cdots & \mathbf{I}_{J\times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{J\times J} & \mathbf{I}_{J\times J} & \cdots & \mathbf{I}_{J\times J} \end{pmatrix}, \mathbf{b}'_A = (\mathbf{z}, \mathbf{z}, \ldots, \mathbf{z})_{JL\times 1}. \tag{1.20}$$

$$\mathbf{W}_E = 2\begin{pmatrix} \mathbf{P}_{J\times J} & \mathbf{P}_{J\times J} & \cdots & \mathbf{P}_{J\times J} \\ \mathbf{P}_{J\times J} & \mathbf{P}_{J\times J} & \cdots & \mathbf{P}_{J\times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{J\times J} & \mathbf{P}_{J\times J} & \cdots & \mathbf{P}_{J\times J} \end{pmatrix}, \mathbf{b}_E = \mathbf{0}_{JL\times 1}, \tag{1.21}$$

   where $\mathbf{P} = \mathbf{D}_E + \mathbf{I}_{J\times J} - 2\mathbf{R}_{J\times J}$ and

$$\mathbf{D}_E = \begin{pmatrix} \mathbf{R}_{1,1}^2 + \mathbf{R}_{1,2}^2 + \ldots + \mathbf{R}_{1,N}^2 & \mathbf{R}_{1,1}\mathbf{R}_{2,1} + \ldots + \mathbf{R}_{1,N}\mathbf{R}_{2,N} & \cdots & \mathbf{R}_{1,1}\mathbf{R}_{N,1} + \ldots + \mathbf{R}_{1,N}\mathbf{R}_{N,N} \\ \mathbf{R}_{1,1}\mathbf{R}_{2,1} + \mathbf{R}_{1,2}\mathbf{R}_{2,2} + \ldots + \mathbf{R}_{1,N}\mathbf{R}_{2,N} & \mathbf{R}_{2,1}^2 + \mathbf{R}_{2,2}^2 + \ldots + \mathbf{R}_{2,N}^2 & \cdots & \mathbf{R}_{2,1}\mathbf{R}_{N,1} + \ldots + \mathbf{R}_{2,N}\mathbf{R}_{N,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{1,1}\mathbf{R}_{N,1} + \mathbf{R}_{1,2}\mathbf{R}_{N,2} + \ldots + \mathbf{R}_{1,N}\mathbf{R}_{N,N} & \mathbf{R}_{2,1}\mathbf{R}_{N,1} + \mathbf{R}_{2,2}\mathbf{R}_{N,2} + \ldots + \mathbf{R}_{2,N}\mathbf{R}_{N,N} & \cdots & \mathbf{R}_{N,1}^2 + \mathbf{R}_{N,2}^2 + \ldots + \mathbf{R}_{N,N}^2 \end{pmatrix}$$

1.4. I have applied a Hopfield neural network to solve the quadratic optimization tasks. I have introduced a procedure to improve the results provided by the HNN solver. This procedure determines good starting points for the HNN recursion and adjusts the heuristic parameters too.

I have used the following recursive state transition rule to solve the previously defined quadratic optimization problems. This rule is used in Hopfield neural networks, and solves the problem in polynomial time order.

$$\mathbf{y}_i(k+1) = \mathrm{sgn}\left(\sum_{j=1}^{N} \hat{W}_{ij} y_j(k) - \hat{b}_i\right), i = \mathrm{mod}_N k. \tag{1.22}$$

It has been proven that this method minimizes the following quadratic function in *polynomial order* of time:

$$\mathrm{f}(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T \mathbf{W} \mathbf{y} + \mathbf{b}^T \mathbf{y}. \tag{1.23}$$

This HNN recursion was repeated several times for each problem with different random starting points and the best solution was used as the solution of the method. This type of execution has a great advantage against other heuristic solvers the capability of parallel execution and implementation. This capability may speed up the time of execution extremely.

I have introduced an improved version of this method, the SHNN method. SHNN method can only be used when other fast heuristics are available to solve the problem, because the solution of the common heuristic will be the initial state of the SHNN method. Other improved version is the PSHNN method, in which the inputs of the HNN recursion are generated with perturbation from the solution of common heuristics. These improvements are needed because the HNN recursion in practical implementations can get stuck in local extreme points, which are not the solution of the optimization task.

REQUIRE $\mathbf{X}, \mathbf{K}, V, e$

$\alpha \leftarrow 0.1, \beta \leftarrow 5, \gamma \leftarrow 5$
$i \leftarrow 0$
REPEAT
  $i \leftarrow i+1$
  $\mathbf{C}_i \leftarrow \mathrm{HNN}(\mathbf{X}, \mathbf{K}, V, \alpha, \beta, \gamma)$
  $\alpha \leftarrow \alpha + 0.01$
UNTIL $\mathrm{error}(\mathbf{C}_i) \le e$
FOR $k = 1 \to i$
  $\mathbf{C}_k \leftarrow \mathrm{correct}(\mathbf{C}_k)$
  $\mathbf{T}_k \leftarrow \mathrm{calculateTWT}(\mathbf{C}_k)$
END FOR

RETURN $\min(\mathbf{T})$

Figure 5. Algorithm for adjusting the heuristic parameter of HNN method

Another issue of HNN is that no analytic method exists to determine the optimal heuristic parameters. In my implementation I have used the algorithm depicted by Figure 5., which adjusts the heuristic parameters during the execution of the recursion rule. The results for TWT problem provided by HNN, SHNN and PSHNN methods are depicted by Figure 7.

1.5. In order to improve the runtime of the HNN/PSHNN iterations I have designed a method to execute the iterations on many-core computing architecture. This method is capable to determine the scheduling matrix up to 400 percent faster, than the common HNN method.

The HNN state transition rule introduced in section I.3 is one of several heuristic solves which is capable of solving the quadratic optimization in polynomial order of time. The HNN iteration is repeated from several initial points and with several heuristic parameters. These repetitions enable us to get an acceptable solution. As we have seen that these repetitions mean two levels of re-iterations. Theoretically, if the runtime of one iteration of a HNN recursion is t, then the total, non-parallelized runtime is:

$$T = R_{init}\left(R_{heuristics}t\right), \tag{1.24}$$

If $R_{init} = 500; R_{heuristics} \sim 10^2; t = 10^{-3}s$ then the estimated time of execution is T~50 seconds, using a 200×200 sized **W** matrix.

Each iteration is independent from all other iterations; therefore it can be executed independently and in parallel. Using an existing parallel HNN implementation [31], we extend the parallelism on nVidia Fermi architecture as follows [32]:

- The **W** matrix is sparse (5-10 percent of values are nonzero, and there are patterns amongst the nonzero values), therefore in the case of large **W** matrix one HNN iteration can be executed by a stream multiprocessor (SM) using its local (shared) memory.
- The independent HNN iterations (different starting points and different heuristic parameters) can be executed on separate SM-s.

Therefore the PSHNN algorithm can be adapted to be executed on Fermi many-core architecture. So, the execution time can extremely be decreased using **W** matrix with proper size. The 400% speedup written above in the statement can be measured between the multicore nVidia 440GTX and single core Intel Core i5. This algorithm provides better scheduling than the existing common heuristic methods; and this scheduling can be determined as fast as it is acceptable in practical applications [32], [33].

Performance analysis for the methods introduced in the first Thesis

The tests of algorithms developed to solve the TWT problem illustrates the relation of performance between the new (LWPF, PLWPF, HNN, SHNN, PSHNN) and existing methods. The results are depicted on Figure 6 and Figure 7.

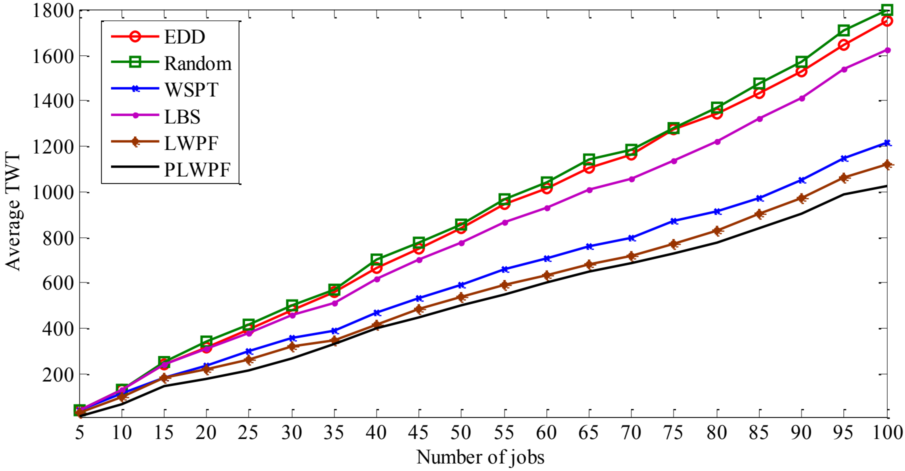

Figure 6. The solution of HNN method compared to other heuristic algorithms (the smaller value is the better)

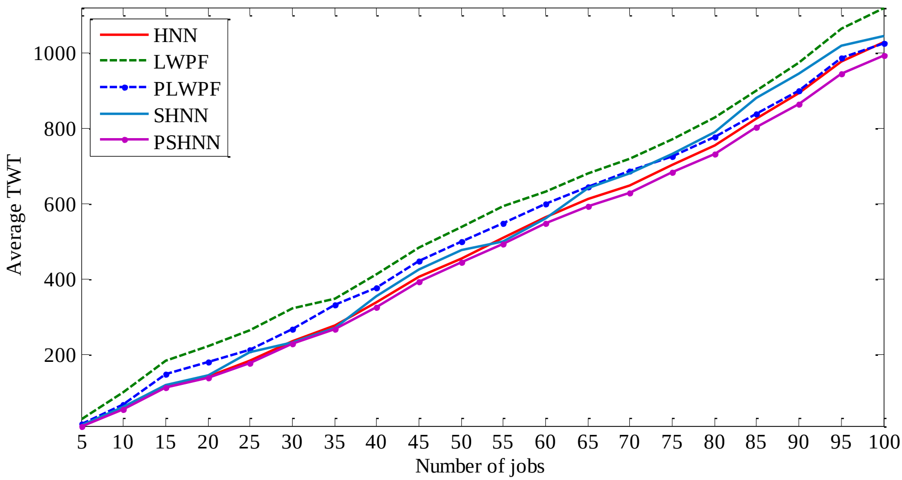

Figure 7. Comparison of the results of improved HNN methods and the results of improved heuristic solutions

The best solution is provided by the HNN based methods; however the execution time is longer than the other common heuristics and LWPF. The execution time of the parallelized HNN method is better than the original HNN method, which is illustrated by Figure 8.

I have developed the parallel version of the scheduling method, which is capable of being executed on GPGPU architecture. The runtimes are compared on the next figure.

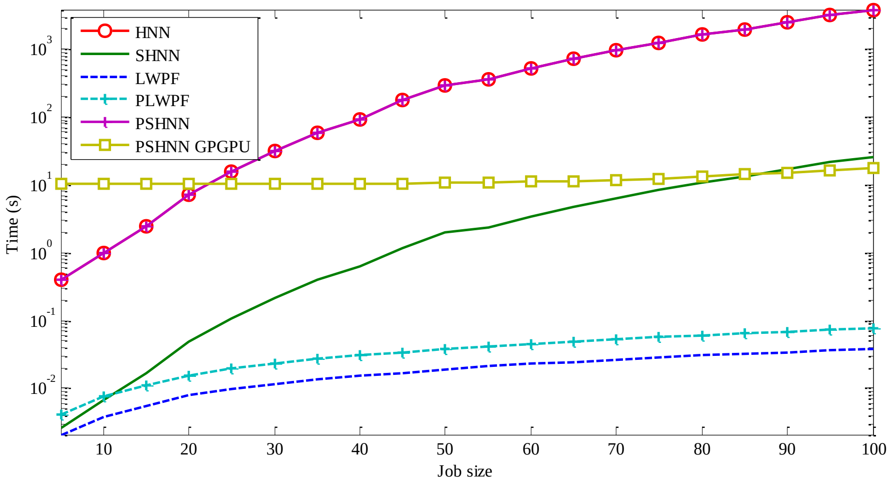

Figure 8. The runtime of the method using parallel implementation of the HNN method compared to the runtime of other methods

Summarizing the previous results, the conclusion is that I have made such algorithms which are capable of providing near optimal solutions for constrained scheduling tasks, with acceptable runtime.

3.2 Outlier- and event detection

2. Thesis. I have developed new methods which detects outlier values and events in time series data at high detection rate. The developed method reduces the number of radio transmissions in the network, therefore it is energy efficient. Furthermore the algorithm is executed in distributed manner; the local detection is performed on the nodes of the WSN. In that way it can be used in WSN applications.

(Publication connected to this thesis: [S3].)

WSNs are often deployed in monitoring or surveillance applications. In these applications a frequent task is to detect the events occurring at the observed area. There are several existing algorithms to solve the event detection problem; however these algorithms do not focus on energy efficiency.
The new method which I have developed has two building blocks, which blocks provide the capability to execute the algorithm in a distributed way on the sensor nodes of the network. Also with the reduction of the radio transmission in the network the lifespan of the network is extended. These components are detailed in the following sections. *(The notation, models, functions are fully explained in the dissertation in chapter three.)*

2.1. I have used a prediction based algorithm to detect anomalies in sensor time series. This method compares the actual measurement and the value predicted from historical data in order to reach decision. I have analytically determined the threshold value of optimal decision. Furthermore the efficiency of the scheme has been verified by extensive simulations.
The basic idea of the method is to exploit the information of temporal correlation of the values in time series. The operation of the method is illustrated by Figure 9.

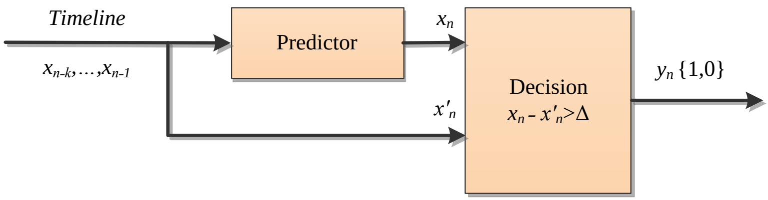

Figure 9. Prediction based outlier detection

The next value of a time series (x_n) can be predicted (x'_n) by using the historical values of the times series ($x_{n-k} \dots x_{n-1}$). The predicted value and the real measurement are compared to each other and a binary decision is made ($y=1$, if $|x_n' - x_n| > \Delta$).Using the model of the sensed time series and the model of the outlier process the optimal decision threshold value (Δ) can be determined. This value is optimal when the detector reports outlier only if an outlier value occurred, otherwise it should report the value as a normal measurement. The optimal decision parameters can be determined by minimizing the probability of the erroneous decision:

$$\Delta_{opt} : \min_{\Delta} P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \middle| \alpha_k = 0\right), \tag{2.1}$$

where $\hat{\xi}, \eta, \alpha$ are the observed and predicted processes, and the value which indicates the presence of outliers. The following constrain have to be considered:

$$P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \middle| \alpha_k = 1\right) = \varepsilon \ , \tag{2.2}$$

where the probabilities (substituting the parameters and values):

$$P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \middle| \alpha_k = 1\right) = \sum_{l=1}^{L-1} \iint_{y,z} \left\{ \Phi\left(\frac{\Delta - y + z}{\sigma}\right) - \Phi\left(\frac{-\Delta - y + z}{\sigma}\right) \right\} f(y) g^{(l)}(z) \mathbf{d}y \mathbf{d}z \binom{L}{l} p^l (1-p)^{L-l} \tag{2.3}$$

and

$$P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \middle| \alpha_k = 0\right) = \\ = \sum_{l=1}^{L-1} \int_{z} \left\{ \Phi\left(\frac{-\Delta - z}{\sigma}\right) + \Phi\left(\frac{-\Delta + z}{\sigma}\right) \right\} g^{(l)}(z) \mathbf{d}z \binom{L}{l} p^l (1-p)^{L-l} . \tag{2.4}$$

I have used linear predictor to predict the forthcoming values such timelines where the time series is generated by a linear autoregressive process. In more general case instead of linear predictor a feed forward neural network can be used as a predictor.

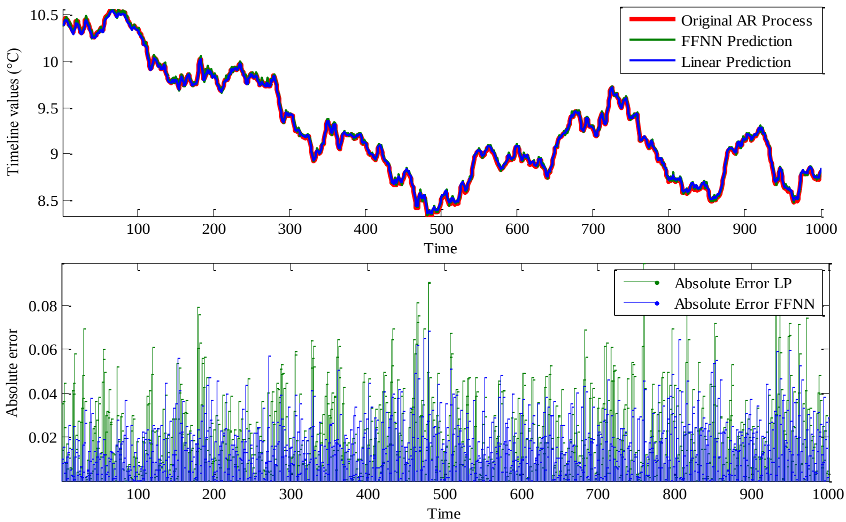

Figure 10. Comparison of linear and non-linear numerical predictor

The detection capability of these two predictors is compared on Figure 10. The cumulated absolute error of the FFNN is lower than of the LP, because of the nonlinearity of the generated process. Using a proper decision threshold values nearly one hundred percent of outlier values can be detected. This partly distributed outlier detection scheme can be applied in systems, where the deployed units have only limited computation power; however the local problems must be detected. (For example: in environment which is observed by WSN, or in the smart meters of Smart Grids.)

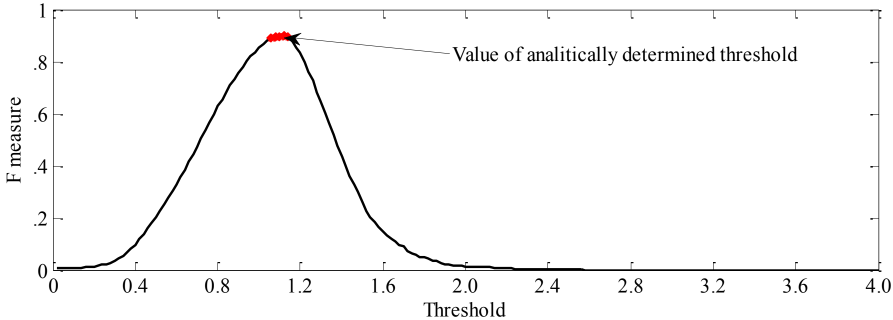

Figure 11. The performance of analytically obtained threshold value is maximal

The threshold we obtain by previous analytic method is compared to empirical results. The simulation demonstrated that the analytically determined threshold parameter is optimal indeed.

2.2. Using the prediction based outlier component I have designed a new method to detect events in wireless sensor networks with low probability of detection error. This new method consumes minimal energy extending the lifespan of the network. The algorithm is executed in semi-distributed fashion on the nodes of the sensor network. Also I have introduced an algorithm to determine the clusters for optimal decision.

Using the results of the outlier detection module I have designed an event detection method which exploits the information of spatial correlation of the sensor measurements. The block diagram of this method is illustrated by Figure 12.

At phase one the outlier detection is performed on the nodes of the network, and the results of the detection are sent to the base station. At this second phase on base station the local decisions are processed by a clustering algorithm. Using the clustering algorithm the i) local measurements errors and local sensor errors; ii) events in a locality (neighborhood of nodes) can be detected. The radius of the locality can also be taken into consideration.

We seek the optimal cluster belonging to the correct decision. In order to determine that cluster, the probability of correct event detection has to be maximized:

$$\max_{\mathbb{C}} \sum_{\substack{\mathbf{y}\in\{0,1\}^{\mathbb{Q}} \\ w(\mathbf{y})\geq\left\lceil \frac{L}{2} \right\rceil}} \sum_{i=1}^{L} y_i P_i, \tag{2.5}$$

where y denotes the decision of the prediction based method, and L denotes the size of the cluster. Therefore such a cluster is needed where in at least the half of nodes have reported outliers in case of an event.

The parameters needed to maximized previous form can be obtained in real WSN. This optimization has to be repeated for each node to obtain all clusters.

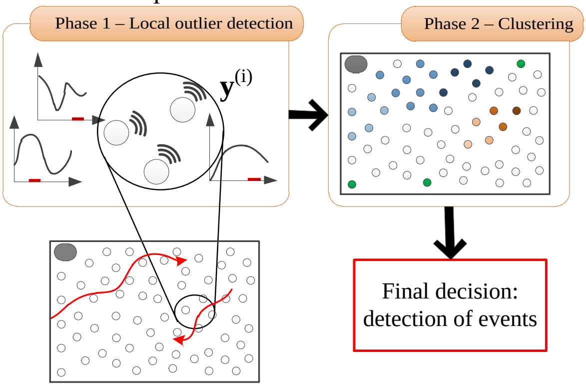

Figure 12. Event detection method based on outlier detection

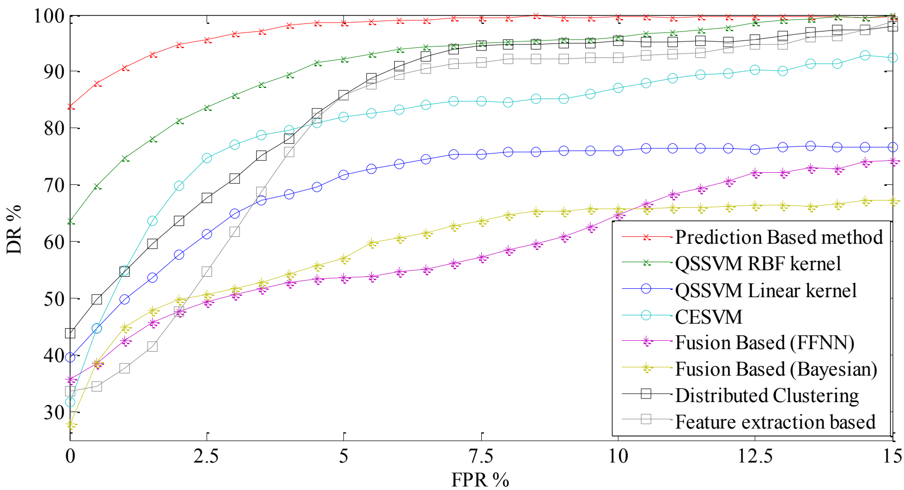

Figure 13. Comparison of the developed event detection method

The results of the developed method are illustrated by Figure 13. On the figure the detection rate of this new and the existing [24]-[30] methods are depicted in function of the false positive detection rate (ROC curve). One may see that the new method outperforms the existing ones.

2.3. I have developed the parallel implementations of the event detection algorithm. This parallel implementation can be executed on many core (i.e. GPGPU) architectures in order to decrease the runtime of the method. The rate of speed improvement achieved by parallel execution is proportional to the number of nodes.

The main phases of the developed event detection method are summarized by Figure 12. The first phase, the prediction based outlier detection is executed in distributed fashion on the nodes of the sensor network. The second phase the clustering based event detection is executed on the BS. However the evaluations of information of different nodes are independent processes so it can be parallel executed.

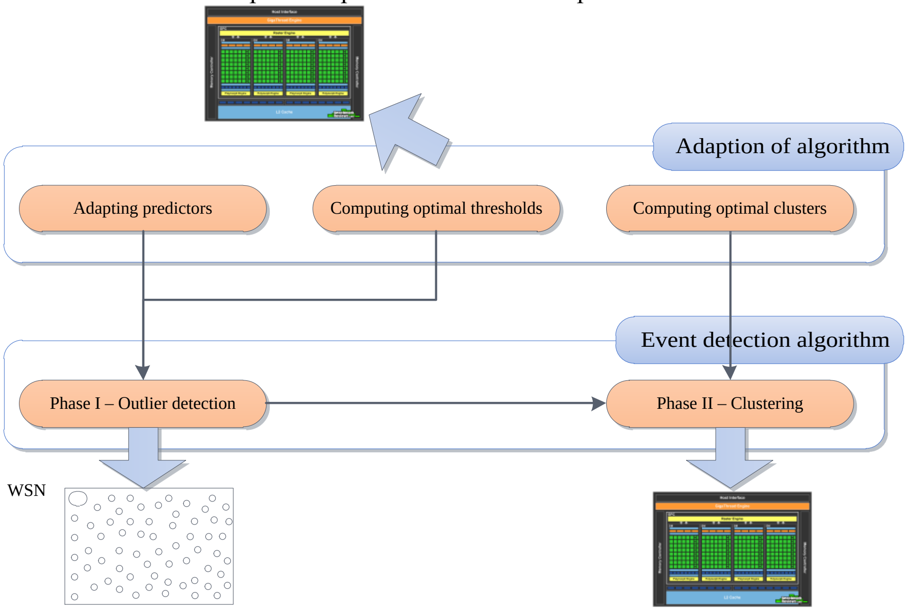

Figure 14. Summary of the parallel implementation of the method. The outlier detection in phase one is executed in distributed way, while the other calculation can be implemented on GPGPU where the tasks can be executed independently, simultaneously.

The training and adaption process of the algorithm has the following parts, which can be parallel executed:

- *Adaption of predictors*: For each node a node-specific training set exists, however the training algorithm is common. Using these training sets, the adaptions can be executed independently and separately.
- *Determining the optimal threshold parameters of prediction based detection:* Each threshold parameter can be determined independently and parallel using node-specific data.
- *Determining the optimal clusters for each node*: The optimal cluster corresponding to one node can be determined generating a simulated event. In order to determine the clusters, the input data and the position of the node is needed. Having this information the cluster can be detected on GPGPU architecture in a parallel fashion.

The computational pattern of this distributed method can be used in other applications such as crowdsourcing.

4. Conclusion of theses and application of the results

I have introduced such new methods and algorithms in both theses which

- performances are better than the existing solution and algorithms;
- computational complexities are low, can be evaluated in polynomial order of time;
- can be executed parallel on many-core architectures in order to decrease significantly the runtime;
- are able to find near optimal solution in the case of new constrains are included to the input parameters.

My developed new methods may be used in monitoring WSN applications. Accomplishing previous requirements I have achieved the objectives of the dissertation.

The results can be used in monitoring systems such as:
Monitoring in healthcare, in environment, in where the area/object is monitored. The aim of monitoring is to collect data or predict behavior. There are several sensors to sense climatic, geological or other environmental properties.

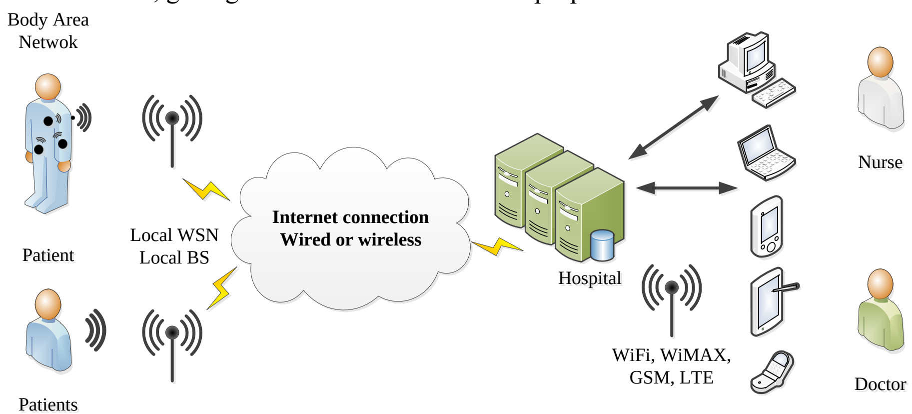

Figure 15. Intelligent monitoring of patients

Monitoring industrial equipment, similar to the environment monitoring the properties and state of the equipment is being monitored.

Observing an area, where the occurring phenomenon is observed at an area or unusual events (such as intrusion) have to be detected.

In previously detailed applications the results of my theses can be deployed especially in monitoring application where the following requirements exist:

- The proper schedule of resources in order to fully utilize the finite capacities of the sensor network;

- The schedule of the packed transmission in order to achieve high throughput and energy efficient operation;
- Event detection algorithm with high detection rate and energy effectiveness in order to detect the unusual phenomenon, events, intrusions.

The applications of the theses are summarized by Table 2.

<table>
<tr><th rowspan="2">Research area</th><th rowspan="2" colspan="2">Performance and its characteristic value</th><th colspan="2">Application of the results</th></tr>
<tr><th>Wireless Sensor Networks</th><th>Other areas</th></tr>
<tr><td>Resource scheduling, optimal channel access</td><td>Latency, total cost of the schedule</td><td>5 % - 10 %</td><td rowspan="2">applications which need energy awareness: alerting, localization, monitoring, controlling</td><td>call admission control, processor scheduling. (For example: in mobile devices, or during data processing.)</td></tr>
<tr><td>Time series analysis, data mining</td><td>Detection rate energy consumption</td><td>10 % -20 %</td><td>data mining: data cleansing, crisis detection; detection on finance time series; crowdsourcing applications; Smart Grid systems (on smart meters.)</td></tr>
</table>

Table 2. Summary of my theses

The theses provide solution for problems existing in other area of sciences where optimal scheduling, outlier detection and parallelism of HNN algorithm are required.

5. Acknowledgements

I would like render thanks to my scientific advisor, János Levendovszky for the support, constructive criticism he gave to improve the level of my dissertation. I express my thanks to Professor Tamás Roska, who is the former leader and to Professor Péter Szolgay, who is the actual leader of the Doctoral School. They have ensured the background for my research. I am grateful to Nyékyné dr. Gaizler Judit, the former Dean of the Faculty who was always ready to discuss my actual problems. My co-authors provided valuable assistance during the research work and preparation of research papers. I would like to say thank for this cooperation to them.

I also render thanks to András Oláh and to all the members of the WSN research group and the fellow PhD students for useful advices, discussions and sympathizing talks, furthermore for the solutions provided in connection with the emerging problems.

In order to create the dissertation the Hungarian and English grammar lectors provided significant assistance, for what I say thank now.

I am especially thankful to my wife, Gabica. She supported me, and sustained me on long nights.

Last but not least, I am grateful to my family and to my friends for their love and support.

6. Publications of the author

In journal

[S1] G. Treplán, **K. Tornai**, J. Levendovszky: "Quadratic Programming for TDMA Scheduling in Wireless Sensor Networks"; *Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks* (Sensor Networks for High-Confidence Cyber-Physical Systems (HCPS)) 2011

[S2] N. Fogarasi, **K. Tornai**, J. Levendovszky, "A Novel Hopfield Neural Network Approach for Minimizing Total Weighted Tardiness of Jobs Scheduled on Identical Machines", *Informatica Acta Universitatis Sapientiae,* **4** (1), 2012, pp. 48-66

[S3] **K. Tornai**, A. Oláh, J. Levendovszky, "Monitoring Algorithm for Intrusion and Danger Detection in Wireless Sensor Networks", reviewed and resubmitted to *Ad hoc & Wireless Sensor Networks,* Old City Publishing, 2012

[S4] **K. Tornai**, N. Fogarasi, J. Levendovszky, "Improvements to the Hopfield Neural Network Solution of the Total Weighted Tardiness Scheduling Problem", in *Electrical Engineering and Computer Science*, Periodica Polytechnica, 57 (**1**), 2013, DOI 10.3311/PPee.2090

On conferences

[S5] J. Levendovszky, E. László, **K. Tornai**, and G. Treplán, "Optimal pricing based resource management", *International Conference on Operations Research Munich 2010,* September 2010, pp. 169

[S6] E. László, **K. Tornai**, G. Treplán, J. Levendovszky, "Novel load balancing scheduling algorithms for Wireless Sensor Networks", *The Fourth International Conference on Communication Theory, Reliability, and Quality of Service*, CTRQ 2011 April 17-22, 2011

[S7] J. Levendovszky, A. Oláh, **K. Tornai**, G. Treplán, "Novel load balancing algorithms ensuring uniform packet loss probabilities for WSN", *2011 IEEE 73rd Vehicular Technology Conference* Budapest, May 15-18, 2011

7. Publications connected to the dissertation

[1] P. Brucker, *Scheduling Algorithms*, 5th edition, New York: Springer, 2007

[2] M. Pinedo, *Scheduling – Theory,* Algorithms and Systems, 3rd edition, New York: Springer, 2008

[3] S. Sahni, "Preemptive scheduling with due dates", *Operations Research*, **27** (5), 1979, pp. 925–934

[4] C. Martel, "Preemptive Scheduling with Release Times, Deadlines, and Due Times", *Journal of the ACM*, **29** (3), 1982, pp. 812–829

[5] E. L. Lawler, "Preemptive scheduling of precedence–constrained jobs on parallel machines, deterministic and stochastic scheduling", *Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems*, 1981, pp. 101–123

[6] J. Du, J. Y. T. Leung, "Minimizing total tardiness on one machine is NP–hard", *Mathematics of Operations Research*, **15** (3), 1990, pp. 483–495

[7] E. L. Lawler, "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness", *Annals of Discrete Mathematics*, 1, 1977, pp. 331–342

[8] M. Azizoglu, O. Kirca, "Tardiness minimization on parallel machines", *International Journal of Production Economy*, 55, 1998, pp. 163–168

[9] R. M. V. Rachamadugu, T. E. Morton, "Myopic heuristics for the weighted tardiness problem on identical parallel machines". Technical report, (CMU–RITR–83–17) Carnegie–Melon University, the Robotics Institute, 1983

[10] D. Biskup, J. Herrman, J. N. D. Gupta, "Scheduling identical parallel machines to minimize total tardiness", *International Journal of Production Economy*, 115, 2008, pp. 134–142

[11] W. Ye, J. Heidemann, and D. Estrin, "An energy–efficient MAC protocol for wireless sensor networks", *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, NY, USA 2002, pp. 1567–1576

[12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, ACM, Baltimore, Md, USA, 2004, pp. 95–107

[13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X–MAC: a short preamble MAC protocol for duty–cycled wireless sensor networks", *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, ACM, New York, NY, USA, 2006, pp. 307–320

[14] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks", *Wireless Networks*, **5** (2), 1999, pp. 81–94

[15] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusent, "TreeMAC: Localized TDMA MAC protocol for realtime high–data–rate sensor networks", *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, 2009, pp. 1–10

[16] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Cross–layer energy and delay optimization in small–scale sensor networks", *IEEE Transactions on Wireless Communications*, **6** (10), 2007, pp. 3688–3699

[17] A. G. Ruzzelli, G. O'Hare, M. O'Grady, and R. Jurdak, "Merlin: cross–layer integration of MAC and routing for low duty–cycle sensor networks", *Ad Hoc Networks*, **6** (8), 2007, pp. 1238–1257

[18] D. M. Hawkins, *Identification of outliers*, Chapman and Hall, 1980

[19] J. Xi, "Outlier detection algorithms in data mining", *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application*, 2008

[20] Y. Zhang and J. Liu, "An outlier mining algorithm based on probability", *Power Electronics and Intelligent Transportation System (PEITS)*, 2009 2nd International Conference on, 2009, pp. 209–212

[21] V. Karioti and C. Caroni, *Detecting an outlier in a set of time series*, Chania, 2002

[22] E. M. Jordaan and G. E. Smits, "Robust Outlier Detection using SVM Regression", *Neural Networks*, IEEE International Joint Conference on, 3, 2004

[23] M. Bahrepour, N. Meratnia, and P. Havinga, "Sensor fusion–based event detection in wireless sensor networks", *Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous*, 2009. MobiQuitous '09. 6th Annual International, 2009, pp. 1–8

[24] Y. Zhang, N. Meratnia, P. Havinga, "Adaptive and Online One–Class Support Vector Machine–based Outlier Detection Techniques for Wireless Sensor Networks", 2009 *International Conference on Advanced Information Networking and Applications Workshops*, IEEE, CIS, 2009

[25] M. Bahrepour, N. Meratnia, P.J. Havinga, "An Online Outlier Detection Technique for Wireless Sensor Networks using Unsupervised Quarter–Sphere Support Vector Machine", *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2008

[26] S. Rajasegarar, C. Leckie, M. Palaniswami, J. C. Bezdek "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks", *ICC 2007 proceedings*, IEEE, 2007

[27] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, P.J.M. Havinga, "Disaster Management", 2010 *International Conference on Intelligent Networking and Collaborative Systems*, 2010

[28] G. Amato, S. Chessa, C. Gennaro, C. Vairo, "Efficient Detection of Composite Events in Wireless Sensor Networks: Design and Evaluation", *2011 Computers and Communications (ISCC),* IEEE Symposium on, 2011

[29] Q. Liang, L. Wang, "Event Detection in Wireless Sensor Networks Using Fuzzy Logic System", *CIHSPS 2005 – IEEE International Conference on computational Intelligence for Homeland Security and Personal Safety*, 2005

[30] N. Dziengel, G. Wittenburg, J. Schiller, "Towards Distributed Event Detection in Wireless Sensor Networks", *DCOSS '08*, 2008

[31] L. Liang, "Parallel implementations of Hopfield neural networks on GPU", Technical Report, Dépót Universitaire de Mémoires Apres Soutenance

[32] R. Farber, *CUDA Application design and development*, Morgan Kaufmann 1st edition, 2009

[33] D.B. Kirk, W.W. Hwu, *Programming massively parallel processors: A hands–on approach*, Morgan Kaufman, 2010