Distributed Algorithm for Energy Optimal Multi-Commodity Network Flow Routing in Sensor Networks.

Jiří Trdlička
Czech Technical University
Faculty of Electrical Engineering
Department of Control Engineering
Technicka 2, 166 27 Prague 6, Czech Republic
Email: trdlij1@fel.cvut.cz

Zdeněk Hanzálek
Czech Technical University
Faculty of Electrical Engineering
Department of Control Engineering
Technicka 2, 166 27 Prague 6, Czech Republic
Email: hanzalek@fel.cvut.cz

***Abstract*—This work proposes a distributed algorithm for the energy optimal routing in wireless sensor network. The routing problem is described as a minimum-cost multi-commodity network flow problem by Linear programming. Based on the convex programming theory we use the dual decomposition theorem to derive the distributed algorithm. The algorithm computes the energy optimal routing in the network without any central node or the knowledge about the whole network structure. Each node only needs to know the flow which is supposed to send or receive and the costs and capacities of the neighboring links.**

I. Introduction

A. Motivation

Our work is focused on a distributed algorithm for data flow routing through the multi-hop static network. An example of a target application is a network periodically sensing some consumption variables (like gas consumption, water consumption, etc.) in large objects. Each sensing device produces a data flow of a particular volume, which is supposed to be routed through the network. The objective is to optimize the energy consumption for the data transfer (minimal possible energy consumption), while constrained by communication capacities for each communication link in the network (maximum data volume which can be transferred trough the link per a time unit).

There are many communication protocols designed for the data routing in wireless sensor networks however, to achieve the energy optimal routing which complies with the communication capacities, the system usually needs a central computational point with the knowledge of the actual network structure and parameters (e.g. [1]). The existence of such a computational point decreases the robustness of the system against the network damage and increases the communication load of the network. Furthermore, the routing of such information (the actual network structure and parameters) has to be solved in the case of the centralized algorithm.

In this paper, we propose a distributed algorithm, which computes the energy optimal routing without the need of any central computational or data point. The algorithm supposes that each node knows only the capacity and the cost (energy consumption per sent data unit) of the outgoing communication links of the node and the data which it is supposed to send and receive. The main purpose of this paper is to present the principle of new distributed routing algorithms rather than to present an application ready algorithm. We believe that the presented approach can lead to a new efficient and highly adaptive routing algorithms for sensor networks.

B. Related Works

Traditionally, the routing problems for data networks have often been formulated as linear or convex multi-commodity network flow routing problems e.g. [2], [3], [4] for which many efficient solution methods exist [5], [6], [7], [8]. One of the advantages of this method is that several cost functions and constraints can be put together (e.g. different types of capacity and energy consumption constraints, Real-time constraints, etc.). Using the same underlying model, we can easily combine the solution of different works focused on partial problems.

There are several works, which focus on the decomposition of network problems described by convex optimization. A systematic presentation of the decomposition techniques for network utility maximization (NUM) is presented in [9], [10], [11]. The authors present several mathematical approaches to structural decomposition of the NUM problems and classify them. In [12], [13], [14] the authors use the dual decomposition to decompose cross-layer optimization problems into optimization of separated layers. The presented approaches lead to structural decomposition (e.g. to routing layer, capacity layer...) which is not suitable for derivation of the in-network distributed algorithm.

The decomposition of an optimal routing problem is presented e.g. in [15], [16], where the authors have focused on the node-path formulation of the routing problem and use the dual decomposition to find the distributed algorithm. The presented algorithms can be described as a negotiation between the source node and the path load. This approach is suitable for problems with a small number of communication

paths. However, in sensor networks routing problems, where many possible communication paths exist, we have to find a different way to distribute the routing algorithm. Moreover, these algorithms are limited to a strictly convex cost functions and fail in the case of linear cost functions.

C. Contribution and Outline

The main contributions of this paper are:

1) Introduction of a new distributed algorithm based on a dual decomposition of the node-link formulation of the routing problem. (The existing approaches use only the node-path formulation, which leads to different algorithms)
2) Presentation of an approach to distribute a general linear problems by dual decomposition (according to our knowledge, all works using the dual decomposition on the routing problems are limited to strictly convex cost functions and fail in the linear case).

The paper is organized as follows: Section II briefly describes the multi-commodity network flow model. In Section III the distributed algorithm and its derivation are presented. An example and computational complexity experiments are given in Section IV. Section V concludes the paper and mentions the future work.

II. Multi-commodity network flow model

In this section, we briefly summarize the basic terminology and specify the multi-commodity network flow model. For more details see e.g. [2], [5], [1].

The network is represented by an oriented graph, where for each device able to send or receive data, a node of the graph exists. The nodes are labeled as $n = 1,\dots,N$. Directed communication links are represented as ordered pairs (n_1,n_2) of distinct nodes. The links are labeled as $l = 1,\dots,L$. We define the set of the links l leaving the node n as $\mathscr{O}(n)$ and the set of the links l incoming to node n as $\mathscr{I}(n)$. The network structure is described with an incidence matrix A' in the node-link form.

$$A'_{n,l} = \begin{cases} 1, & l \in \mathscr{I}(n) \text{ (link } l \text{ enters node } n) \\ -1, & l \in \mathscr{O}(n) \text{ (link } l \text{ leaves node } n) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

By m we denote an index of the communication demand and by $\mathscr{M}$ we denote set of all communication demands. The communication demands can be seen as flow of various commodities incoming/leaving the network in some nodes. The flow of each communication demand has to satisfy the flow conservation law at each node (for given commodity the sum of flow incoming to the node is equal to the sum of flow leaving the node). $A'\vec{x}^{(m)} = \vec{s}^{(m)}_{out} - \vec{s}^{(m)}_{in},\ \forall m \in \mathscr{M}$. Where the column vector $\vec{s}^{(m)}_{in} \geq \vec{0}$ denotes the flow coming into the network, the $\vec{s}^{(m)}_{out} \geq \vec{0}$ denotes the flow leaving the network and the $\vec{x}^{(m)} \geq \vec{0}$ denotes the flow routed through the network for demand m. Notice, that a multi-source multi-sink problem can be described in this way.

Total volume of the flow in the links over all communication demands has to satisfy the capacity constraint $\sum_{m\in\mathscr{M}} \vec{x}^{(m)} \leq \vec{\mu}$. Where $\vec{\mu} \geq \vec{0}$ is column vector of the links capacities.

In summary, the network flow model imposes the following constraints on the network flow variables $\vec{x}^{(m)}$:

$$\begin{array}{rcll} A'\vec{x}^{(m)} & = & \vec{s}^{(m)}_{out} - \vec{s}^{(m)}_{in} & \\ \sum\limits_{m\in\mathscr{M}} \vec{x}^{(m)} & \leq & \vec{\mu} & \forall m \in \mathscr{M} \\ \vec{x}^{(m)} & \geq & \vec{0} & \forall m \in \mathscr{M} \end{array} \quad (2)$$

The task of the total energy minimization is to minimize the cost function $f_{cost} = \vec{c'}^{T} \sum_{m\in\mathscr{M}} \vec{x}^{(m)}$ by setting the flow vector $\vec{x}^{(m)}$ for all $m \in \mathscr{M}$ subject to the system of inequalities (2). The vector $\vec{c'}$ is a column vector of the energy consumption per sent data unit.

III. Decomposition of the routing problem

Without lost of generality, we rewrite the optimal routing problem into the equality form for more transparent presentation.

$$\begin{array}{rcl} \min\limits_{\vec{x}} & \vec{c}^{T}\vec{x} & \\ \textbf{subject to:} & & \\ A\vec{x} & = & \vec{b} \\ \vec{x} & \geq & \vec{0} \end{array} \quad (3)$$

Where

$$A = \begin{pmatrix} A' & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & A' & 0 \\ I & I & I & I \end{pmatrix} \quad \vec{b} = \begin{bmatrix} \vec{s}^{(1)}_{out} - \vec{s}^{(1)}_{in} \\ \vdots \\ \vec{s}^{(M)}_{out} - \vec{s}^{(M)}_{in} \\ \vec{\mu} \end{bmatrix} \quad (4)$$

$$\vec{x} = \begin{bmatrix} \vec{x}^{(1)} \\ \vdots \\ \vec{x}^{(M)} \\ \vec{z'} \end{bmatrix} \quad \vec{c} = \begin{bmatrix} \vec{c'} \\ \vdots \\ \vec{c'} \\ \vec{0} \end{bmatrix} \quad (5)$$

where I is the identity matrix.

To decompose the routing algorithm, we use a gradient optimization method to solve its dual problem. However, the linearity of the cost function of the problem (3) would cause oscillations in the gradient algorithm and prevents to find the optimal solution. Therefore, we use the proximal-point method (for details see [5]) to modify the problem into a strictly convex form, which allows the usage of the gradient method.

Using the proximal-point method the modified problem is:

$$\begin{array}{rcl} \min\limits_{\vec{y}} \min\limits_{\vec{x}} & \vec{c}^{T}\vec{x} + \varepsilon(\vec{x}-\vec{y})^{T}(\vec{x}-\vec{y}) & \\ \textbf{subject to:} & & \\ A\vec{x} & = & \vec{b} \\ \vec{x} & \geq & \vec{0} \end{array} \quad (6)$$

where $\varepsilon > 0$.

Please notice that the set of optimal solutions for the problem (6) is the same as for the problem (3). For optimal solution of the problem (6) holds $\vec{x} = \vec{y}$. In this way the routing

problem has been separated into two nested subproblems. The internal subproblem is minimization over the variable $\vec{x}$ and it is strictly convex. The outer subproblem minimize the internal one by the proximal-point variable $\vec{y}$.

A. Dual Problem

To solve the internal subproblem of (6) (minimization over the variable $\vec{x}$) we present its dual problem, which allow to derive the distributable gradient algorithm. According to Slater's conditions (see e.g. [7]) the optimal solution of the dual and primal problems are equal.

The Lagrangian function of the problem (6) is:

$$L(\vec{x},\vec{y},\vec{\theta}) = \vec{c}^T\vec{x} + \varepsilon(\vec{x}-\vec{y})^T(\vec{x}-\vec{y}) + \vec{\theta}^T(A\vec{x}-\vec{b}) \quad (7)$$

Where $\vec{x} \geq \vec{0}$ is primal variable and $\vec{\theta}$ is dual variable. The dual function W is:

$$W(\vec{y},\vec{\theta}) = \min_{\vec{x}\geq\vec{0}} L(\vec{x},\vec{y},\vec{\theta}) \quad (8)$$

Differentiation of the Lagrangian function (7) gives:

$$\nabla_x L = \vec{c} + A^T\vec{\theta} + 2\varepsilon(\vec{x}-\vec{y}) \quad (9)$$

$$\nabla_y L = -2\varepsilon(\vec{x}-\vec{y}) \quad (10)$$

The dual problem of (6) is:

$$\max_{\vec{\theta}} \; W(\vec{y},\vec{\theta}) \quad (11)$$

And its gradient:

$$\nabla_\theta W = A\vec{x} - \vec{b} \quad (12)$$

B. Dual Gradient Algorithm

Using the dual problem (11) and the dual function (8) we rewrite the routing problem (6) into form:

$$\min_{\vec{y}} \; \max_{\vec{\theta}} \; \min_{\vec{x}\geq\vec{0}} \; L(\vec{x},\vec{y},\vec{\theta}) \quad (13)$$

A gradient algorithm created from Equation (13) consist of 3 nested loops (one loop for each variable). The internal loop solves the subproblem (8) using the gradient of the Lagrangian function (9). The middle loop solves the dual problem (11) using its gradient (12). The outer loop minimizes over the proximal-point variable $\vec{y}$ using the gradient (10).

$$\begin{array}{l}
\text{LOOP 1} \\
\quad \text{LOOP 2} \\
\quad\quad \text{LOOP 3} \\
\quad\quad\quad \vec{x} = \left[\; \vec{x} - \alpha\nabla_x L \;\right]^+ \\
\quad\quad \text{END 3} \\
\quad\quad \vec{\theta} = \vec{\theta} + \alpha\nabla_\theta W \\
\quad \text{END 2} \\
\quad \vec{y} = \vec{y} - \alpha\nabla_y L \\
\text{END 1}
\end{array} \quad (14)$$

Where $\alpha > 0$ is a constant step size of the algorithm. However, distributed version of such algorithm would have problems with termination of the middle loop and with synchronization of the loops between the nodes in the network. Moreover the nested loops would cause a cubic increase of number of internal iterations.

To overcome these problems, we join all the loops into gradient algorithm with only one loop, where we update all the variables $\vec{y},\vec{\theta},\vec{x}$ simultaneously. Using Equations (9), (10) and (12) we derive one iteration of the algorithm:

$$\begin{aligned}
\vec{x}_{k+1} &= \left[\vec{x}_k - \alpha\left(\vec{c} + A^T\vec{\theta}_k + 2\varepsilon(\vec{x}_k - \vec{y}_k)\right)\;\right]^+ \\
\vec{y}_{k+1} &= \vec{y}_k + \alpha 2\varepsilon(\vec{x}_k - \vec{y}_k) \\
\vec{\theta}_{k+1} &= \vec{\theta}_k + \alpha(A\vec{x}_k - \vec{b})
\end{aligned} \quad (15)$$

The $\alpha > 0$ is a constant step size of the algorithm, k denotes the iteration number and symbol $[..]^+$ denotes a positive or zero value in each component of the vector $[..]^+ = max(\vec{0},..)$. The correctness of such algorithm is not seen directly form its derivation and has to be proven. The proof of the algorithm convergence is not a trivial problem and we have presented it in [17]. A necessary condition for the algorithm convergence is $\alpha \leq 1/2\varepsilon$. Moreover, we have performed several simulation experiments to test the algorithm convergence in Section IV.

The variables $\vec{x}_0$, $\vec{y}_0$ and $\vec{\theta}_0$ are set to arbitrary initial values. The closer the values are to the final solution, the faster the algorithm converges. This property can be used in the case of minor changes of the network structure during its operation or in case of a pre-computed routing e.g. based on Dijkstra's algorithm. However, we will not investigate this problem further in this paper.

C. Distributed Algorithm

The system of equations (15) is a description of one iteration of the distributable routing algorithm. However, to define the distributed algorithm for each node, we have to rewrite the equations (15) using (4), (5). Moreover we need to define for the variables $\vec{\theta}_k$ and $\vec{y}_k$:

$$\vec{\theta}_k = \begin{bmatrix} \vec{\theta}_k^{(1)} \\ \vdots \\ \vec{\theta}_k^{(M)} \\ \lambda_k \end{bmatrix} \quad \vec{y}_k = \begin{bmatrix} \vec{y}_k^{(1)} \\ \vdots \\ \vec{y}_k^{(M)} \\ \vec{z}_k'' \end{bmatrix} \quad (16)$$

The presented distributed algorithm is running on each node in the network and it is synchronized by the communication between the nodes. The algorithm for node n is presented in Table I.

We use $x_{k,i}^{(m)}$, $y_{k,i}^{(m)}$, $\theta_{k,i}^{(m)}$, c_i etc. to denote the i-th component of the corresponding vector.

Due to the structure of the matrix A and vectors $\vec{x}_k$, $\vec{y}_k$, $\vec{b}$ and $\vec{\theta}_k$ we rewrite the expressions (15) in order to compute the flow of the communication demand m in the link l into form of Equation (17). Where the expression l^- denotes index of the start node of the link l and l^+ denotes index of the end node of the link l.

Each node n is responsible for computation of the flow volume of the links leaving the node n and for computation of the corresponding dual variables. Therefore, node n computes

Table I
DISTRIBUTED ROUTING ALGORITHM EXECUTED IN NODE n.

1) Initialize the variables:

$$\begin{aligned} x_{0,l}^{(m)} &= x_{start,l}^{(m)} && \forall m \in \mathscr{M} \quad \forall l \in \mathscr{O}(n) \\ y_{0,l}^{(m)} &= y_{start,l}^{(m)} && \forall m \in \mathscr{M} \quad \forall l \in \mathscr{O}(n) \\ \lambda_{0,l} &= \lambda_{start,l} && \forall l \in \mathscr{O}(n) \\ \theta_{0,n}^{(m)} &= \theta_{start,n}^{(m)} && \forall m \in \mathscr{M} \\ z'_{0,l} &= z'_{start,l} && \forall l \in \mathscr{O}(n) \\ z''_{0,l} &= z''_{start,l} && \forall l \in \mathscr{O}(n) \\ k &= 0 \end{aligned}$$

2) Send/receive the variables to/from the neighbors.

$$\begin{aligned} \textbf{Send:} \quad & x_{k,l}^{(m)} && \forall m \in \mathscr{M}, \forall l \in \mathscr{O}(n) \\ & \theta_{k,n}^{(m)} && \forall m \in \mathscr{M} \\ \textbf{Receive:} \quad & x_{k,l}^{(m)} && \forall m \in \mathscr{M}, \forall l \in \mathscr{I}(n) \\ & \theta_{l^-}^{(m)} && \forall m \in \mathscr{M}, \forall l \in \mathscr{I}(n) \end{aligned}$$

3) Evaluate the equations for $k+1$ and for $\forall l \in \mathscr{O}(n)$ and $\forall m \in \mathscr{M}$:

$$\begin{aligned} x_{k+1,l}^{(m)} &= \Big[x_{k,l}^{(m)} - \\ &- \alpha\big(c'_l + \theta_{k,l^+}^{(m)} - \theta_{k,n}^{(m)} + \lambda_{k,l} + 2\varepsilon(x_{k,l}^{(m)} - y_{k,l}^{(m)})\big) \Big]^+ \\ y_{k+1,l}^{(m)} &= y_{k,l}^{(m)} + 2\alpha\varepsilon(x_{k,l}^{(m)} - y_{k,l}^{(m)}) \\ z'_{k+1,l} &= \Big[z'_{k,l} - \alpha\big(\lambda_{k,l} + 2\varepsilon(z'_{k,l} - z''_{k,l})\big) \Big]^+ \\ z''_{k+1,l} &= z''_{k,l} - \alpha\big(2\varepsilon(z'_{k,l} - z''_{k,l})\big) \\ \theta_{k+1,n}^{(m)} &= \theta_{k,n}^{(m)} + \\ &+ \alpha\Big(\sum_{i \in \mathscr{I}(n)} x_{k,i}^{(m)} - \sum_{i \in \mathscr{O}(n)} x_{k,i}^{(m)} - \vec{s}_{out,n}^{(m)} + \vec{s}_{in,n}^{(m)}\Big) \\ \lambda_{k+1,l} &= \lambda_{k,l} + \alpha\Big(\sum_{m \in \mathscr{M}} x_{k,l}^{(m)} + z'_{k,l} - \mu_l\Big) \end{aligned} \tag{17}$$

4) $k = k+1$, go to step 2 and start a new iteration loop.

$x_{k+1,l}^{(m)}$ and $y_{k+1,l}^{(m)}$ for all $l \in \mathscr{O}(n)$ and all $m \in \mathscr{M}$, $z'_{k+1,l}$ and $z''_{k+1,l}$ and $\lambda_{k+1,l}$ for all $l \in \mathscr{O}(n)$ and $\theta_{k+1,n}^{(m)}$ for all $m \in \mathscr{M}$.

The algorithm for node n is presented in Table I. In step 1, the algorithm initializes the variables. In steps 2 the node communicates the variables to the neighbor nodes. In step 3 the node computes new values of the k-th iteration.

In (17), node n computes $x_{k+1,l}^{(m)}$ for all links leaving node n. It is a function of the local variables $x_{k,l}^{(m)}$, $y_{k,l}^{(m)}$, $\lambda_{k,l}$, $\theta_{k,n}^{(m)}$ and the variables $\theta_{k,l^+}^{(m)}$ of the neighbor nodes. Similarly, the computation of the others variables is a function of the local variables and the variables of the neighbor nodes that are within one hop communication distance.

IV. EXPERIMENTS

To demonstrate the behavior and the correctness of the distributed routing algorithm, we have performed several experiments in Matlab. We have focused on a basic problem, where for each communication demand one node is supposed to send data flow to one sink node (i.e. multi-commodity mono-source, mono-sink problem). $s_{in,n_1}^{(m)} = 1$ for the source node n_1 and $s_{out,n_2}^{(m)} = 1$ for the sink node n_2 of communication demand m.

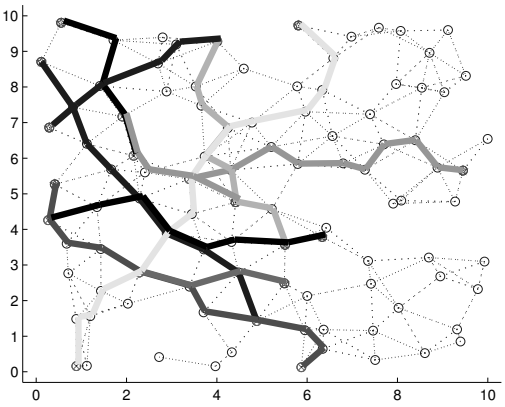

Figure 1. Optimal data flow routing (multi-commodity, mono-source, mono-sink problem)

The random networks for the experiments have been constructed as follows: We consider a square field of size [*size* × *size*], where the *size* is changing during the experiments. The field is divided into sub-squares of size [1 × 1]. One node is randomly placed into each sub-square and the communication distance is set to 1.5 (i.e. node n_1 can communicate with node n_2, if and only if their Euclidean distance is less than 1.5). Please notice, that our network is close to the "unit-disk network" [18]. The communication costs $\vec{c'}$ per transmitted data flow unit have been set as the power of the distance between the nodes. The link capacities have been set to one $\mu_l = 1$. The constants of the algorithm have been set as: $\alpha = 0.03$, $\varepsilon = 0.3$. The initial values $x_{start,l}^{(m)}$, $y_{start,l}^{(m)}$, $\theta_{start,n}^{(m)}$ have been set to 0 and $z'_{start,l} = \mu_l$ and $z''_{start,l} = \mu_l$ for all experiments except IV-B. Only feasible problems are used.

During the experiments we evaluate the number of iterations k needed to achieve the optimal solution as a number of iterations needed to achieve 0.01% deviation of the cost function from the optimal value during last 100 iterations. (the optimal value was computed separately by a centralized algorithm for evaluation purposes only)

A. Example

To present the resulting optimal data flow routing in the network and the progress of the Lagrangian function during the computation, we have performed an experiment based on the network described above. The field *size* has been set to 10 (i.e. 100 nodes in the network) and the number of communication demands has been set to 10.

The optimal data flow routing is shown in Figure 1. The progress of the Lagrangian function (7) and its optimal value are presented in Figure 2.

On the progress of the Lagrangian function the algorithm convergence can be observed as difference from its optimal value. Unfortunately, the progress of the Lagrangian function is not generally monotonic, which makes the proof of the

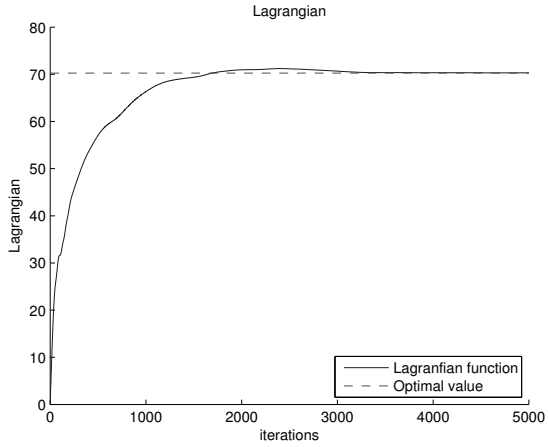

Figure 2. Progress of the Lagrangian function

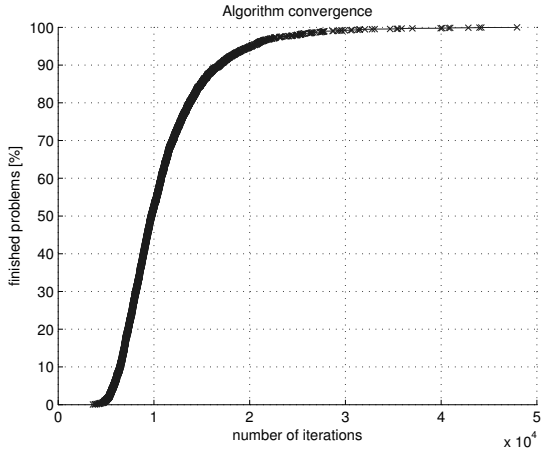

Figure 3. Algorithm convergence with random starting points.

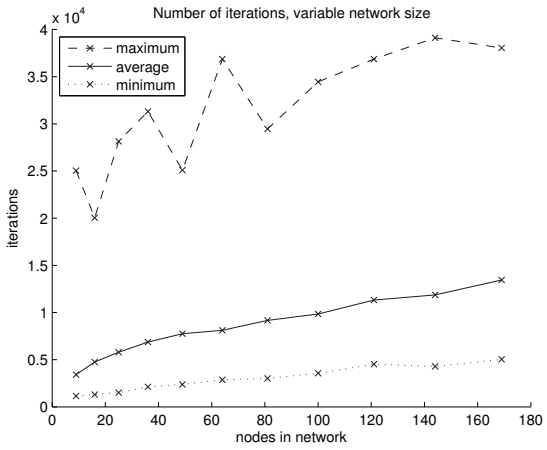

Figure 4. Number of iterations in relation to the number of nodes.

algorithm convergence more difficult.

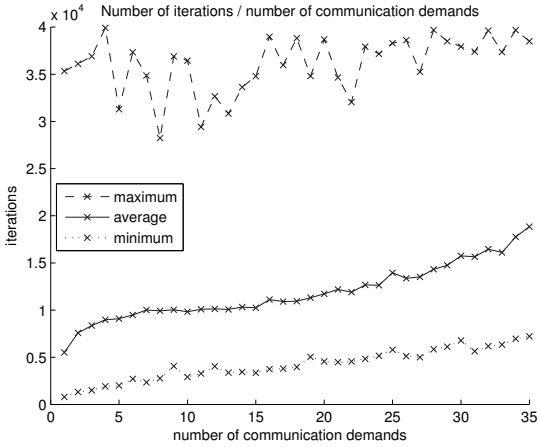

Figure 5. Number of iterations in relation to the number of communication demands.

B. Algorithm convergence

We have preformed a set of experiments with random starting points on random networks and we have evaluate the algorithm convergence. The field *size* has been set to 10. There are 10 communication demands in the network. The initial values have been set randomly from intervals: $x^{(m)}_{start,l} \in \langle 0,2\rangle$, $y^{(m)}_{start,l} \in \langle 0,2\rangle$, $\theta^{(m)}_{start,n} \in \langle -20,20\rangle$, $\lambda^{(m)}_{start,l} \in \langle -20,20\rangle$ and $z'_{start,l} = \mu_l$ and $z''_{start,l} = \mu_l$. The intervals have been chosen as double value of maximum/minimum of typical optimal values.

The algorithm has been run 2800 times on random networks and the results are presented on Figure 3. There is number of iterations placed on the horizontal axis and number of experiments which has been finished before the number of iterations on the vertical axis.

This experiment provides an important practical verification of the theoretical proof of the algorithm convergence. It can be seen, that approximately 95% of the experiments have been finished in 20000 iterations. It is obvious, that the number of the iterations is too big for the practical implementation of the algorithm. However, the main aim of this work was to introduce a new approach for distributed routing algorithms. The objective of our future work is to decrease the number of iterations.

C. Number of iterations

To demonstrate the statistical behavior of the algorithm, we have performed two tests. In the first one we have gradually increased the field *size* from 3 to 13 (i.e. from 9 to 169 nodes) for 10 communication demands. In the second one we have gradually increased the number of communication demands for field *size* 10. The computation has been repeated, on random networks, 300 times for each field *size*/number of demands.

The results have been evaluated as a maximum, average and minimum number of iterations needed to achieve the optimal value and it is presented in Figure 4 for variable field *size* and in Figure 5 for variable number of demands.

The important outcome of this experiment is the observation, that the number of the iterations is approximately linear. It follows, that the algorithm is well applicable to a big networks with many communication demands.

V. Conclusion

In this paper we have presented a distributed algorithm for the energy optimal data flow routing in sensor networks. We have described the routing problem as a multi-commodity network flow optimization problem and used the dual decomposition method to derive the distributed algorithm. The algorithm does not need any central computational node with knowledge about the whole network structure. This rapidly increases the robustness of the algorithm in the case of partial network damage. The algorithm uses only peer-to-peer communication between the neighboring nodes which allows the routing update using only the local communication. We have performed several simulation experiments to evaluate the algorithm behavior and to test its convergence. The mathematical proof of the algorithm convergence is available in [17].

The main purpose of this paper was to present the basic concept of new distributed routing algorithms. From the experimental section it is seen that the presented algorithm is not application ready because of the high number of iterations, which would lead to high number of communications. However the main strength of the algorithm is not to find the whole optimal routing in unknown networks, but to adapt an existing routing in case of local network changes (dead/new node, lost of connection, etc.) where the number of iterations should significantly decrease.

In future work we are going to improve the performance of the algorithm, using heuristics based on a partial knowledge about the network structure (e.g. node geographical position) and heuristics based on Newton's method, which should significantly decrease the number of iterations. Moreover we are going to evaluate the algorithm performance in the case of local network changes.

Considering the fact, that the algorithm is based on Linear programming formulation, we believe that the principle of the algorithm and the approach used to its derivation can be used to solve many different problems in the sensor networks area, like resource sharing, network localization, object tracking, etc.

Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic under the Project P103/10/0850, and Research Program MSM6840770038.

References

[1] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, July 2004.

[2] D. P. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 2004.

[3] M. Chiang, *Geometric programming for communication systems*. Foundations and Trends in Communications and Information Theory, August 2005, vol. 2.

[4] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2004.

[5] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[6] A. Ouorou, P. Mahey, and P. Vial, "A survey of algorithms for convex multicommodity flow problems," *Management Science*, vol. 46, no. 1, pp. 126–147, January 2000.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[8] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 435–445, February 2006.

[9] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1439 – 1451, Aug 2006.

[10] ——, "Alternative decompositions for distributed maximization of network utility: Framework and applications." in *25th IEEE International Conference on Computer Communications (INFOCOM 2006).*, April 2006, pp. 1 – 13.

[11] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures." in *Proceedings of the IEEE.*, vol. 95, Jan. 2007, pp. 255–312.

[12] B. Johansson, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks." *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1535 – 1547, Aug 2006.

[13] B. Johansson and M. Johansson, "Primal and dual approaches to distributed cross-layer optimization." in *16th IFAC World Congress*, Prague, Czech Republic, July. 2005.

[14] H. Nama, M. Chiang, and N. Mandayam, "Utility-lifetime trade-off in self-regulating wireless sensor networks: A cross-layer design approach," *ICC '06. IEEE International Conference on Communications*, vol. 8, pp. 3511–3516, June 2006.

[15] J. Tsitsiklis and D. Bertsekas, "Distributed asynchronous optimal routing in data networks." *IEEE Transactions on Automatic Control*, vol. 31, pp. 325 – 332, Apr. 1986.

[16] S. Low and D. Lapsley, "Optimization flow control. I. basic algorithm and convergence." *IEEE/ACM Transactions on Networking (TON)*, vol. 7, pp. 861 – 874, Dec. 1999.

[17] J. Trdlička and Z. Hanzálek, "Proof of the algorithm convergence," Department of Control Engineering, Faculty of Electrical Engineering, CTU Prague, Czech Republic, Tech. Rep., 2010. [Online]. Available: http://dce.felk.cvut.cz/hanzalek/DistRouting/ConvergenceProof.pdf

[18] M. G. C. Resende and P. M. Pardalos, *Handbook of Optimization in Telecommunications*. Springer, 2006.

[19] D. P. Bertsekas, *Nonlinear Programming*. Massachusetts Institute of Technology, 1999.