



Data Replication Based on Cuckoo Search in Mobile Ad-Hoc Networks

Takeru Kurokawa and Naohiro Hayashibara^(✉)

Kyoto Sangyo University, Kyoto, Japan
{i1986061, naohaya}@cc.kyoto-su.ac.jp

Abstract. Mobile ad-hoc network consists of a collection of mobile devices that are interconnected each other. It has attracted attention in various areas such as information sharing and intelligent transportation. In particular, information sharing on MANET in an emergency is crucial for victims. In this situation, information on relief supplies and safety confirmation should be replicated to keep it on the network as long as possible. On the other hand, the cost of replicated information should be taken into account since the storage of each mobile device is limited. In this paper, we proposed a data replication protocol based on the cuckoo search that is a meta-heuristic algorithm inspired by the egg-laying habits of cuckoos to improve the availability of low demand data in MANET. It realizes the space-efficient data replication without a significant impact on data availability.

1 Introduction

Mobile ad-hoc network (MANET) is widely used by many applications such as the information sharing system in an emergency [10], indoor location tracking [1] and intelligent transportation in a smart city.

Now, we suppose the information sharing in the case of emergency (e.g., disaster). In this case, there is no network infrastructure available. So, MANET that consists of mobile phones is an alternative to a network infrastructure (e.g., wifi access points) for sharing information among victims. Victims in a shelter upload information on disaster relief supplies, medical support, livelihood support, and confirmation of individual safety.

The number of replication of data is often in proportion to the number of requests in most of replication approaches, for example, Path replication [3] and Owner replication [8]. Thus, data that are frequently required by users create many replicas. So, data on disaster relief supplies and medical, livelihood support are required by many people and replicated actively. Contrary to this, data with low demand creates the small number of replicas. This type of data could be disappeared easily in MANETs by node's leaving or churn. On the other hand, those data could be important for specific individuals, for example, the safety information on family and relatives.

In general, the number of replicas is correlated with the availability of the data. The availability of the data increases as the number of its replicas increases.

Meanwhile, the cost of the storage of each node becomes high in this case. Thus, there exists a tradeoff between data availability and storage cost.

Now, we assume a mobile ad-hoc network that consists of mobile devices and information is shared on it in the case of emergency (e.g., disaster). In this case, we need to take into account the storage cost in each device because it is strictly limited. We also make low demand (but still important) information available on the network as long as possible at the same time. For this purpose, we propose a novel approach for data replication on MANETs based on Cuckoo search. Cuckoo search [13] is a meta-heuristic algorithm inspired by the egg-laying habits of cuckoos. It is known as an efficient solution for nonlinear global optimization problems, and it is proven that it guarantees the global convergence by using Lévy flight [5].

In this paper, we use Cuckoo search for allocating the replicated data to nodes in MANET. Our approach is based on the protocol proposed in [7], which is a replication protocol to improve the availability of low demand data. We extend the protocol using Cuckoo search for properly allocating replicated data. It improves the efficiency of the storage cost without a significant reduction of data availability.

We also measure the availability of data, the storage cost, the hit ratio of search queries, and the average distance to data while searching on the proposed protocol, and compare with other protocols.

2 Related Work

First, we introduce some of the replication protocols in MANETs and P2P systems. Then we explain Cuckoo search for solving optimization problems.

2.1 Replication in MANETs and P2P Systems

Many data replication protocols in MANETs and P2P systems have been developed so far.

Owner replication [8] is a simple replication protocol which replicates data to the node that issues a search request for it. It means that one replicated data corresponds to the search query. Thus, the cost of replicating data is low. However, this protocol may require a lot of messages for searching for data.

Path replication [3] creates replicas of the data and allocates them to all nodes along the path from the requesting node to the providing node in a search query for it. This protocol reduces the search traffic (i.e., the number of messages) by a factor of three compared to Owner replication. On the other hand, it imposes higher storage cost than Owner replication because of the larger number of replicated data.

Kageyama and Shibusawa proposed the replication protocol for improving the availability of low demand files in super-node based P2P systems [7]. It replicates the high demand files using Owner replication. In addition, it computes a demand forecast for low demand files and decides the number of replicas to be created.

2.2 Cuckoo Search

Cuckoo search proposed by Yang and Deb [13] is a metaheuristic algorithm inspired by the brood parasitic behavior of the common cuckoo species. They lay their eggs in the nest of other species of birds. A cuckoo egg closely mimics the eggs of the host ones. The host birds may incubate and hatch the cuckoo egg, or they may recognize the intruding egg and abandon the nest. We now explain Cuckoo search algorithm shown in Algorithm 1.

Algorithm 1. Cuckoo search via Lévy flight

```

1: Initialize:
   Objective function :  $f(x), x = (x_1, \dots, x_d)^T$ ;
   Initial population of host nests :  $x_i (i = 1, 2, \dots, n)$ ;
    $t \leftarrow 0$  ▷ Counter of the current generation
    $t_{max} \leftarrow$  Maximum number of generations
2: while  $t < t_{max}$  do
3:   Choose a nest  $x_j^{t-1}$  randomly.
4:   Search a new nest  $x_i^t$  by Lévy walk.
5:   if  $(F(x_i^t) > F(x_j^{t-1}))$ , then
6:     Replace  $x_j^{t-1}$  by  $x_i^t$ .
7:   end if
8:   Some of the worse nests are abandoned in probability  $p_a$  and new ones are built.
9:   Sort the list of nests and find the best nest.
10:   $t \leftarrow t + 1$ 
11: end while

```

First, each cuckoo chooses a nest x_j^{t-1} randomly, and a nest x_i^t by using Lévy flight. Then they are evaluated by the function F . The cuckoo replaces x_j^{t-1} by x_i^t if $F(x_i^t) > F(x_j^{t-1})$ and dumps an egg in the nest x_i^t . Then, it finds the best nest at the generation t by sorting the list of nests. Meanwhile, some of the worse nests are abandoned with a probability $p_a \in [0, 1]$ and new ones are built.

The line 6 in Algorithm 1 is the brood parasitic behavior of cuckoos. A new nest (or solution) x_i^{t+1} is computed by Lévy flight in Eq. 1.

$$x_i^{t+1} = x_i^t + \alpha \times \text{step_length} \quad (1)$$

step_length is a step length of Lévy flight, which obeys Lévy distribution, and α is a weight parameter for the step length. We set $\alpha = 1$ since this configuration is used in most cases [13].

3 System Model

We assume that a MANET consists of a set of nodes $N = \{N_1, N_2, \dots, N_{|N|}\}$, which are mobile devices (e.g., smartphones, laptop computers, wearable

devices). Each node has a unique identifier. Each pair of nodes can communicate with each other. It means that a node in the network has a multi-hop path to any other node. We can obtain the remaining amount of battery RB_i^t and storage RS_i^t of a node N_i at time t . We also define the capacity of battery CB_i and storage CS_i of a node N_i . Moreover, A_i^t is defined as a duration (minutes) that N_i joins the network at time t . We conducted simulations by a cycle-based simulator. Thus, A_i^t is represented as the number of cycles.

The MANET we assume is a dynamic distributed system. Nodes leave from the network by going out of the communication range of any other node or turning off. In addition, nodes can join the network as new nodes. Once a node leaves from the network, it joins it again with another identifier without any data and logs in the previous execution. It means that we assume fail-stop failure. The topology of a MANET is assumed to be a random geometric graph.

4 Data Replication Based on Cuckoo Search

Our approach is an extension of the protocol proposed by Kageyama and Shibusawa [7] for improving the availability of low demand data in MANETs. It allocates the replicated data based on the evaluation of nodes regarding the residual battery and storage. Although it assumes a super-node based P2P system, we assume a fully decentralized P2P system. We also use Cuckoo search [13] to find an eligible node for allocating replicated data.

First of all, we introduce the replication protocol proposed in [7]. It replicates the data based on the prediction of requests to each data. It assumes that the distribution of the data requests obeys Poisson distribution as follows.

$$P(X = t) = \frac{\lambda^t}{t!} e^{-\lambda} \quad (2)$$

λ is the average number of requests on certain data in the Eq. 2. So, the probability of requests to the data at time t is represented as the equation. Every node measures the number of requests to data that are stored in the node. The prediction of data requests is computed based on the exponential smoothing. Let MR^t and PR^t be the number of requests which is measured, and the predicted number of requests. So, PR^{t+1} will be represented as follows.

$$PR^{t+1} = \delta \cdot MR^t + (1 - \delta) \cdot PR^t \quad (3)$$

Data is distinguished as low demand if PR^{t+1} for the data becomes lower than the bottom five percent of the probability distribution $P(t + 1)$.

4.1 Allocation of Replicated Data

As we mentioned in Sect. 3, we assume that data are distinguished and labeled in the frequency of requests. So, data on disaster relief supplies, medical support and livelihood support are expected to be labeled as *high demand* because many

people could search and require this type of data. We assume that this type of data is replicated by Owner replication in this paper.

On the other hand, data that are labeled as *low demand* based on the prediction of requests (Eq. 3) are replicated and allocated by the proposed data allocation protocol based on Cuckoo search to reduce the storage cost of each node. The main purpose of the proposed protocol to improve the availability of this type of data without the additional cost on storage.

We use Cuckoo search for the allocation of replicated data in MANET. Each node N_i has two parameters, the capacity of battery CB_i and storage CS_i . Moreover, it has three variables, the remaining amount of battery RB_i^t and storage RS_i^t , and the duration A_i^t at time t . The proposed protocol evaluates the nodes in the network based on the following equation and computes the evaluation value E_i of each node N_i .

$$E_i = \ln \left(\left(\alpha \times \frac{RB_i^t}{CB_i} + \beta \times \frac{RS_i^t}{CS_i} \right) \times A_i^t \right)$$

α and β are weights for the evaluation where $\alpha, \beta = [0, 1]$. To simplify the capacity of battery and storage, CB_i and CS_i are defined as 100.0. Thus, RB_i^t and RS_i^t are variables from 0.0 to 100.0.

We assume that data are disappeared from the network by nodes' leaving. A node that has a large enough amount of battery are not likely to leave the network because of the battery shortage. The proposed protocol tends to allocate replicated data into such a node. On the other hand, it takes into account the residual space of storage. It computes E_i for each node N_i and searches the node that has the maximum value of E_i by Cuckoo search. It uses Lévy walk that is known as an efficient algorithm for a wide-area search. Since the topology of a MANET is usually represented as a graph, we use the Lévy walk algorithm for unit disk graphs [12].

The original Cuckoo search [13] assumes the optimization problem on continuous values. On the other hand, our target is the optimization of discrete values to find the optimal node for allocating replicated low demand data. So, we have to modify the Cuckoo search into the discrete one.

4.1.1 Discretization of Cuckoo Search

We assume that the topology of MANET is represented as a random geometric graph. Cuckoo search finds a node N_i with a high E_i value. For this purpose, we modify the original Cuckoo search into the discrete version as follows:

1. We assume the optimal node N_i regarding E_i .
2. Find a new node N_j by Lévy walk starting from N_i .
3. Compare the evaluation values E_i and E_j of N_i and N_j .
4. Replace the optimal one by N_j if $E_i < E_j$.

5 Performance Evaluation

The purpose of the evaluation is to clarify the following things.

- Data availability of replicated data in MANETs.
- Storage cost in replicated data.

We evaluate the things listed above by comparing with other replication protocols, Owner replication [8], Path replication [3], the protocol proposed by Kageyama [7]. We implemented these protocols and the proposed one on PeerSim simulator [6]. PeerSim is a cycle-based simulator for Peer-to-Peer systems implemented in Java.

5.1 Environment and Scenarios

We assume that the number of nodes $|N| = 2000$ and they are located uniformly at random in the 1000×1000 field. Each node is interconnected with the neighbor nodes that are within a certain communication range of it, hence the topology of a MANET is a random geometric graph.

The storage residual of each node is set 100, and the battery residual of it is decided randomly from 50 to 100. We execute 500 cycles for a simulation run.

We assume that 50 unique data (files) uploaded to the network. We also assume two scenarios regarding the ratio of high demand and low demand data as follows:

- Scenario 1: 50% of data is low demand and the rest of data are high demand.
- Scenario 2: 80% of data is low demand and the rest of data are high demand.

5.2 Data Availability

One of the purposes of data replication is improvement of data availability. In MANET, data may be disappeared by nodes' leave. Thus, data availability is an essential criterion in the evaluation.

We now define the data availability DA in each cycle of a simulation run as follows.

$$DA^t = \frac{D^t}{D_{original} + D_{replicated}} \quad (4)$$

$D_{original}$ and $D_{replicated}$ are the total number of the original data and that of the replicated data, respectively. The total number of data available at time t is denoted as D^t where $D^t = D_{original}^t + D_{replicated}^t$. Thus, the data availability DA^t at time t is represented in Eq. 4.

Figures 1 and 2 show the availability of high demand data and that of low demand data in Scenario 1, respectively. There is no difference between the approaches we compared. On the other hand, Kageyama's protocol and the proposed protocol improves the availability of low demand data compared to Owner replication and Path replication. In particular, Kageyama's protocol is better than the proposed one from 280 cycles though they are almost the same until 250 cycles (see Fig. 2).

Figures 3 and 4 show the availability of high demand data and that of low demand data in Scenario 2, respectively. There is also no difference between the

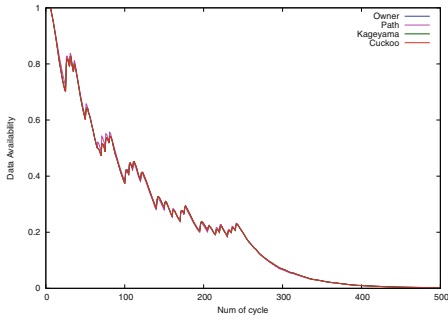


Fig. 1. The availability of high demand data in Scenario 1.

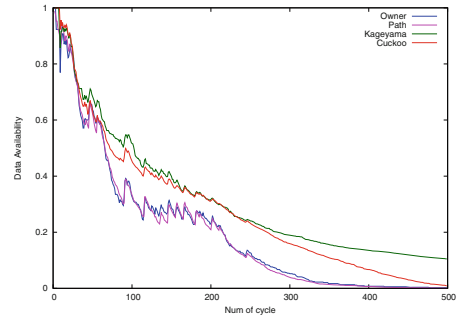


Fig. 2. The availability of low demand data in Scenario 1.

approaches, just like Fig. 1. Figure 4 shows that Kageyama's protocol is slightly better than the proposed protocol from 150 to 250 cycles, and the difference becomes bigger from 300 cycles. Owner replication and Path replication drop the availability of low demand data at 100 cycles, and the gap between others is getting bigger.

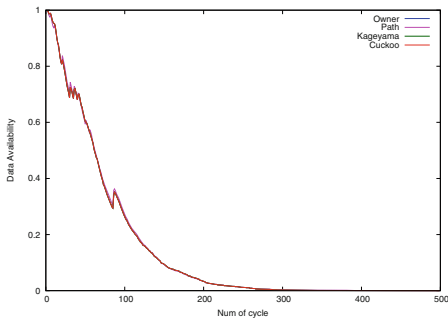


Fig. 3. The availability of high demand data in Scenario 2.

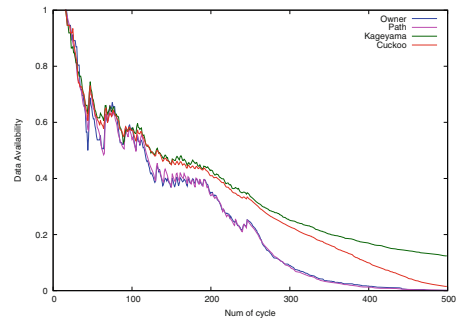


Fig. 4. The availability of low demand data in Scenario 2.

5.3 Storage Cost

The storage cost is crucial for replication protocols. Improvement of availability of data usually imposes the cost of storage. In other words, there exists a tradeoff between them. We measure the average residual of storage of each node and the cumulative storage usage.

Figures 5 and 6 show the average residual of storage of each node and the cumulative storage usage in Scenario 1. Figure 5 indicates that the proposed protocol is most efficient regarding the storage cost of each node. It improves

the average residual of storage more than 20% compared to Kageyama’s protocol at 180 cycles or later. Figure 6 shows that the difference between the proposed protocol and Kageyama’s one regarding the cumulative storage usage becomes bigger from 300 cycles.

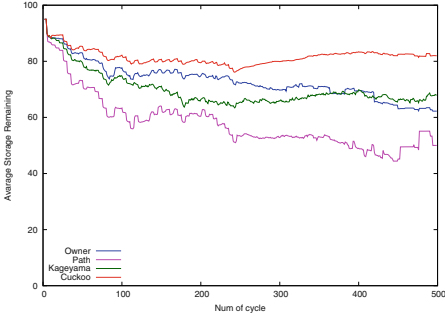


Fig. 5. The average storage residual of each node in Scenario1.

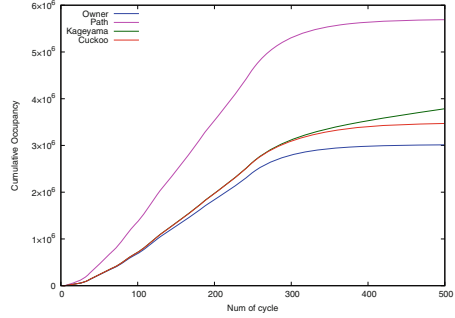


Fig. 6. The cumulative storage usage in Scenario 1.

Figures 7 and 8 show the average residual of storage of each node and the cumulative storage usage in Scenario 2. Figure 7 indicates that the proposed protocol and Owner replication show a similar tendency on the average residual of storage. On the other hand, the difference between the proposed protocol and Kageyama’s protocol becomes more significant from an early time (50 cycles). Figure 8 also shows the efficiency of the proposed protocol on the cumulative storage usage compared to Kageyama’s protocol that increases the storage cost linearly. In the case that has a large proportion of low demand data, the storage efficiency of the proposed protocol is emphasized compared to Kageyama’s protocol.

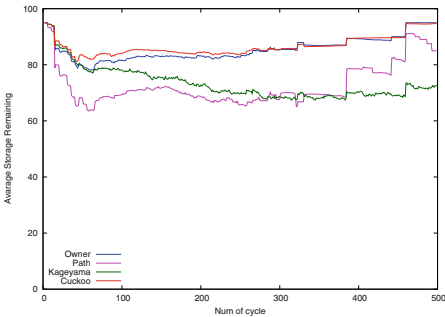


Fig. 7. The average storage residual of each node in Scenario2.

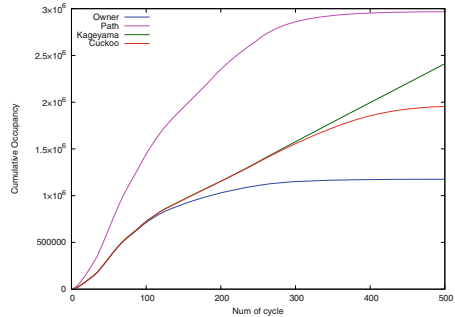


Fig. 8. The cumulative storage usage in Scenario 2.

5.4 Hit Ratio and Average Distance to Data

We measured the hit ratio of search queries and the average distance to the required data when it was found in the network. The average distance between the searcher and the node that stores the data is represented by the hop count in the network. We compare four protocols, Owner replication [8], Path replication [3], the protocol proposed by Kageyama et al. [7], and our proposed protocol. Tables 1 and 2 show the results on the hit ratio and the average distance to the required data in Scenario 1 and 2, respectively.

Our proposed protocol and Kageyama's protocol show the same result as Owner replication regarding high demand data because they use Owner replication for replicating high demand data.

The proposed protocol is slightly worse than Kageyama's protocol on low demand data. This is because the number of replication of each data is reduced in the proposed one for saving the storage usage of each node. On the other hand, it is still better than Owner replication and Path replication.

Table 1. The comparison on the hit ratio and the distance to the data in Scenario 1.

Protocol	Demand	Hit ratio	Avg. distance
Proposed	High	81%	8.09
	Low	88%	12.88
Kageyama et al. [7]	High	81%	8.09
	Low	97%	9.28
Owner replication	High	81%	8.09
	Low	55%	17.57
Path replication	High	88%	8.47
	Low	63%	14.30

Table 2. The comparison on the hit ratio and the distance to the data in Scenario 2.

Protocol	Demand	Hit ratio	Avg. distance
Proposed	High	75%	8.20
	Low	92%	10.93
Kageyama et al. [7]	High	75%	8.20
	Low	98%	8.42
Owner replication	High	75%	8.20
	Low	56%	17.00
Path replication	High	78%	7.05
	Low	70%	14.44

6 Conclusion

We proposed the allocation mechanism of replication data based on Cuckoo search for improving the availability of low demand data and implemented it in the modified version of the replication protocol proposed in [7]. The primary purpose of this paper is to clarify the impact of Cuckoo search on the availability of low demand data and the storage cost of replicated data. There exists a tradeoff between the availability and the storage cost.

Our simulation result shows that the proposed protocol improves the efficiency of both the average storage cost of each node and the cumulative storage usage compared to Kageyama's protocol. On the other hand, the proposed protocol is getting worse than Kageyama's one regarding the availability of replicated data as time passes though both of them are almost the same on it at an early period. Our proposed protocol is slightly worse than Kageyama's one regarding the hit ratio of search queries and the average distance to the required data. However, It is still useful when considering the storage cost of each device.

When we suppose the information sharing in the case of a disaster, the availability of data is essential as well as the storage cost of mobile devices, the hit ratio and distance to the required data. Our proposed protocol shows the efficiency of the storage of each device without a significant impact on the availability, especially on low demand data. Thus, our protocol is suitable for such a situation.

In future work, we will further study to improve the availability take into account several factors of each node (mobile devices).

References

1. Ali, A., Latiff, L.A., Faisal, N.: GPS-free indoor location tracking in mobile ad hoc network (MANET) using RSSI. In: 2004 RF and Microwave Conference (IEEE Cat. No. 04EX924), pp. 251–255 (2004). <https://doi.org/10.1109/RFM.2004.1411119>
2. Chandrakala, C.B., Prema, K.V., Hareesha, K.S.: Improved data availability and fault tolerance in MANET by replication. In: 2013 3rd IEEE International Advance Computing Conference (IACC), pp. 324–329 (2013). <https://doi.org/10.1109/IAdCC.2013.6514244>
3. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. *SIGCOMM Comput. Commun. Rev.* **32**(4), 177–190 (2002). <https://doi.org/10.1145/964725.633043>
4. Hara, T.: Data management issues in mobile ad hoc networks. *Proc. Jpn. Acad. Ser. B* **93**(5), 270–296 (2017). <https://doi.org/10.2183/pjab.93.018>
5. He, X., Wang, F., Wang, Y., X.S.Y.: Nature-inspired algorithms and applied optimization. In: *Studies in Computational Intelligence*, vol. 744. Springer (2018)
6. Jelasity, M., Montresor, A., Jesi, G.P., Voulgaris, S., Artecconi, S., Hales, D., Marcozzi, A., Picconi, F.: *Peersim* (2016). <http://peersim.sourceforge.net/>
7. Kageyama, J., Shibusawa, S.: Replication that prevents low-demand files from disappearing in P2P network. In: *Proceedings of The First Forum on Data Engineering and Information Management* (2009). (in Japanese)

8. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th International Conference on Supercomputing, ICS 2002, pp. 84–95. ACM, New York (2002). <https://doi.org/10.1145/514191.514206>
9. Mukilan, P., Wahi, A.: CDRA: consistency based data replication algorithm for manet. *Int. J. Comput. Appl.* **51**(14), 1–8 (2012)
10. Sakano, T., Kotabe, S., Komukai, T., Kumagai, T., Shimizu, Y., Takahara, A., Ngo, T., Fadlullah, Z.M., Nishiyama, H., Kato, N.: Bringing movable and deployable networks to disaster areas: development and field test of mdru. *IEEE Netw.* **30**(1), 86–91 (2016). <https://doi.org/10.1109/MNET.2016.7389836>
11. Shen, H.: An efficient and adaptive decentralized file replication algorithm in P2P file sharing systems. *IEEE Trans. Parallel Distrib. Syst.* **21**(6), 827–840 (2010). <https://doi.org/10.1109/TPDS.2009.127>
12. Shinki, K., Nishida, M., Hayashibara, N.: Message dissemination using lévy flight on unit disk graphs. In: Proceedings of the 31st IEEE International Conference on Advanced Information Networking and Applications (AINA 2017), Taipei, Taiwan ROC (2017)
13. Yang, X.: Suash deb: cuckoo search via lévy flights. In: 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), pp. 210–214 (2009). <https://doi.org/10.1109/NABIC.2009.5393690>