Applying neural network based algorithms in communication technology

Theses excerpt

Dávid Tisza

Scientific advisors:
Dr. János Levendovszky
dr. András Oláh

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
Roska Tamás Doctoral School of Sciences and Technology

Budapest, 2018

1 Technological motivation

Due to recent and vast improvement in sensor technology and to the yet unfailing trend set by Moore's law, multitude of novel fields opened up for new applications. The platform carrying these novel applications are at the same time required to support mobility and flexibility at the end user side. As the applications become evermore complex their supporting systems have to deal with more demands. These (such as their own communication networking subsystem, central processing units or the devices in the background network which they communicate with) have to act more intelligently and adapt to the new demands arriving from the upper layers. Also the majority of the end user devices are portable and use battery as a power source, it becomes increasingly important to take this into consideration at the widest range of system design possible.

Typically into these areas we could count in the IoT based applications, monitoring and intervening systems that are based on WSNs [2], peer-to-peer and “broadcast" type relaying and processing systems dealing with multimedia streams (let that be video or audio) or most of the cloud based services [40]. These services and systems have the common aspect of providing a certain QoS, while their resources are time and location dependent and also limited [45]. On inherent shared resources like on the radio subsystem these constraints appear even more stringent. Furthermore scheduling tasks efficiently in these distributed environments are imperative.

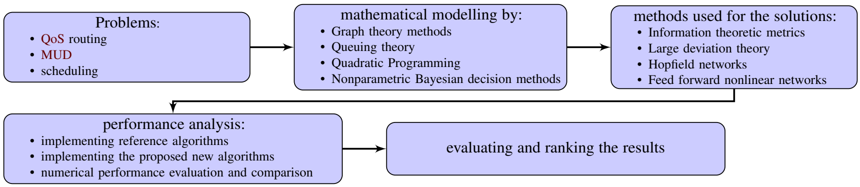
Networking technologies used today are following a layered structure [16] and most of them form packet switched communication networks. In these networks typically there are no separate resource dedicated for signaling and controlling, but without these processes providing a required QoS can be mostly done in a best effort manner if possible at all. Requiring these crucial processes to be present means that they have to isolate an additional portion from the resources bearing the payload to themselves. Thus the total capacity from the end user perspective further diminishes in contrast to their signaling-less counterparts. The best effort like structures which build up the traditional packet switched networks do not or rarely use signaling to govern their internal mechanisms in respect to QoS. These structures consist of protocols defined by OSI in several layers. Examples for these in the lower OSI layers (physical, data-link, network) are the Ethernet [1], the 802.11 [15] or the IP. Although their design does not incorporate providing QoS directly [32, 20], they are widespread because of their reliability and usability [20]. One of the base questions of the packet switched networks is to find an appropriate path and scheduling for the data packets to traverse through the network from the source to the destination. [39, 20, 28, 9, 8]. Applications using wireless technologies in the physical layer face even more constraints due to the shared nature of the radio media, which has to be accounted for if one requires to provide QoS.

In the light of this the open questions that this dissertation aims to answer are:

- How can one find an appropriate path in a packet switched network which provides a QoS. (QoS unicast, multicast routing)
- How can one perform scheduling tasks in communication networks efficiently with algorithms which lend themselves to parallelization. (scheduling)
- How can one solve at the physical layer which use wireless technologies near optimally the MUD problem in a parallelized fashion. (Multiuser Detection)

1.1 Applied methodology

This dissertation uses the following general approach to investigate the problems and bring up possible solutions as theses:



The summary of the problems, the used algorithms, theories and used tools can be found at Table 1.

technological challenges	***model formulation and problem category***	***theoretical performance***	***used methods and algorithms***	***related theses***
Unicast routing	using random link descriptors and LAS reducing the problem to additive and bottleneck type metrics	The path of choice is capable of providing a QoS	Queuing models , Markov modulated Poisson distributions, Gaussian approximation, Large deviation theory	Thesis I.1 Thesis I.2
Optimizing LAS	Applying information theoretic metrics (Link Entropy and Signaling Entropy) as constrains in optimization	Using the resulting LAS the bandwidth of the signaling process in the network can be bound and kept under a predefined value while at the same time the uncertainty of the link states in the network is also kept under a well defined value.	Information theoretic measures, exhaustive search, general nonlinear constrained optimization methods	Thesis I.4

- continuing on the next page -

Table 1 – continued from previous page –

technological challenges	*model formulation and problem category*	*theoretical performance*	*used methods and algorithms*	*related theses*
Multicast routing	using random link descriptors search for a CGSMT in the network and reformulating the search of the CGSMT as an UBQP	Sending streams through the paths from the source and destination points provide the prescribed QoS, while they perform optimally in the sense that they strain the chosen resource minimally. These constraints posed on the resources are like minimal energy consumption or minimal total used bandwidth to deliver the payload.	Gaussian approximation, large deviation theory, Hopfield network, binary quadratic programming methods	Thesis I.3
Multiuser Detection (MUD)	UBQP and non-parametric Bayesian detection	The exact solution of the UBQP is the optimal solution of the problem. The proposed UBQP "solvers" lend themselves to parallelization well, since they are made up from simple nonlinear computing units. They give well performing sub-optimal solutions with fast convergence time. In case of the non-parametric models the proposed algorithm can arbitrarily approximate the optimal decision at the expense of increasing its complexity.	Algorithms operating on a hypergraph based dimension reduction and dimension addition, Hopfield network, binary quadratic programming methods, Feed Forward Neural Network (FFNN), logarithmic search	Thesis II.1 Thesis III.1
Scheduling	UBQP	Exact solution to the UBQP is the optimal solution to the problem. The proposed "solvers" give sub-optimal solutions at fast convergence speeds.	Hopfield networks, binary quadratic programming methods	Thesis II.1

Table 1: Problems investigated, used algorithmic tools and the related theses

2 New scientific results and theses of the dissertation

This dissertation presents the problems and the theses in three groups in three sections.

I) In the first group, I elaborate on the route searching problem providing QoS in packet switched networks both for unicast and multicast cases and propose novel methods to solve or approximate them. In this section I also investigate the link scaling problem and propose a method for the optimization using information theoretical measures.
The theses in this group are Thesis I.1, Thesis I.2, Thesis I.3 and Thesis I.4.

II) In the second group, I elaborate on the Multiuser Detection (MUD) and scheduling problems in communication technologies. I formulate these tasks as Binary Quadratic Programming (BQP) and I propose a family of algorithms which act as a suboptimal solver to the BQP problem. These algorithms lend themselves to parallelization well because of their inherent structure.
Thesis II.1 of this group can be found on page 23

III) In the third group, I propose a FFNN based algorithm which performs comparably to the non-parametric optimal Bayesian decision for the MUD problem defined for the wireless networks PHY layer.
Thesis III.1 of this group can be found on page 29.

Thesis group I - routing with incomplete information in unicast and multicast scenarios

In this thesis group I propose novel solutions to the problem of unicast and multicast QoS routing with incomplete information. These new methods are capable of finding a sub-optimal path and a multicast route set in polynomial time. Also in this thesis group I provide a solution to the OLS task by using information theoretical measures, such as "signaling entropy" and "link entropy". *The corresponding publication of the author is titled "Multicast Routing in Wireless Sensor Networks with Incomplete Information" [48]*

In networks where is no dedicated channel for signaling or controlling purposes (typical in IoT, networks using IP) these procedures consume additional resources. As a result it diminishes the capacity available for information transfer, however at the same time services heavily demand the speed and reliability of the underlying communication stack. This gives rise to the problem of finding QoS fulfilling paths. For example in the IoT vision every device can send and receive information and might act as an intermediate node. From an angle these devices can be seen forming a WSN. If the network contains battery operated devices then the applications also require reliable communication while keeping energy consumption at a minimal level (e.g.

consider a smart home application where the user should not be forced to change the batteries frequently or a smart agricultural application where battery change might not be feasible at all). On the other hand in an application where the energy consumption might not be a problem (e.g. a smart fridge, automotive application or a factory with smart production appliances) other types of reliability criteria exist that the (sub)networks must meet. Among others, these can be robustness to communication shortage, redundancy, efficient use of the communication bandwidth, responsiveness, etc. One can also consider peer-to-peer applications, e.g. video on demand services or voice over IP services, where large quantity of data needs to be reliably transported to the peers. In these scenarios both unicast and multicast type communication is common. Data is to be transmitted to a single or a set of destination nodes with the packets routed in a multi-hop manner where intermediate nodes are also used for packet forwarding.

Legacy network structures which were not designed for QoS but mostly for best effort usually have no dedicated channel for signaling. Examples of these protocols in the lower OSI layers (physical, data-link, network) are the Ethernet, 802.11 or IP. Despite they can not provide QoS natively by their design[32, 20], they are widespread because of their reliability and interoperability[20]. Nevertheless, there is a need to run QoS communication over these best effort platforms. One of the major challenges in IP networking is to ensure QoS routing, which selects paths to fulfill end-to-end delay or bandwidth requirements [39, 20, 28, 9, 8] as opposed to traditional shortest path routing. Because of the manifold optimization criteria[17], even in the unicast case QoS routing can prove to be much harder than the problem of finding optimal paths based on merely the hop count [17]. Protocols striving to provide QoS need to have some knowledge about the state of the network. This information has to be propagated also (signaling). QoS-aware routing protocols often propose a method to reduce the amount of state information that have to be kept synchronized among routers, called topology aggregation (e.g. in: QOSPF, PNNI). Thus, routing information describing a certain domain of the network have to be aggregated, which acts as another source of uncertainty of resource availability information. Another source of uncertainty is introduced when multiple hierarchical levels are connected. Such a pattern is inevitable from the base rules of the network design. These hierarchical levels are introduced because these networks connects through inter-domains.

Incorporating QoS into routing has been long studied[20]. For example an extension to the classical IP over ATM system to support application level QoS is studied in [32]. Also QoS aware routing algorithms exist in many levels, such as in inter-domain level [19], in MPLS based network parts [31], inside ASs [39], but routing is done mostly by a hierarchical manner. In networks following the OSI model the routing is done in the 3rd (network) layer. The OSPF routing protocol offers two, while PNNI offers many levels, where routing can be performed in a hierarchical manner [13]. Subnetworks on a given level of the hierarchy are abstracted as "nodes" for a higher layer and delay information in those subnetworks are aggregated into an average QoS parameter. On the other hand, randomly fluctuating traffic load on links can also result in random delays. Thus link delays are periodically advertised when the delay surpasses a given threshold (e.g. in PNNI and QOSPF standards, see [3, 13]). These thresholds are defined in advance. This prompts us to take delays into account as random variables characterized by their probability distribution functions over the interval between two reported thresholds [14, 43, 27]. The distribution of these thresholds (and therefore the length of the intervals over which the link delay is not fully characterized) can be equidistant or non-equidistant. In practice non-equidistant thresholds are used, since in this case the impact on network utilization by sending only signaling information (part of the available bandwidth is used for carrying information about link delays) is minimized.

The phenomena described above give rise to the task of routing with incomplete information. Namely, how to select paths to fulfill end-to-end delay requirements in the lack of the exact values of link delays [14, 9, 8]. Routing is then perceived as an optimization problem to search over different quality paths, where the quality of a path is determined by the probability of meeting the end-to-end QoS requirement [26, 14]. Unfortunately, routing with incomplete information in general cannot be reduced to the well-known Shortest Path Routing (SPR), thus it cannot be solved in polynomial time in general.

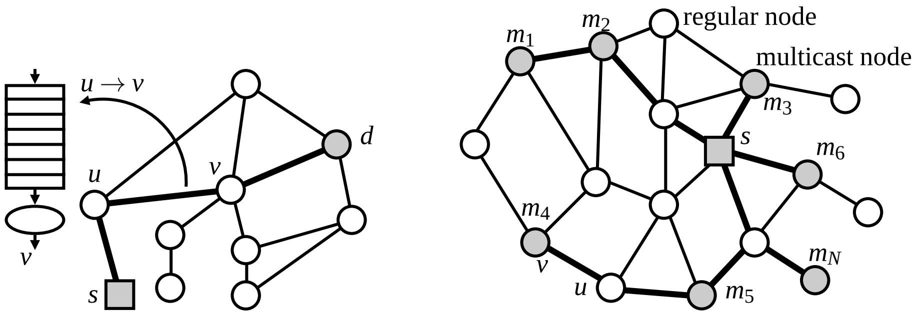
The multicast scenario can be seen as an extension of the previous problem and can be treated as the well-known GSMT problem, which has proved to be NP-hard even for deterministic link descriptors and cost functions[23]. The CGSMT problem is even more difficult, where the minimal cost tree is sought by guaranteeing a given QoS at the same time. This has proved to be NP-hard as well[24]. On the other hand, common multicast routing algorithms utilize stored network state information[18], which can quickly become obsolete due to the changing fading radio environment or traffic pattern. Link state information in clustered networks can also be incomplete due to aggregation. Heuristic algorithms for finding Multicast Trees are published in [42], however the computational complexity becomes overwhelming as the number of nodes increases in the network. That is why, we would like to benefit from the fast optimization properties of the HNN [38]. The execution time of such algorithms only depends on the interconnectivity of the network because every neuron represents an edge in the graph.

The QoS over an obtained path, however strongly depends on the "incompleteness" of link descriptors which is determined by the thresholds initiating Link State Advertisement (LSA)s. These thresholds are referred to as a Link Advertisement Scheme (LAS). The process of defining LASs over the network is called link scaling, which can be subject to further optimization in order to economize on signaling bandwidth. The smaller these intervals are, the smaller is the measure of “incompleteness” under which a path is selected. As a result, the routing performance is higher. On the other hand, small intervals forces more frequent announcements of the values of the link descriptors throughout the network, which implies that considerable portion of bandwidth is used up for transmitting signaling information. Thus, the optimization of the size of these intervals is a crucial task for network management. This problem is referred to as OLS. As can be seen, OLS is a constrained optimization problem, in which one has to maximize routing performance under the constraint of keeping the signaling bandwidth bellow a given threshold.

The communication network is modeled by a graph

$$G\left(V,E,\delta_{(u,v)},F_{(u,v)}(x)\right),$$

where nodes are denoted by index $u \in V$ and links referred to as node pairs $(u,v) \in E$; Each link $(u,v) \in E$ has a QoS link descriptor $\delta_{(u,v)}$ which is assumed to be a random variable subject to a CDF $F_{(u,v)}(x) = \mathbb{P}\left(\delta_{(u,v)} < x\right)$; The random variable $\delta_{(u,v)}$ is referred to as "bottleneck measure" if the smallest link measure determine the quality of the route ($\delta_{(u,v)}$ is a "bandwidth-like" variable); It is also referred to as "bottleneck measure" if the largest link measure determine the quality of the route ($\delta_{(u,v)}$ is a "energy consumption-like" variable) The random variable $\delta_{(u,v)}$ is referred to as "additive measure" if the sum of link measures contained in the path determine the quality of the path ($\delta_{(u,v)}$ is a "delay-like" variable);



(a) unicast case

(b) multicast case

Figure 1: graph model of the network

For the unicast case the objective is to find an optimal path $\tilde{R}$ from all possible paths $\mathscr{R}^{s\to d}$, which most likely fulfills the given QoS criterion, namely:

$$\tilde{R} = \operatorname*{argmax}_{R \in \mathscr{R}^{s \to d}} \mathbb{P}\left(\min_{(u,v) \in R} \delta_{(u,v)} \geq \mathrm{B} \right) \qquad \tilde{R} = \operatorname*{argmax}_{R \in \mathscr{R}^{s \to d}} \mathbb{P}\left(\sum_{(u,v) \in R} \delta_{(u,v)} < \mathrm{T} \right)$$

for the bottleneck and additive measures respectively.

For the multicast case the information source, typically a Base Station, is denoted by $s \in V$ and the set of multicast destination nodes by $M = \{m_1, m_2, \ldots, m_N\} \subset V$. In this case the objective is to find an optimal tree $\tilde{\mathrm{A}}$ from all multicast trees $\mathscr{A}^{s \to M}$ which most likely fulfills the given QoS criterion, namely:

$$\tilde{\mathrm{A}}_1 = \operatorname*{argmax}_{\mathrm{A} \in \mathscr{A}^{s \to M}} \mathbb{P}\left(\max_{(u,v) \in \mathrm{A}_E} \delta_{(u,v)} < \mathrm{P} \right) \qquad \tilde{\mathrm{A}}_2 = \operatorname*{argmax}_{\mathrm{A} \in \mathscr{A}^{s \to M}} \mathbb{P}\left(\max_{R_{sm} \in \mathscr{R}_T} \sum_{\substack{(u,v) \\ \in \\ R_{sm}}} \delta_{(u,v)} < \mathrm{T} \right)$$

for the bottleneck and additive measures respectively. Note that for the multicast bottleneck case the problem is formulated differently than in the unicast case, because the link descriptor is typically power consumption like quantity in WSNs, thus the bottleneck in this case is the most consuming element. But the derived conclusions are also applicable to a bandwidth-like link descriptor with minor modifications.

An advertisement of a link state change happen when the current value of link descriptor steps out from an interval defined by two threshold values, which is called Link State Advertisement (LSA). Figure 2 depicts an LSA in case the link descriptor is of delay type. When a link advertisement is received, the receiver could assume

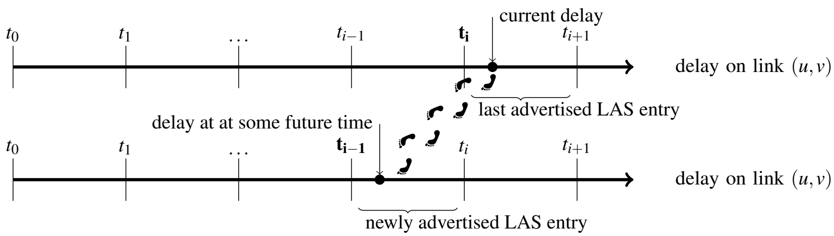


Figure 2: LSA in case the link descriptor is of delay type.

that the link descriptor of the sender node is within the new interval. The finer the threshold grid is defined on the link descriptors the more precise the receiver can be about the current state of the sender, but on the other hand it also means the frequency of the advertisement to increase. Here our objective is to strike an optimal balance between the signaling bandwidth and routing performance. To capture the underlying phenomena two information theoretical measures are introduced: the *Signaling Entropy (SE)* and the *Link Entropy (LE)*.

Since link state advertisement occurs when randomly jumping over one (or several) advertisement threshold, it can be regarded as an information theoretical source denoted by $\kappa_{(u,v)}$ with values $t_0, t_1, \ldots, 1_{Z-1}$ covering the axis of the link descriptor with probabilities $\hat{p}_0, \hat{p}_1, \ldots, \hat{p}_{Z-1}$. The source does not emit any symbol (inactive) with probability $\hat{P} = 1 - \sum_{i=0}^{Z-1} \hat{p}_i$. These probabilities are governed by the randomly fluctuating link traffic (i.e. by the varying queue length in the buffer as each link is perceived as a buffer). Assuming optimal source coding, the necessary bandwidth to distribute the link state information is related to the conditional entropy $\mathrm{H}\left(\kappa_{(u,v)} \middle| \chi_{(u,v)} = 1\right)$ of $\kappa_{(u,v)}$, where the Bernoulli random variable $\chi_{(u,v)}$ describes whether the source is active or not. The link entropy on the other hand is the measure which quantifies the uncertainty about state of link (u,v) if $[t_i, t_{i+1})$ interval was advertised for this link ($\nu_{(u,v)} = i$). Using these two quantities the bandwidth and the quality of the signaling processes can be kept at bay.

$$\text{link entropy: } H\left(\delta_{(u,v)} \middle| \nu_{(u,v)}\right)$$
$$\text{signaling entropy: } H\left(\kappa_{(u,v)} \middle| \chi_{(u,v)}\right)$$

THESIS I.1 (unicast routing with incomplete information by Gaussian approximation)**.** *I gave a mapping for the link descriptors under the condition that the link descriptors have normal distributions with parameters m and $\tilde{\sigma}_{(u,v)} = \sqrt{m_{(u,v)}}$ and also the LAS follows $m = \frac{t_{i+1}+t_i}{2}$ in Theorem 1*

Theorem 1. *If $\delta_{(u,v)}$ is a subject to a normal distribution with parameters $\tilde{\sigma}_{(u,v)} = \sqrt{m_{(u,v)}}$, then the solution of ARII*

$$\tilde{R} = \operatorname*{argmax}_{R \in \mathscr{R}^{s \to d}} \mathbb{P}\left(\sum_{(u,v) \in R} \delta_{(u,v)} < \mathrm{T}\right) \qquad \text{(2 revisited)}$$

is equivalent to minimizing the objective function

$$\tilde{R} = \operatorname*{argmin}_{R} \sum_{(u,v) \in R} m_{(u,v)} \qquad (2.2)$$

by using the Bellman-Ford algorithm in polynomial time.

Using these assumptions the ARII problem can be reduced to a deterministic traditional SPR.

THESIS I.2 (unicast routing with incomplete information by recursive path finder algorithm)**.** *I gave procedures that can find routes in a packet switched network which satisfy the required QoS parameter with a given probability in Algorithm 1 and Algorithm 2 (restating below).*

The algorithms are based on a transformation of the random link descriptors using the large deviation theory which is described in Theorem 2:

Theorem 2. *Using the logarithm of the moment generating function (log-moment generating function)*

$$\mu_{(u,v)}(s) = \ln \mathbb{E}\left(\exp\left(s\delta_{(u,v)}\right)\right) = \ln \int_{-\infty}^{\infty} \exp(sx)\, dF_{(u,v)}(x), \tag{2.3}$$

or in case of a discrete random variable

$$\mu_{(u,v)}(s) = \ln \mathbb{E}\left(\exp\left(s\delta_{(u,v)}\right)\right) = \ln \sum_{i=1}^{\infty} \exp(sx_i)\, p_i, \tag{2.4}$$

the solution of the ARII is equivalent with minimizing the objective function

$$\tilde{R} = \underset{R}{\operatorname{argmin}} \sum_{(u,v)\in R} \mu_{(u,v)}(\hat{s}) \tag{2.5}$$

where the optimal $\hat{s}$ parameter is

$$\hat{s} = \inf_{s} \sum_{(u,v)\in \tilde{R}} \mu_{(u,v)}(s) - sT. \tag{2.6}$$

Where the two algorithms are the following:

Algorithm 1 Exhaustive-s algorithm

Input: $G\left(V,E,\delta_{(u,v)},F_{(u,v)}(x)\right),src,dst$

Define a grid on the set of possible values of s denoted by $\mathscr{S} = \{s_i, s_i > 0, i = 1,...,M\}$.

for all $i = 1,\ldots,M$ **do**

Pick $s_i \in \mathscr{S}$ and perform path selection R_i by an SPR algorithm with link measures $\mu_{(u,v)}(s_i) := \ln \mathbb{E}\left(\exp\left(s_i \delta_{(u,v)}\right)\right)$.

Based on the selected path R_i determine

$$\hat{s}_i = \text{Solve}\left[\sum_{(u,v)\in\tilde{R}_i} \frac{d\mu_{(u,v)}(s)}{ds} = T, s\right], \tag{2.7}$$

and calculate the bound

$$B_i := \exp\left(\sum_{(u,v)\in R_i} \mu_{(u,v)}(\hat{s}_i) - \hat{s}_i T\right). \tag{2.8}$$

end for

Find the path which belongs to minimal bound

$$\tilde{R}_j : j = \operatorname*{argmin}_i B_i. \tag{2.9}$$

Output: $\tilde{R}_j$ chosen path between *src* and *dst*

Algorithm 2 The Recursive Path Finder - s Finder Algorithm

Input: $G\left(V,E,\delta_{(u,v)},F_{(u,v)}(x)\right),src,dst$

Pick $s \leftarrow$ a positive starting value and compute the path independent $\mu(s)$

repeat

Associate measure $\mu_{(u,v)}(s)$ to each link $(u,v) \in E$.

Perform the SPR algorithm to find the optimal path $\tilde{R}(s)$ for parameter s.

For the obtained $\tilde{R}$ determine $\tilde{s}$ by expression

$$\tilde{s} = \frac{\mu'^{-1}\left(T - \sum_{(u,v)\in R} a_{(u,v)}\right)}{|R|}.$$

$s \leftarrow \tilde{s}$.

until $\tilde{R}(\tilde{s}) \neq \tilde{R}(s)$

Output: $\tilde{R}(s)$ chosen path between *src* and *dst*

Performance analysis: In order to compare the algorithms I introduce a performance measure for comparing two paths R_a and R_e both starting from node *src* and ending at node dst. This is defined as the ratio of the probability of the path R_e (found by the exhaustive search) and the probability of the path R_a (found by a given algorithm) fulfilling the end-to-end QoS criterion, given as follows:

$$\eta(R_a,T) := \mathbb{P}\left(\sum_{(u,v)\in R_e} \delta_{(u,v)} < T\right) - \mathbb{P}\left(\sum_{(u,v)\in R_a} \delta_{(u,v)} < T\right), \tag{2.10}$$

Given that R_e is the best route in the sense that it fulfills any T QoS criterion with the largest probability, $\eta(R_a,T) \geq 0$, and it measures the “performance drop” in probability for given T QoS value.

The normalized area under these curves are defined as:

$$\chi(R_a) = \int_0^{T_{\max}} \eta(R_a,T)dT \Bigg/ \int_0^{T_{\max}} \mathbb{P}\left(\sum_{(u,v)\in R_e} \delta_{(u,v)} < T\right) dT \tag{2.11}$$

where $T_{\max}$ is the investigated maximum value for T. χ corresponds to the performance drop of a path compared to the best path. It is easy to see that the “worst path” R_0 which can fulfill any $T \leq T_{\max}$ only with 0 probability corresponds to the value $\chi(R_0) = 1$ and the best path R_e corresponds to $\chi(R_e) = 0$. The closer this function approximates the value 0, the better the performance of the corresponding route is.

Furthermore to obtain more general numerical results, the algorithms were run not only on one graph with particular link states, but on a set of graphs denoted by $\mathscr{G}$. Each graph had 10 nodes having cardinalities at least 3, and the random graph generator made sure that all nodes has belonged to the same component. Let us denote all possible routes (from all possible *src* to all possible *dst*, $src \neq dst$) in a graph G with $\mathscr{R}(G)$. I characterize the "ensemble" performance by metrics $\chi(G)$ and χ_e, given as follows:

$$\chi(G) := \frac{1}{|\mathscr{R}(G)|} \sum_{R\in\mathscr{R}(G)} \chi(R) \qquad \chi_e := \frac{1}{|\mathscr{G}|} \sum_{G\in\mathscr{G}} \chi(G) \tag{2.12}$$

The performance of four algorithms will be shown and compared to the performance of the exhaustive search: “OSPF”,”Gaussian approximation”,”exhaustive-s”,”recursive-s” The “OSPF” is a simple shortest path algorithm having a metric as the advertised LAS entry. The “Gaussian approximation” had its metric according to Theorem 1, the “exhaustive-s” is based on Algorithm 1 and “recursive-s” is based on Algorithm 2.

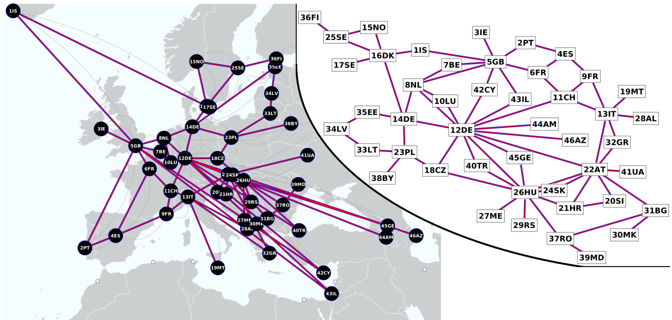


Figure 3: GEANT network topology used to evaluate the proposed algorithms.

On the GEANT topology I have chosen Island "1ISL" for the *src* node and Romania "37RO" for the *dst* node, as this pair offers a wide variety of routes in between them. All the routes from *src* to *dst* can be labeled with an index (1 to 4937) in an increasing order according to $\chi(R_a)$. This labeling can be seen in Figure 4 Using the fixed and

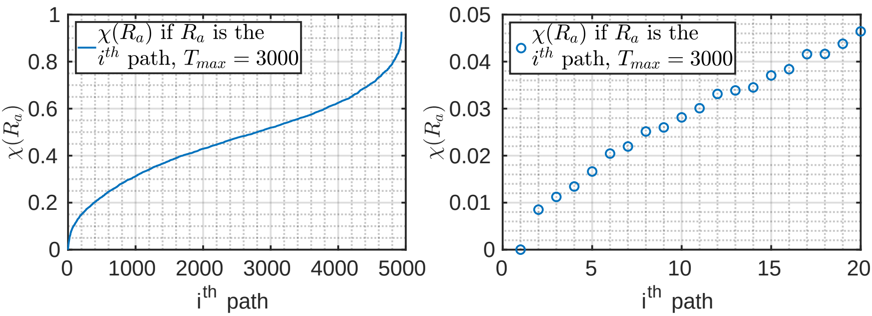


Figure 4: Path index vs its performance χ, T_{max}=3000. The left figure depicts all the paths, while the figure on the right zooms in to the first 20 best path.

known link descriptor distribution I sampled the network. When the OSPF algorithm found the best route (path #1) so did all other algorithms. For this particular link distribution ensemble the relative frequency for the OSPF not choosing path #1 was $P = 0.4148$, so almost half of the time there were better performing paths. To quantify the improvement of the introduced algorithms I chose 10000 sample points when the OSPF algorithm did not choose path #1. The frequency of the routes chosen by the algorithms are depicted in Figure 5. The total performance metric χ_e on the graph ensemble can be found in Figure 7.

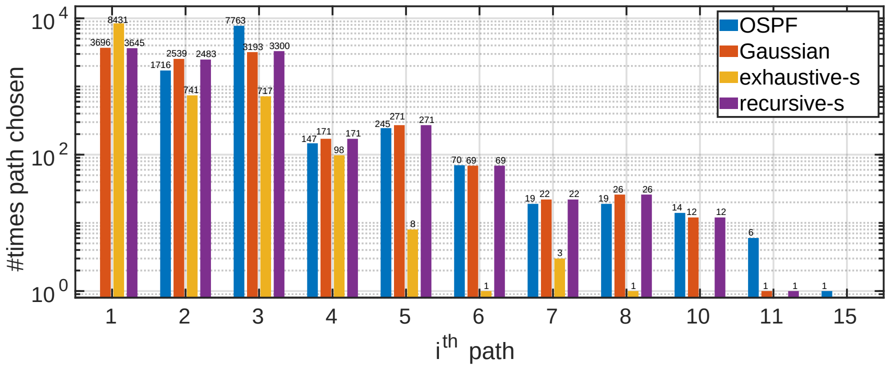


Figure 5: Path choice frequency out of 10^4 samples when the OSPF did not choose path #1 in the GEANT topology

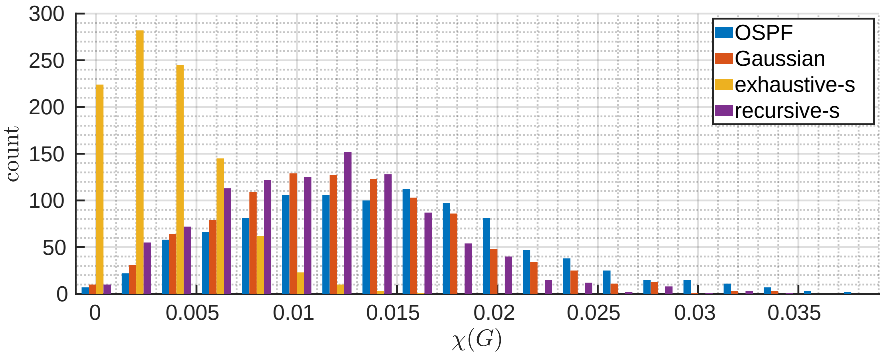


Figure 6: The distributions of the ensemble performance metric $\chi(G)$ over all graphs G in the random graph ensemble $\mathscr{G}$ for all for all introduced algorithms.

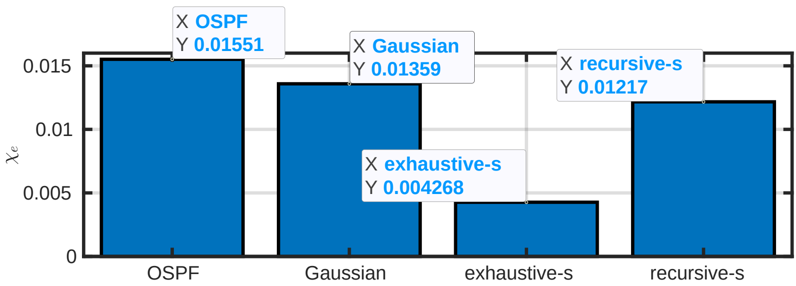


Figure 7: The total ensemble performance metric χ_e for all introduced algorithms.

THESIS I.3 (multicast routing with incomplete information with HNN)**.** *I defined algorithm to find a sub-optimal solution to the multicast routing problem with random link descriptors in Algorithm 3 (restating):*

Algorithm 3 Find optimal tree for end-to-end requirement

Input: $G\left(V, E, \delta_{(u,v)}, F_{(u,v)}(x)\right)$, $\kappa = 1$, $\mathrm{T} > 1$ *src*,*m*
repeat
  A = find tree with HNN(G, κ, T)
  if A is found **then**
    decrease κ
  else
    increase κ
  end if
until no significant increase in performance
Output: A is the multicast tree between *src* and *m*

The procedure transforms the random link descriptors into deterministic ones by using results from large deviation theory, which I formulated:

$$\tilde{\mathrm{A}}_2 : \underset{\mathrm{A}\in\mathscr{A}}{\operatorname{argmin}} \sum_{(u,v)\in\mathrm{A}} C_{uv},$$

$$s.t. \quad \mu_{R_{src,m}}(s) < \ln(\kappa) + s\cdot\mathrm{T},$$

The transformed problem can be seen as a CGSMT, which is still NP-hard, but I propose a sub-optimal solution by using HNN, where the corresponding parameters are described at sub-subsection 2.4.3.

$$\mathscr{E}(\mathbf{y}) = \alpha_1\left(2\mathbf{y}^{tr}\mathbf{b}^{(1)}\right) + \alpha_2\left(\mathbf{y}^{tr}\mathbf{W}^{(2)}\mathbf{y} + 2\mathbf{y}^{tr}\mathbf{b}^{(2)}\right),$$

Performance analysis: The $\tilde{\mathrm{A}}_1$ and the $\tilde{\mathrm{A}}_2$ objective functions were evaluated by exhaustive search and the HNN based algorithm on a graph with the following parameters: The size of the network $N = 8$, the Rayleigh channel parameters were chosen to typical or better indoor environment: $\gamma = 3, g = 1, \theta = 10, s^2 = 1$. The positions of the nodes were randomly generated according to i.i.d. uniform distributions in the unit square. The group of the multicast nodes consisted 3 randomly chosen nodes.

I have performed the exhaustive search by enumerating all the possible trees and evaluating the objective functions on the trees. I have compared the results of the HNN algorithm to the exhaustive solution. For the $\tilde{\mathrm{A}}_2$ objective function I have evaluated the performance given by the Chernoff bound and also the corresponding theoretical probability by performing convolutions on the known distributions.

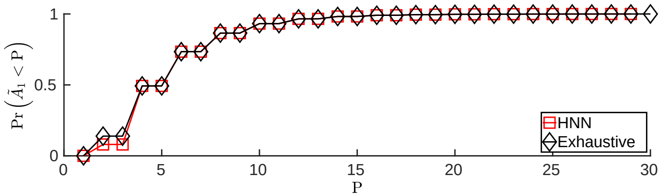
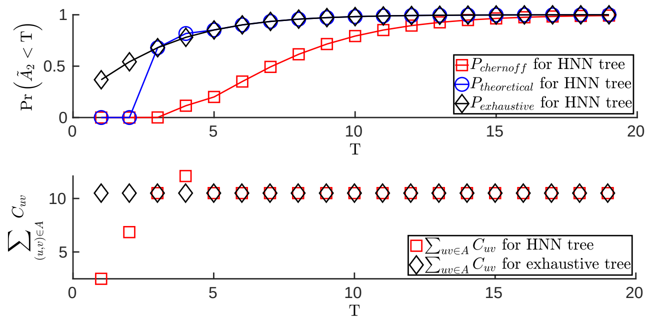
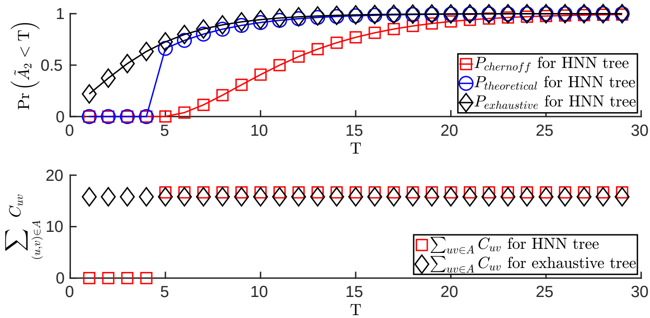


Figure 8: A typical evaluation of the Ã$_1$ objective function.



(a) Near optimal solution for Ã$_2$ obj. func.



(b) A typical evaluation of the Ã$_2$ obj. func.

Figure 9: Performance of the multicast tree finder algorithm in case of additive measures

THESIS I.4 (optimizing link scaling using MAP/M/1). *I formulated a constrained optimization problem which connects the information about the random link descriptors (Link Entropy) and the appropriate bandwidth of the signaling process to support that information (Signaling Entropy) at a certain probability.*

$$\min_{\Delta t} \mathrm{H}\left(\delta_{(u,v)} \middle| \nu_{(u,v)}(\Delta t)\right)$$
$$s.t.\ \ \mathrm{H}\left(\kappa_{(u,v)}(\Delta t) \middle| \chi_{(u,v)}(\Delta t)\right) < \alpha$$

I proposed a computable solution to this problem by modeling the dynamics of the link descriptors as MAP/M/1 described as:

$$\mathrm{H}\left(\kappa_{(u,v)}\left(\boldsymbol{\pi}, \mathbf{D0}, \mathbf{D1}, \mu, \Delta t\right) \middle| \chi_{(u,v)}(\Delta t)\right)$$
$$\mathrm{H}\left(\delta_{(u,v)}\left(\boldsymbol{\pi}, \mathbf{D0}, \mathbf{D1}, \mu\right) \middle| \nu_{(u,v)}\left(\boldsymbol{\pi}, \mathbf{D0}, \mathbf{D1}, \mu, \Delta t\right)\right)$$

Consequently the information theoretical quantities can be obtained analytically and the optimal solution can be found.

Performance analysis: I have based my traffic models on the publicly available DISCO data-set from the Measurement-lab data-set [34]. I choose the switch connected to the "mlab1.dub01.measurement-lab.org" server. Since June 2016, M-Lab has collected high resolution switch telemetry for each M-Lab server and site uplink. I have used the "Bytes received by the machine switch port" and "Unicast packets received by the machine switch port" to gather the statistics for the traffic models. The data-set contains these metrics with sampling time of 10s. I assumed that within

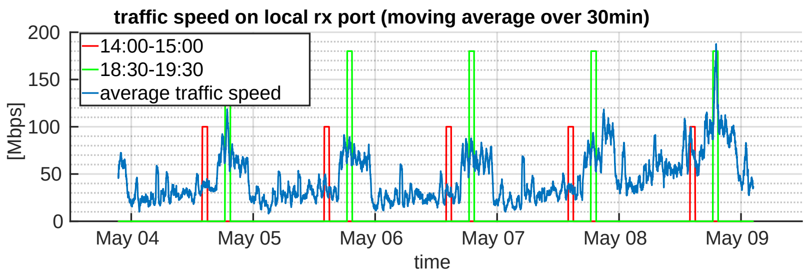


Figure 10: Daily self similarity pattern in the data from mlab1.dub01.measurement-lab.org. Moving averaged data was plotted from 2018. May 4-9.

an hour interval the traffic somewhat stays the same (relative to the daily regular fluctuations). Based on this I have chosen two time periods over which I have aggregated the necessary statistics to derive the traffic models. An average traffic load:

from 14:00-15:00 each day in 2018. May. 1. to Jun. 30. Traffic situation 1 has a mean packet rate of 6066.341pps, std:6865.61pps, max:75944.03pps and mean speed 62Mbps, std:78.28Mbps, max:883.66Mbps.
A more intensive traffic load: from 18:30-19:30 each day in 2018. Sep. 1. to Nov 30. Traffic situation 2 has a mean packet rate of 13613.81pps, std:12278.3pps, max:137180.7pps and mean speed 143.5Mbps, std:141.5Mbps, max:1605.45Mbps.

I have used 16 state MAP-s in both cases to model the sequence of events. From the identified models I also generated sequences of events, which then were mapped back to the average packet rate metric for comparison. It can be seen that the identified models are detailed enough to reproduce the statistics of the real life data. On Figure 13 and Figure 14 one can see the SE and the LE plotted against the link scaling. Here the link scaling means the number of divisions over the link descriptor.

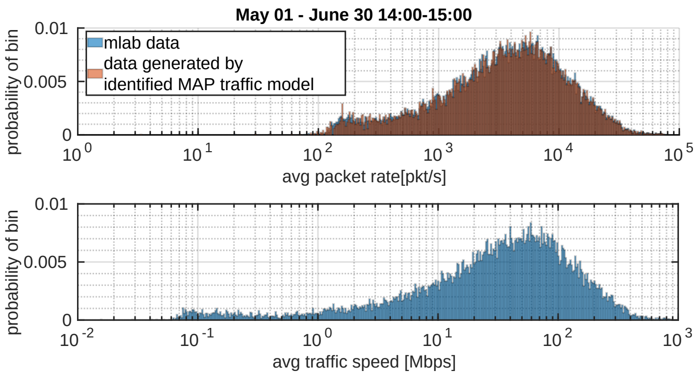


Figure 11: Traffic statistics for 14:00-15:00 each day in 2018. May. 1. to Jun. 30

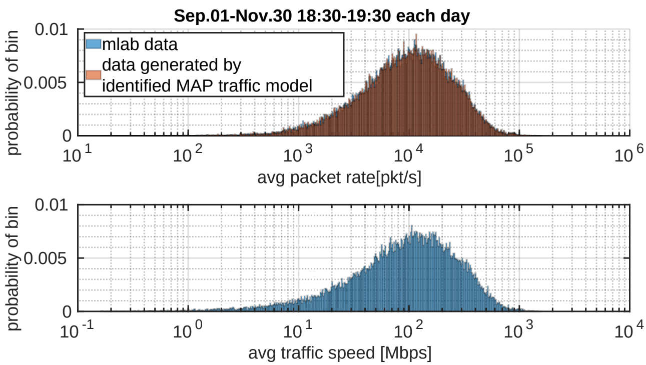


Figure 12: Traffic statistics for 18:30-19:30 each day in 2018. Sep. 1. to Nov 30

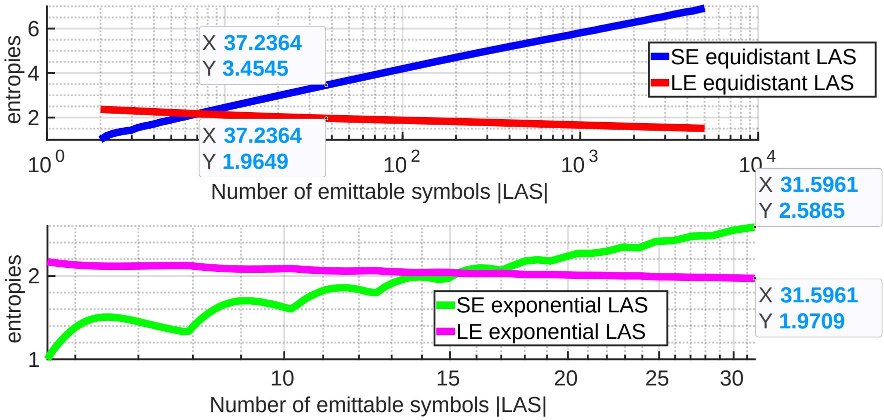


Figure 13: Information theoretic metrics for traffic situation 1

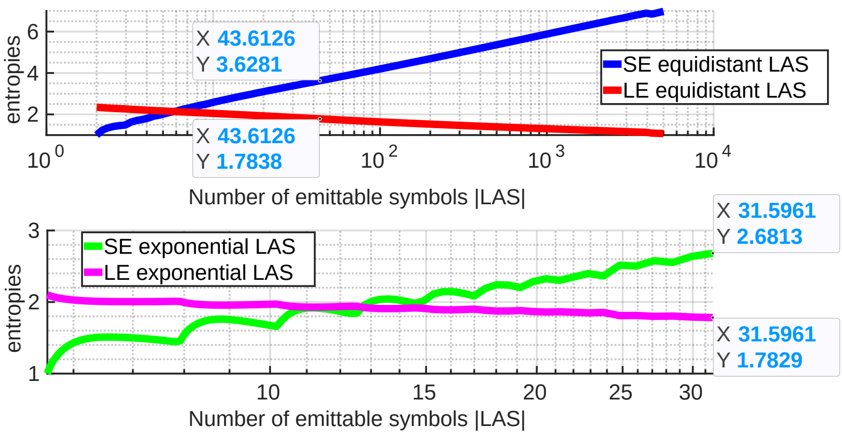


Figure 14: Information theoretic metrics for traffic situation 2

Thesis group II - a heuristic solver based on hypergraphs for UBQP and its applicability in ICT

This chapter is organized around the combinatorial approach and tractability of several ICT problems, since they can be formulated as an Unconstrained Binary Quadratic Programming (UBQP). These applications include load balancing, a wide class of scheduling problems, MUD, VLSI design, Steiner tree problems...etc. A survey of these applications can be found in [22].

In cloud computing environments and in IoT the efficient scheduling [30, 52] and distribution of tasks plays a central role in performance and scalability. Several approaches exist [41, 51, 44] - usually metaheuristics - to address these problems but at their core almost each of them contains a method for approximating a solution of a constrained optimization problem. This core step can be usually formulated as a UBQP, therefore the proposed algorithms can be utilized on it. Recent surveys on scheduling in IaaS cloud computing environments and load balancing can be found in [41, 51, 44]

Furthermore, other problems under linear constraints can also be transformed into UBQP as demonstrated in [21, 6, 35, 50, 10, 29]. Unfortunately, UBQP has proved to be NP-hard [12], but in some special cases it can be solved in polynomial time [6, 36, 37, 4]. In general though, there is still a great need for developing fast methods which can reach near-optimal solutions when the size of the problem goes beyond a given limit. Thus the aim of this chapter is to present some novel approaches to UBQP which are based on recursive dimension reduction (or addition) techniques. Although the more complex applications are more relevant, the proposed algorithms and their performance will be presented in detail on simpler applications for traceability:

- large scale problems listed in ORLIB.
- simple scheduling;
- Multiuser Detection;

Based on the performance analysis the new algorithms prove to be superior to the known heuristics regarding both the quality of the achieved solution and the convergence time. *The corresponding publication of the author is titled "Novel algorithms for quadratic programming by using hypergraph representations" [47].*

The UBQP is a quadratic COP where each component of vector $\mathbf{y}$ can have two distinct values, which are taken to be -1 and +1.

$$\begin{aligned}
&\mathscr{L}(\mathbf{y},\mathbf{W},\mathbf{b}) := \mathbf{y}^T\mathbf{W}\mathbf{y} - 2\mathbf{y}^T\mathbf{b},\\
&\mathbf{y} \in \{\pm 1\}^N, \mathbf{b} \in \mathbb{R}^N, \mathbf{W} \in \mathbb{R}^{N\times N}\\
&\mathbf{y}_{opt} = \min_{\mathbf{y}\in\{\pm 1\}^N} \mathscr{L}(\mathbf{y},\mathbf{W},\mathbf{b})
\end{aligned}$$

The state space of the N dimension UBQP problem can be represented by a weighted graph, where the vertices of the graph correspond to the state vector $\mathbf{y}$, the weights of the vertices are the values of the objective function $\mathscr{L}(\mathbf{y}, \mathbf{W}, \mathbf{b})$, while the edges between the vertices are defined by a given neighborhood function.

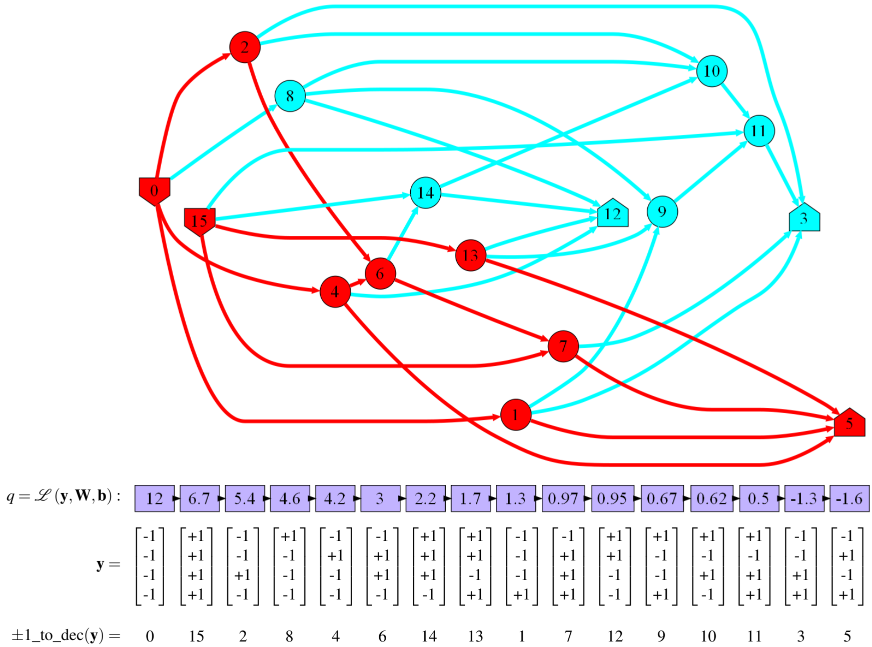


Figure 15: Search space of a 4 dimension UBQP - vertices of a 4 dimension hypercube

We can also extend the graph to a hypergraph based representation of the problem. In this the search space is extended by including all possible points from the subspaces. The extension of the search space on one hand is motivated with the reduced run time, i.e. searching for a candidate solution is performed in a smaller space, and on the other hand since we reduced the dimension we can get out from certain local minima. The new sub space points will be the intersections of the cutting plane with the edges of the hypercube.

If we do this in all possible combination we get 2^N sub spaces for search. One can connect these sub spaces with a hypergraph, where the vertices of this hypergraph are the graphs defined over the sub spaces of the original problem. In the following picture a hypergraph corresponding to the previous example is shown.

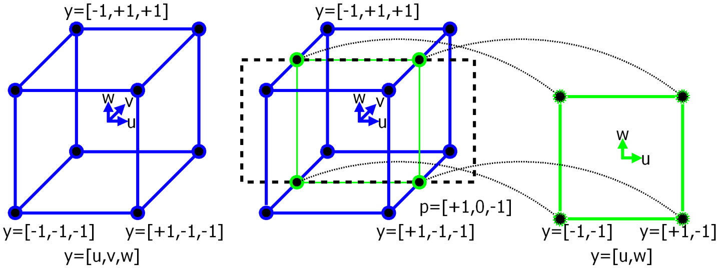


Figure 16: discarding the 2nd dimension from the 3 dimension hypercube

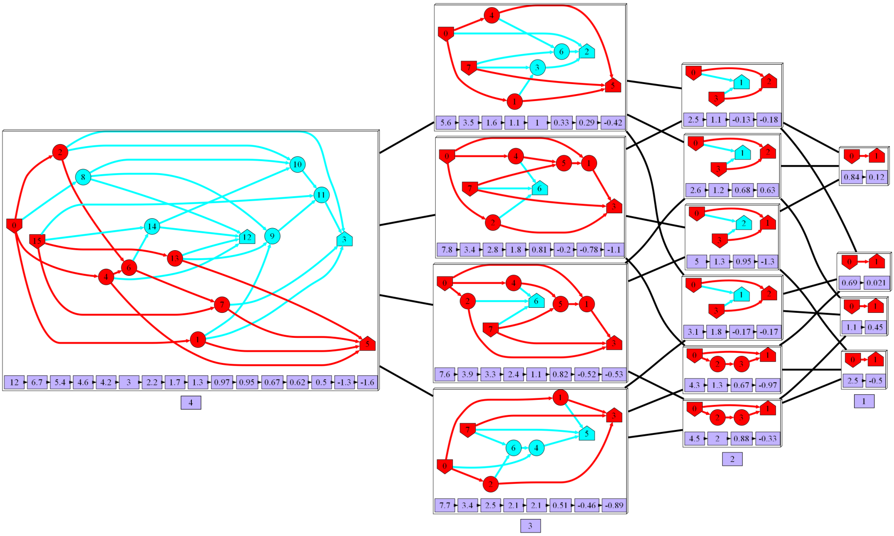


Figure 17: A hypergraph of a 4 dimension UBQP

THESIS II.1 (A heuristic solver family based on hypergraphs for UBQP)**.** *In Algorithm 4, I have given a hypergraph based, easily parallelizable algorithm family to sub-optimally solve the UBQP problem.*

The algorithms project the original search space into a hypergraph representation and use a HNN based internal solver to find a solution. I have given four instances of which two employs dimension reduction and two dimension addition. Table 3 (restating) summarizes the operation modes of the instances. (The precise description of the algorithms can be found in Appendix.

Table 3 Categorization of the algorithms

	greedy	opportunistic
dim. reducer	L01	D01
dim. adder	DA02	DA01

I have tested the performance on three different problem sets: on the standard ORLIB UBQP benchmark set, on a scheduling problem, and on a simulated MUD problem. I have shown that the proposed methods perform near optimal on the investigated ICT problems.

Algorithm 4 Pseudo code of the general UBQP solver algorithm

function INNER_SOLVER($\mathbf{W} \in \mathbb{R}^{k\times k}, \mathbf{b} \in \mathbb{R}^{k}, \mathbf{y}(\text{init}) \in \{\pm 1\}^{k}$)
    an arbitrary UBQP minimizer
    return $\mathbf{y} \in \{\pm 1\}^{k}$
end function
function $\Psi(u \in V_H, \mathbf{y} \in u_V)$
    choose $\widehat{u} \in V_H$ ▷ choose the next hypernode and
    choose $\widehat{\mathbf{y}} \in \widehat{u}_V$ ▷ choose a state in that hypernode
    return $\widehat{u}, \widehat{\mathbf{y}}$
end function

Input: $\widehat{\mathbf{W}}, \widehat{\mathbf{b}}$ and $u(\text{init})$ ▷ the problem and the starting hypernode
$\widehat{u} \leftarrow u(\text{init}) \in V_H$ ▷ start hypernode of the alg
choose $\widehat{\mathbf{y}} \in u(\text{init})_V$ ▷ init state in the hypernode
repeat
    define $L(\mathbf{W}, \mathbf{b}, \mathbf{y})$ objective function
    $u \leftarrow \widehat{u}$ and $\mathbf{y} \leftarrow \widehat{\mathbf{y}}$
    $\mathbf{W}, \mathbf{b} \leftarrow$ parameters from $u = G(V, E, Q(\mathbf{W}, \mathbf{b}))$
    if SHOULD_EMPLOY_INNER_SOLVER( )**then**
        $\mathbf{y}^{*} \leftarrow$ INNER_SOLVER($\mathbf{W}, \mathbf{b}, \mathbf{y}$)
    else
        $\mathbf{y}^{*} \leftarrow \mathbf{y}$
    end if
    $\widehat{u}, \widehat{\mathbf{y}} \leftarrow \Psi(u, \mathbf{y}^{*})$
until STOP_CRIT( )
Output: $\widehat{\mathbf{y}}$ ▷ the best solution found by the alg.

Performance analysis: For the ORLIB [5] test set we have implemented all the algorithms in the same programming environment and performed the tests within the same computational environment as well. Besides the new algorithms we also use the following traditional algorithms for the sake of comparison:

- "HNN" - a discrete time and discrete valued Hopfield type recurrent network.
- "1-opt LS" - a 1-opt local search type algorithm.
- "BLS" an algorithm presented by Beasley [6] based on a 1-opt local search method.
- "BTS" a taboo search algorithm also presented by Beasley [6].
- "DDT" the well-known DDT algorithm by Boros, Hammer, and Sun [7]
- "SDR" an SDR type algorithm without randomization presented by Luo et al. [29].
- "SDR with randomization" is the same algorithm but with randomization [29].

Table 4 Performance comparison of BTS vs DA01

		$\mathbb{E}$solution		best solution		relative freq of best solution		mean run time		mean run time until best solution found	
	prob\alg	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01
dim=50 K=10000	1	***2160***	2055,4	***2160***	***2160***	1	0,1052	232,23	1	232,2286	***9,5057***
	2	***3658***	3588,6	***3658***	***3658***	1	0,1316	251,05	1	251,0545	***7,59878***
	3	4650	***4680,4***	4650	***4778***	1	0,4924	240,36	1	240,3645	***2,03087***
	4	***3472***	3400,7	***3472***	***3472***	1	0,1275	234,53	1	234,5311	***7,84314***
	5	***4152***	4098,5	***4152***	***4152***	1	0,7053	234,99	1	234,9933	***1,41784***
	6	***3842***	3823,6	***3842***	***3842***	1	0,5405	239,71	1	239,7059	***1,85014***
	7	***4588***	4535,1	***4588***	***4588***	1	0,2746	241,11	1	241,107	***3,64166***
	8	***4222***	4195,2	***4222***	***4222***	1	0,8435	233,71	1	233,7129	***1,18554***
	9	***3862***	3829,9	***3862***	***3862***	1	0,6273	228,99	1	228,9912	***1,59413***
	10	***3496***	3450,8	***3496***	***3496***	1	0,2163	236,26	1	236,264	***4,62321***
	prob\alg	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01
dim=100 K=10000	1	***7910***	7631	***7910***	***7910***	1	0,0039	65,814	1	***65,8142***	256,41
	2	***11178***	11030	***11178***	***11178***	1	0,0192	75,619	1	75,61899	***52,0833***
	3	***12956***	12875	***12956***	***12956***	1	0,1854	73,601	1	73,60066	***5,39374***
	4	***10606***	10493	***10606***	***10606***	1	0,0722	70,157	1	70,15745	***13,8504***
	5	***8994***	8777,2	8994	***8996***	1	0,0173	70,659	1	70,6588	***57,8035***
	6	***10470***	10362	10470	***10486***	1	0,011	71,521	1	***71,5213***	90,9091
	7	***9980***	9877,6	9980	***10030***	1	0,0182	72,419	1	72,41868	***54,9451***
	8	***11380***	11240	***11380***	***11380***	1	0,1357	71,742	1	71,74218	***7,3692***
	9	***11340***	11246	***11340***	***11340***	1	0,269	79,918	1	79,91803	***3,71747***
	10	***12438***	12348	***12438***	***12438***	1	0,032	79,959	1	79,95852	***31,25***
	prob\alg	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01	BTS	DA01
dim=500 K=1000	1	***116526***	114780	116526	***116532***	1	0,001	13,642	1	***13,6418***	1000
	2	***128678***	127801	***128678***	***128678***	1	0,002	14,043	1	***14,0425***	500
	3	***131084***	130013	***131084***	***131084***	1	0,006	13,294	1	***13,2944***	166,667
	4	***129794***	128646	***129794***	129784	1	0,001	13,209	1	***13,2092***	1000
	5	***125008***	123859	125008	***125062***	1	0,001	11,694	1	***11,6937***	1000
	6	***121868***	120189	***121868***	***121868***	1	0,001	13,336	1	***13,3355***	1000
	7	***122730***	121163	122730	***122756***	1	0,001	13,989	1	***13,989***	1000
	8	***123454***	121958	***123454***	123428	1	0,001	13,811	1	***13,8106***	1000
	9	***121622***	120026	121622	***121668***	1	0,001	13,889	1	***13,8885***	1000
	10	***130900***	130219	130900	***131374***	1	0,01	13,865	1	***13,8647***	100

When applying the algorithms to the scheduling problem we compared to the traditional Earliest Deadline first, Weighted Shortest Processing Time first and its wariants. The Total Weighted Tardiness is displayed as the performance measure.

In this case the best TWT values achieved by the different algorithms are: HNN:84,

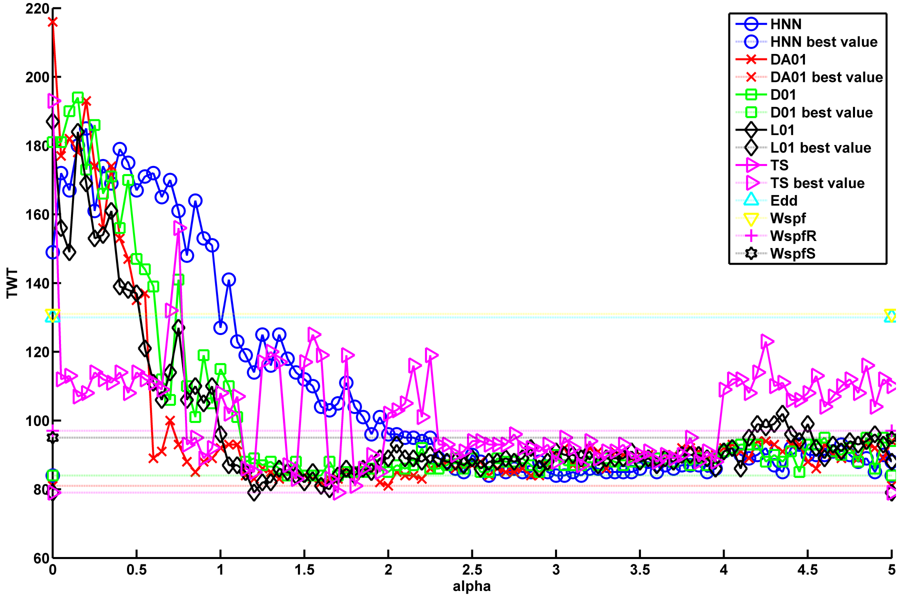


Figure 18: TWT performance of algorithms versus heuristic parameter α for a specific case $J = 10$

DA01:81, D01:84, L01:79, TS:79, EDD:130, WSPT:131 WSPTR:97, WSPTS:95, respectively. For this specific case, L01 and Taboo Search perform the best, but their performances are nearly identical with the others.

For the Multi User Detection application the following detectors were used [25]: "threshold" is the plain sign decision rule following the matched filter which filters the symbols with the appropriate code. The "invFilter" is the generalized Zero-Forcing detector trying to equalize the channel and cancel the multi user interference. The "QR" and the "DF" variant of this algorithm uses the QR factorization and reformulation of the problem, and a decision feedback respectively. The "MMSE" is the minimum mean square detector and the sphere detector is denoted with the abbreviation "SD". In the following figures the performance measures are shown for an unsaturated multi user configuration. We used $M = 7$ users to communicate simultaneously. It can be seen that two of our solvers perform as well as the Sphere Detector and the other traditional methods perform very poorly under this condition. From the other figure it can be seen that in terms of objective function value for almost all SNR values the "D01" algorithm and the "SD" performs best, but the "DA01" algorithm performs almost identically to them. Referring back to the previous chapter we recall that "DA01" needs much smaller time to reach its candidate solution than the "D01".

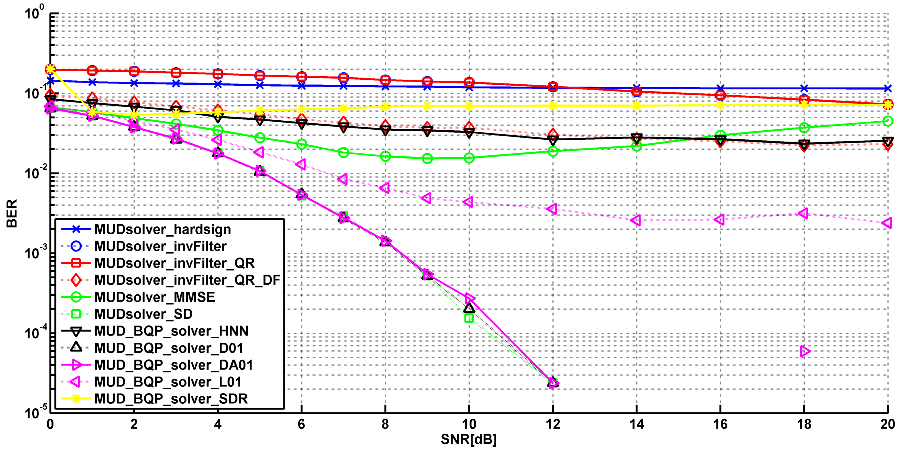


Figure 19: BER performance of the algorithms for 7 users

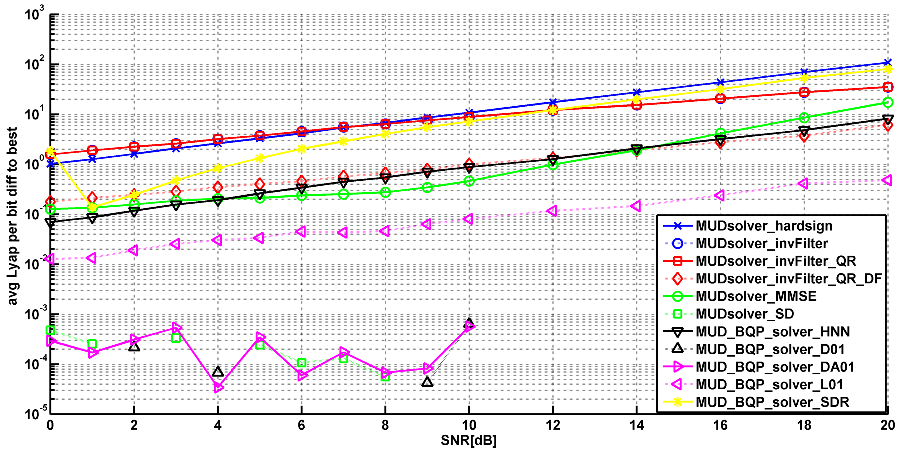


Figure 20: Objective function performance of the algorithms for 7 users

The sphere detector gives a very good quality solution, however it converges much more slowly than the DA01. Furthermore, our proposed algorithms lend themselves to easy parallel implementation.

Thesis group III - near Bayesian performance non-parametric detection with Feed Forward Neural Networks

This thesis group elaborates on developing novel encoding techniques for implementing non-parametric, neural network based detectors for pattern recognition on noisy input. This fundamental problem appears in many real world applications like in information extraction in big data context, automated surveillance, speech recognition, content based search or in legacy systems using CDMA to name a few.

I propose an FFNN based algorithm which I present on the MUD problem, but due to the nature of the method it can be easily generalized. In the MUD scenario it is capable of achieving near optimal performance with relatively limited complexity. These new encoding methods on the one hand can increase the processing speed and reduce the complexity of the FFNN based detector, on the other. Furthermore, we demonstrate that an asymptotically optimal detection performance can be achieved by the proposed algorithms. Due to the increased processing rate, the new scheme may further improve SE. Extensive simulations and the corresponding numerical analysis demonstrate that the proposed algorithms yield near optimal performance on real channel models (COST-207). *The corresponding publication of the author is "Multi-user detection using non-parametric Bayesian estimation by feed forward neural networks" [46].*

In the most general case the optimal decision after receiving a sequence $\mathbf{x}$ is symbol $\hat{\mathbf{y}}$ which is the most probable that had been sent through the channel, as it will minimize the BER [49, 10]. This is also called the Bayesian or MAP decision, which is given as follows:

$$\hat{\mathbf{y}}_{opt} = f_{opt}(\mathbf{x}) = \underset{\mathbf{y}\in\{\pm 1\}^L}{\operatorname{argmax}} \mathbb{P}\left(Y = \mathbf{y} | X = \mathbf{x}\right), \tag{2.13}$$

In other words the MAP decision can be carried out by searching for the maximum among the components of this vector $\mathbf{p}(\mathbf{y}|\mathbf{x})$. Please also note that computing this probability vector directly is also of exponential complexity, since $N = 2^L$.

$$\hat{\mathbf{y}}_{opt} = \mathbf{y}^{(i)} : \quad i = \underset{n\in 1\ldots N}{\operatorname{argmax}}\, p(\mathbf{y}^{(n)}|\mathbf{x}) \tag{2.14}$$

The block diagram of this exponential complexity optimal detector can be seen in Figure 21

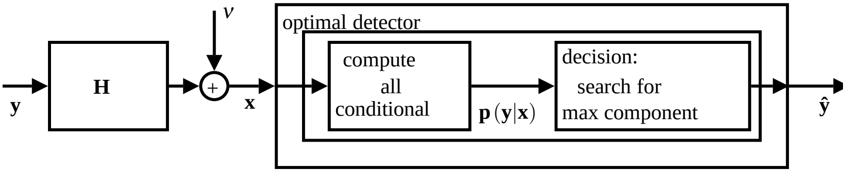


Figure 21: Flow graph representation of the optimal detector

THESIS III.1 (blind detection by interval halving and FFNN)**.** *I have defined an FFNN based blind detector for the MUD problem, which lends itself to easy parallel-*

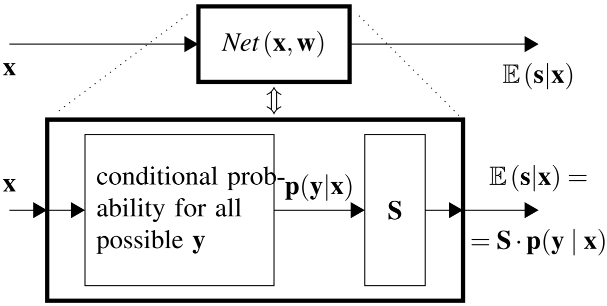


Figure 22: Equivalence of the FFNN with an encoding

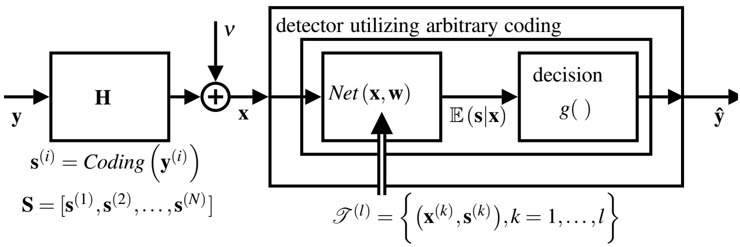


Figure 23: flow graph representation of the detector using an arbitrary encoding

ization and can perform optimally under the constraint

$$
\begin{aligned}
& j = \max_{i \in 1 \ldots N} p(\mathbf{s}^{(i)}|\mathbf{x}) \\
& \textit{if } j \in E \\
& \sum_{k \in E} p(\mathbf{s}^{(k)}|\mathbf{x}) > \sum_{i \in \{1 \ldots N\} \setminus E} p(\mathbf{s}^{(i)}|\mathbf{x}).
\end{aligned}
$$

In the following I give the linear encoding based on interval halving which is used to generate a training set for an FFNN

$$
S_{i,j} = s_i^{(j)} = \operatorname{sgn}\left(-\sin\left(2\pi \cdot 2^{(i-1)} \cdot \frac{j}{N+1}\right)\right), \; i = 1 \ldots L, \; j = 1 \ldots N
$$

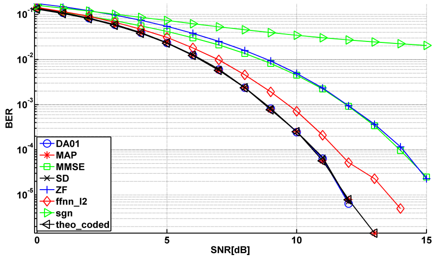
and the low complexity decision function which is to be employed on the output of the net is

$$
\hat{\mathbf{y}} = \operatorname{sgn}\left(\mathbb{E}\left(\mathbf{s}|\mathbf{x}\right)\right).
$$

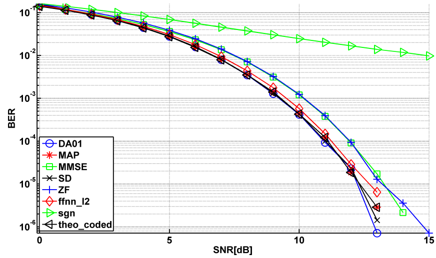
I have shown that the detector performs near optimally on the investigated MUD scenarios.

Performance analysis: For comparison, beside the new methods I also show a few traditional detector (MMSE, ZF, SDR) and also the theoretical limit. The channel model for each user was computed based on the COST 207 models [11]. Four models were used, namely "COST207 Hilly Terrain 6 tap alternative", "COST207 Rural Area 6 tap", "COST207 Typical Urban 12 tap" and "COST207 Bad Urban 12 tap" models.

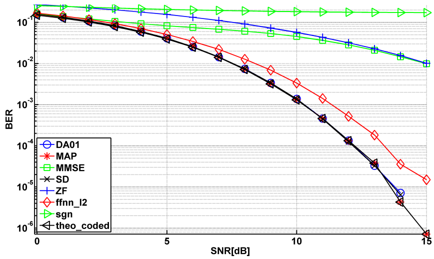
The following figures show the performance of the methods for $M = 7$, $K = 2$ and $L = 14$ for different radio channel models. One can see, when the channel changes from the one measured on hilly terrain to another one measured in urban area, practically can achieve the optimal performance. The difference between the theoretical performance and the actual one is due to the asymptotic nature of learning. Namely the FFNN failed to capture the conditional expected value perfectly. However, it achieves similar BER than the best performing SD. Note that while the SD is a parametric detection which needs the channel characteristics as opposed to our proposed method which does not. One can see that "FFNN I2" produces only slightly worse performance than the MAP decision. But even in this case the novel method provide low BER with respect to SNR.



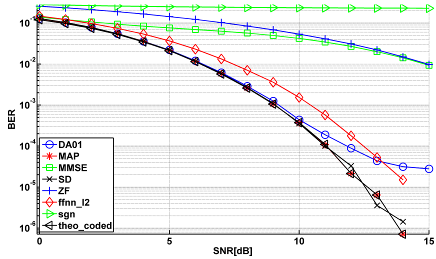
(a) BER using channel COST 207 Hilly Terrain 6 tap alternative



(b) BER using channel COST 207 Rural Area 6 tap

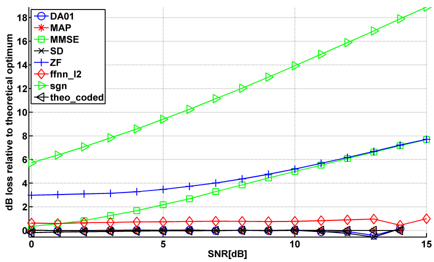


(c) BER using channel COST 207 Typical Urban 12 tap

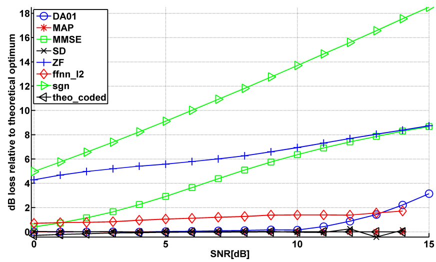


(d) BER using channel COST 207 Bad Urban 12 tap

Figure 24: Performance curves with parameter $L = 14$ for the four channel models



(a) SNR loss curves with parameter $L = 10$ for channel COST 207 Bad Urban 12 tap channel



(b) SNR loss curves with parameter $L = 14$ for channel COST 207 Bad Urban 12 tap channel

Figure 25: SNR loss curves for the Bad Urban channel model

3 Summary of the dissertation and closing remarks

In this dissertation I have given the following answers to the posed questions:

One can efficiently find an appropriate path or tree in a packet switched network which provides a QoS (either bottleneck or additive type). This can be achieved by exploiting the statistical properties of the traffic and transforming the models in such a way, that they become a natural fit for the traditional route finding algorithms or neural networks. Furthermore the precision with which the QoS is met can be scaled at the expense of some bandwidth by applying information theoretic measures.

One can solve efficiently and near optimally the UBQP problem which is present in relevant ICT applications: the scheduling tasks in communication networks (IoT or cloud computing environments), load balancing and MUD for wireless technologies. This problem can be treated in a parallel fashion with the aid of both Feed Forward and Recurrent type neural networks. To achieve this I have demonstrated how to reformulate the problems to fit these algorithms. For the Recurrent type neural network I posed these problems as an "energy based" optimization problem. For the Feed Forward neural networks, I exploited their general approximation capabilities.

One can solve efficiently and near optimally a general pattern recognition problem with the aid of Feed Forward neural networks and a linear encoding technique. I have demonstrated the efficiency of the algorithm on the MUD problem, but it is applicable on a wide range of problems including automated surveillance applications, content based search, speech recognition.

The numerical examples presented, back up my conjecture that these algorithms are in deed applicable and perform efficiently.

Although a lot of aspects were not addressed, this work gives sufficient details, such that it can be used as a basis for further investigation. For example how a physical implementation of such neural networks could speed up finding sub-optimal solutions for these problems. Furthermore it gives a common numerical reference for comparison to other types of algorithms. A possible natural extension of the proposed methods would be to change the currently used neural networks with "deep-learning" based variants, compare the performance and investigate the gains and losses. Certainly if a particular application is to be considered, these algorithms need tailoring. Also note that the appropriate physical architecture may not exist at the time of writing but this work might point to such possible directions.

4 Acknowledgement

First I would like to give thanks to my supervisor *Dr. János Levendovszky*, to †*prof. Tamás Roska*, to *Dr. Árpád Csurgay*, to *Judit Nyékyné dr. Gaizler*, to *Dr. Péter Szolgay* and to all the present and past directors of the doctoral school, to *Martin Haenggi* and *prof. Géza Kolumbán* for their encouragement, advises and critics of high standards which all modulated the trajectory I traveled so far.

I would like to express my special gratitude to *dr. András Oláh* and to *Péter Vizi*, *dr. Kornél Németh*, *János Pásztor*, *József Somogyi*, *Dániel Süttő* and *Dávid Kauker* for their support, endurance and help, without which this work would not have been completed.

I would like to thank my fellow doctoral school members and co-workers for their community and support, inter alia *András Bojárszky*, *dr. Gergely Treplán*, *dr. Kálmán Tornai*, *dr. Csaba Józsa*, *dr. Reguly István*, *Miklós Koller*, *Imre Juhász*, *Gábor Abonyi*, *Péter Pál Porázik*, *Tamás Kosztolánczi*, *Attila Fehér*, *Marcell Tibély*, *Máté Nagy*, *Ferenc Varjasi*, *Alexander Dvorzsák*, *Milán Györki*.

I would like to give thanks to *Katinka Vida Tivadarné* for her precise and attentive support throughout my years in the doctoral school and beyond.

To the members of the finance and registrars department, the members of the deans office, the members of the IT department and for all fellow members of the faculty for their work, which are often not as visible but equally important.

I would also like to give thanks to the members and founders of the *SP CEE Scholarship* program for their friendship, guidance, patience and financial support.

And I am deeply grateful to my parents, brothers and to the other members of my family: *Kálmán Tisza*, †*Kálmánné Tisza* born *Katalin Adorján*, *Kálmán* and *Levente*, †*Gergelyné Busa Ildikó*, also to my magisters: †*János Gutai*, †*Mária Kovaliczky* and teachers: namely *Katalin Szabó Kálmánné* and *Erika Szeitzné Viski* for building the foundations on which I could set a firm stand.

References

[1] *802.3-2012*. 2012. DOI: 10.1109/ieeestd.2012.6419735. URL: http://dx.doi.org/10.1109/IEEESTD.2012.6419735.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "Wireless sensor networks: a survey". In: *Computer Networks* 38.4 (2002), pp. 393 –422. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/S1389-1286(01)00302-4. URL: http://www.sciencedirect.com/science/article/pii/S1389128601003024.

[3] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda. *QoS Routing Mechanisms and OSPF Extensions*. RFC 2676 (Experimental). 1999. URL: http://www.ietf.org/rfc/rfc2676.txt.

[4] F. Barahona. "A solvable case of quadratic 0-1 programming". In: *Discrete Applied Mathematics* 13.1 (1986), pp. 23–26.

[5] J. E. Beasley. "OR-Library: distributing test problems by electronic mail". In: *Journal of the Operational Research Society* 41.11 (1990), pp. 1069–1072.

[6] J.E. Beasley. "Heuristic algorithms for the unconstrained binary quadratic programming problem". In: *London, UK: Management School, Imperial College* (1998).

[7] E. Boros, P. Hammer, and X. Sun. "The DDT method for quadratic 0-1 minimization". In: *RUTCOR Research Center, RRR* 39 (1989), p. 89.

[8] Shigang Chen and Klara Nahrstedt. "An overview of quality of service routing for next-generation high-speed networks: problems and solutions". In: *IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video* 12.6 (Nov. 1998), pp. 64–79. ISSN: 0890-8044. DOI: 10.1109/65.752646.

[9] Shigang Chen and Klara Nahrstedt. "On finding multi-constrained paths". In: *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*. Vol. 2. IEEE. June 1998, pp. 874–879. DOI: 10.1109/ICC.1998.685137. URL: http://citeseer.nj.nec.com/chen98finding.html.

[10] M.O. Damen, H. El Gamal, and G. Caire. "On maximum-likelihood detection and the search for the closest lattice point". In: *Information Theory, IEEE Transactions on* 49.10 (2003), pp. 2389 –2402. ISSN: 0018-9448. DOI: 10.1109/TIT.2003.817444.

[11] M. Failli. *Digital land mobile radio communications COST 207*. Tech. rep. European Commission, 1989.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Vol. 174. New York, NY, USA: W.H.Freeman & Co Ltd, 1979. ISBN: 0716710447.

[13] M. Goguen. “Private Network-Network Interface Specification Version 1.0”. In: *PNNI Specification Working Group ATM forum, March*. 1996.

[14] Roche A Guérin and Ariel Orda. “QoS routing in networks with inaccurate information: theory and algorithms”. In: *IEEE/ACM Transactions on Networking (TON)* 7.3 (1999), pp. 350–364.

[15] *IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE -Org, 2012. DOI: 10.1109/ieeestd.2012.6178212. URL: http://dx.doi.org/10.1109/IEEESTD.2012.6178212.

[16] *ISO/IEC standard 7498-1:1994 - OSI/ISO 7 layer model*. http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip.

[17] J.M. Jaffe. “Algorithms for finding paths with multiple constraints”. In: *Networks* 14.1 (1984), pp. 95–116.

[18] Luo Junhai, Xue Liu, and Ye Danxia. “Research on multicast routing protocols for mobile ad-hoc networks”. In: *Computer Networks* 52.5 (2008), pp. 988 –997. ISSN: 1389-1286. DOI: DOI:10.1016/j.comnet.2007.11.016.

[19] Mirosław Kantor, Piotr Chołda, and Andrzej Jajszczyk. “Least Cost Routing (LCR) solution for inter-domain traffic distribution”. In: *Telecommunication Systems* (2011). 10.1007/s11235-011-9606-1, pp. 1–13. ISSN: 1018-4864. URL: http://dx.doi.org/10.1007/s11235-011-9606-1.

[20] John Klincewicz, James Schmitt, and Richard Wong. “Incorporating QoS into IP Enterprise Network Design”. In: *Telecommunication Systems* 20 (1 2002). 10.1023/A:1015441400785, pp. 81–106. ISSN: 1018-4864. URL: http://dx.doi.org/10.1023/A:1015441400785.

[21] G.A. Kochenberger, F. Glover, B. Alidaee, and C. Rego. “A unified modeling and solution framework for combinatorial optimization problems”. In: *OR Spectrum* 26.2 (2004), pp. 237–250.

[22] GaryA. Kochenberger, Fred Glover, Bahram Alidaee, and Cesar Rego. “Solving Combinatorial Optimization Problems Via Reformulation and Adaptive Memory Metaheuristics”. In: *Frontiers of Evolutionary Computation*. Ed. by Anil Menon. Vol. 11. Genetic Algorithms and Evolutionary Computation. Springer US, 2004, pp. 103–113. ISBN: 978-1-4020-7524-7. DOI: 10.1007/1-4020-7782-3_5. URL: http://dx.doi.org/10.1007/1-4020-7782-3_5.

[23] Vachaspathi P. Kompella, Joseph C. Pasquale, and George C. Polyzos. “Multicast routing for multimedia communication”. In: *IEEE/ACM Trans. Netw.* 1.3 (1993), pp. 286–292. ISSN: 1063-6692. DOI: http://dx.doi.org/10.1109/90.234851.

[24] Vachaspathi Peter Kompella. “Multicast routing algorithms for multimedia traffic”. PhD thesis. La Jolla, CA, USA: University of California at San Diego, 1993.

[25] E.G. Larsson. “MIMO Detection Methods: How They Work [Lecture Notes]”. In: *Signal Processing Magazine, IEEE* 26.3 (2009), pp. 91 –95. ISSN: 1053-5888. DOI: 10.1109/MSP.2009.932126.

[26] Whay Chiou Lee. “Spanning tree method for link state aggregation in large communication networks”. In: *INFOCOM'95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*. IEEE. Washington, DC, USA: IEEE Computer Society, 1995, pp. 297–302. ISBN: 0-8186-6990-X.

[27] János Levendovszky, Alpár Fancsali, Csaba Végso, and Gábor Rétvári. “QoS Routing with Incomplete Information by Analog Computing Algorithms”. In: *Quality of Future Internet Services: Second COST 263 International Workshop, QofIS 2001 Coimbra, Portugal, September 24–26, 2001 Proceedings*. Ed. by Mikhail I. Smirnov, Jon Crowcroft, James Roberts, and Fernando Boavida. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 127–137. ISBN: 978-3-540-45412-0. DOI: 10.1007/3-540-45412-8_10. URL: http://dx.doi.org/10.1007/3-540-45412-8_10.

[28] Dean H. Lorenz and Ariel Orda. “QoS routing in networks with uncertain parameters”. In: *IEEE/ACM Transactions on Networking* 6.6 (Dec. 1998), pp. 768–778. ISSN: 1063-6692. DOI: 10.1109/90.748088. URL: citeseer.ist.psu.edu/lorenz98qos.html.

[29] Z. Luo, W. Ma, A.M.C. So, Y. Ye, and S. Zhang. “Semidefinite relaxation of quadratic optimization problems”. In: *Signal Processing Magazine, IEEE* 27.3 (2010), pp. 20–34.

[30] Nguyen Cong Luong, Dinh Thai Hoang, Ping Wang, Dusit Niyato, Dong In Kim, and Zhu Han. “Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey”. In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2546–2590.

[31] Haci Mantar. “A scalable QoS routing model for diffserv over MPLS networks”. In: *Telecommunication Systems* 34 (3 2007). 10.1007/s11235-007-9035-3, pp. 107–115. ISSN: 1018-4864. URL: http://dx.doi.org/10.1007/s11235-007-9035-3.

[32] Stéphane Martignoni and Thomas Kühnel. “Extension of Classical IP over ATM to support QoS at the application level”. In: *Telecommunication Systems* 11 (3 1999). 10.1023/A:1019161721356, pp. 291–303. ISSN: 1018-4864. URL: http://dx.doi.org/10.1023/A:1019161721356.

[33] *MeasurementLab disco dataset blogpost*. https://www.measurementlab.net/blog/disco-dataset/#new-disco-switch-telemetry-dataset. Accessed: 2019.05.14.

[34] *MeasurementLab homepage*. https://www.measurementlab.net. Accessed: 2019.05.14.
[35] Peter Merz and Kengo Katayama. “Memetic algorithms for the unconstrained binary quadratic programming problem”. In: *Biosystems* 78.1–3 (2004), pp. 99–118. ISSN: 0303-2647. DOI: 10.1016/j.biosystems.2004.08.002. URL: http://www.sciencedirect.com/science/article/pii/S0303264704001376.
[36] J.C. Picard and H.D. Ratliff. “A graph-theoretic equivalence for integer programs”. In: *Operations Research* (1973), pp. 261–269.
[37] J.C. Picard and H.D. Ratliff. “Minimum cuts and related problems”. In: *Networks* 5.4 (1975), pp. 357–370.
[38] C. Pornavalai, G. Chakraborty, and N. Shiratori. “A neural network approach to multicast routing in real-time communication networks”. In: *ICNP’95: Proceedings of the 1995 International Conference on Network Protocols*. IEEE Computer Society, 1995, p. 332. ISBN: 0-8186-7216-1.
[39] Chotipat Pornavalai, Goutam Chakraborty, and Norio Shiratori. “Routing with multiple QoS requirements for supporting multimedia applications”. In: *Telecommunication Systems* 9 (3 1998). 10.1023/A:1019160226383, pp. 357–373. ISSN: 1018-4864. URL: http://dx.doi.org/10.1023/A:1019160226383.
[40] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. “A taxonomy and survey of cloud computing systems”. In: *INC, IMS and IDC, 2009. NCM’09. Fifth International Joint Conference on*. Ieee. 2009, pp. 44–51.
[41] Maria Alejandra Rodriguez and Rajkumar Buyya. “A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments”. In: *Concurrency and Computation: Practice and Experience* (2016).
[42] Hussein F. Salama, Douglas S. Reeves, and Yannis Viniotis. “Evaluation of multicast routing algorithms for real-time communication on high-speed networks”. In: *IEEE Journal on Selected Areas in Communications* 15 (1997), pp. 332–345.
[43] Anees Shaikh, Jennifer Rexford, and Kang G Shin. “Dynamics of quality-of-service routing with inaccurate link-state information”. In: *Ann Arbor* 1001.CSE-TR-350-97 (1997), pp. 48109–2122. URL: citeseer.ist.psu.edu/shaikh97dynamics.html.
[44] Sukhpal Singh and Inderveer Chana. “A survey on resource scheduling in cloud computing: Issues and challenges”. In: *Journal of Grid Computing* 14.2 (2016), pp. 217–264.
[45] Tim Szigeti and Christina Hattingh. *End-to-end qos network design*. Cisco press, 2005.

[46] **Dávid Tisza**, András Oláh, and János Levendovszky. "Multi-user detection using non-parametric Bayesian estimation by feed forward neural networks". In: *Telecommunication Systems* (2015), pp. 1–11.
[47] **Dávid Tisza**, András Oláh, and János Levendovszky. "Novel algorithms for quadratic programming by using hypergraph representations". In: *Wireless personal communications* 77.3 (2014), pp. 2305–2339.
[48] **Dávid Tisza**, Péter Vizi, Janos Levendovszky, and András Oláh. "Multicast Routing in Wireless Sensor Networks with Incomplete Information". In: *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*. 2011, pp. 1–5.
[49] Sergio Verdu. *Multiuser Detection*. Cambridge University Press, 1998. ISBN: 0521593735.
[50] J. Wang. "Discrete Hopfield network combined with estimation of distribution for unconstrained binary quadratic programming problem". In: *Expert Systems with Applications* 37.8 (2010), pp. 5758–5774.
[51] Minxian Xu, Wenhong Tian, and Rajkumar Buyya. "A Survey on Load Balancing Algorithms for VM Placement in Cloud Computing". In: *arXiv preprint arXiv:1607.06269* (2016).
[52] Changsheng Yu, Li Yu, Yuan Wu, Yanfei He, and Qun Lu. "Uplink Scheduling and Link Adaptation for Narrowband Internet of Things Systems". In: *IEEE Access* 5 (2017), pp. 1724–1734.