Information Sciences and Technologies Bulletin of the ACM Slovakia

Association for Computing Machinery

Advancing Computing as a Science & Profession

September 2014
Volume 6, Number 3

Aim and Scope of the Information Sciences and Technologies Bulletin of the ACM Slovakia

ACM Slovakia offers a forum for rapid dissemination of research results in the area of computing/informatics and more broadly of information and communication sciences and technologies. It is primarily a web based bulletin publishing results of dissertations submitted at any university in Slovakia or elsewhere, perhaps also results of outstanding master theses. Besides that, conferences that meet bulletin's expectations with regard to scientific rigor are invited to consider publishing their papers in the bulletin in form of special issues. Besides the web version of the bulletin, a paper version is available, too.

The Bulletin aims:

- To advance and to increase knowledge and interest in the science, design, development, construction, languages, management and applications of modern computing a.k.a. informatics, and more broadly of information and communication sciences and technologies.
- To facilitate a communication between persons having an interest in information and communication sciences and technologies by providing a forum for rapid dissemination of scholarly articles.

Scope of the Bulletin is:

- original research in an area within the broader family of information sciences and technologies, with a particular focus on computer science, computer engineering, software engineering and information systems, and also other similarly well established fields such as artificial intelligence or information science.

Types of contributions:

- **Extended abstracts of doctoral dissertations.** This is the primary type of article in the Bulletin. It presents main contributions of the dissertation in form of a journal paper together with separate section with list of published works of the author. In Slovakia and the Czech Republic, it corresponds to typical length of so called *autoreferat*. In fact, it is envisaged that publishing the extended abstract in the Bulletin makes *autoreferat* obsolete and eventually can replace it completely. It should be noted that by publishing it in the Bulletin, the extended abstract will receive a much wider dissemination. Exceptionally, at the discretion of the Editorial Board, the Bulletin may accept extended abstracts of other than doctoral theses, e.g. Master theses, when research results reported are sufficiently worthy of publishing in this forum. Rules and procedures of publishing are similar.
- **Conference papers.** The Bulletin offers organizers of interesting scientific events in some area within the scope of the Bulletin to consider publishing papers of the Conference in the Bulletin as its special issue. Any such proposal will be subject of discussion with the Editorial Board which will ultimately decide. From the scientific merit point of view, method of peer reviewing, acceptance ratio etc. are issues that will be raised in the discussion.

Besides that the Bulletin may include other types of contributions that will contribute to fulfilling its aims, so that it best serves the professional community in the area of information and communication sciences and technologies. There are four regular issues annually.

Editorial Board

Executive Editor: *Peter Lacko*
Cover Design: *Peter Lacko*
Typeset in LaTeX using style based on ACM SIG Proceedings Template.

Partitioning Networks into Cliques: A Randomized Heuristic Approach

David Chalupa*
Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
david.chalupa@stuba.sk

Abstract

In the context of community detection in social networks, the term community can be grounded in the strict way that simply everybody should know each other within the community. We consider the corresponding community detection problem. We search for a partitioning of a network into the minimum number of non-overlapping cliques, such that the cliques cover all vertices. This problem is called the clique covering problem (CCP) and is one of the classical NP-hard problems. For CCP, we propose a randomized heuristic approach. To construct a high-quality solution to CCP, we present an iterated greedy (IG) algorithm. IG can also be combined with a heuristic used to determine how far the algorithm is from the optimum in the worst case. Randomized local search (RLS) for maximum independent set was proposed to find such a bound. The experimental results of IG and the bounds obtained by RLS indicate that IG is a very suitable technique for solving CCP in real-world graphs. In addition, we summarize our basic rigorous results, which were developed for analysis of IG and understanding of its behavior on several relevant graph classes.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods; F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures; G.2.3 [**Discrete Mathematics**]: Applications; I.5.3 [**Pattern Recognition**]: Clustering

*Recommended by thesis supervisor: Prof. Jiří Pospíchal Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on June 25, 2014.

Chalupa, D. Partitioning Networks into Cliques: A Randomized Heuristic Approach. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 6, No. 3 (2014) 1-8

Keywords

clique covering problem, heuristic algorithms, complex networks, iterated greedy, community detection

1. Introduction

Many significant real-world problems can be modeled using networks. For the purpose of this work, a network will be formalized as a structure consisting of vertices and undirected edges between pairs of vertices. Thus, a network will simply be an undirected, unweighted graph $G = [V, E]$.

The problem we study in this work is the (vertex) *clique covering problem* (CCP). In CCP, the aim is to divide the vertices of a graph into the minimum number of subsets, in which every pair of vertices has an edge between them. A subgraph with this property is called *clique*. For example, in the context of social networks, the aim of CCP is to partition the network into minimum number of groups, such that everybody knows each other within each group.

One can intuitively see the relation between CCP and *community detection*. An interesting fact is that community detection is a relatively loosely defined problem [24] and the properties of communities found by an algorithm strongly depend on the optimized objective [19]. On the other hand, CCP has a strict definition. The optimal number of cliques needed to solve CCP is a specific value for a network, while there might be more reasonable values for the number of communities. In community detection, this is a rather rare but desirable property of CCP. However, it is *NP-hard* to solve CCP for an arbitrary graph [16].

Our work is closely related to two major fields within computer science. The first field is *artificial intelligence*, since we use heuristic algorithms (sometimes also referred to as heuristics) to solve our problem. Such algorithms provide a solid solution to a problem but are not designed directly with respect to a guarantee of solution quality. Design, analysis and applications of heuristic algorithms are integral parts of the artificial intelligence field [23].

The second related field is *graph mining*, which aims at analysis of large real-world graphs using algorithmic methods. Emphasis in graph mining is not only on using the properties of real-world graphs, but also on scalability of the algorithms, which are used to discover information from structure of the graphs [3].

In the following, we present a randomized heuristic approach to solve CCP. This approach uses our own *greedy clique covering* (GCC) algorithm, which receives a permutation of vertices and partitions the vertices into cliques in the order determined by the permutation. Next, we use *iterated greedy* (IG) as a technique to optimize the permutation, thus, optimizing the resulting clique covering. This approach showed much promise for real-world networks. As the last step, we extended IG with a heuristic to compute a lower bound, since for real-world networks, we do not know the optimum in advance and have to bound it. A lower bound based on a heuristic for maximum independent set was used. This, in combination with GCC and IG, allowed us to show that CCP can be solved near-optimally and often even optimally for many real-world networks.

The paper is structured as follows. In Section 2, we briefly provide the background and motivation for our investigations, as well as a short review of related work. In Section 3, we present our approach to solve CCP. In Section 4, we provide an overview of experimental results achieved by our approach. In Section 5, we shortly summarize the analytical results, which shed light on how our approach really is able to find solutions to CCP and what are its disadvantages. Finally, in Section 6, we summarize the contribution.

2. Background, Motivation and Related Work

We now formulate the studied problem formally. Let *density* of a graph $G = [V, E]$ be determined by $d(G) = \frac{|E|}{|V|(|V|-1)/2}$. Furthermore, let $G(V_i)$ denote the *subgraph of G induced by class $V_i \subseteq V$*, i.e. the graph $G' = [V_i, E_i]$, where E_i contains only the edges in E between the vertices in V_i. We will simplify the notation of graph's metrics in the way that the number of vertices $|V| = n$ and the number of edges $|E| = m$.

Let $S = \{V_1, V_2, ..., V_k\}$ be a partitioning of the vertex set V of an undirected, unweighted graph G into classes $V_1, V_2, ..., V_k$ such that

- the classes cover all vertices, i.e. $\cup_{i=1}^{k} V_i = V$ and
- the classes are non-overlapping, i.e. $\forall\ i, j$ such that $i \neq j$ it holds that $V_i \cap V_j = \emptyset$.

Then, we will call S a *(vertex) clique covering* of G if and only if all subgraphs induced by this partitioning are cliques, i.e. $\forall\ i = 1, 2, ..., k\ \ d(G(V_i)) = 1$. We will refer to the minimum k, for which CCP can be solved, as the *clique covering number* of G and denote it by $\vartheta(G)$.

We note that CCP is closely related to the *graph coloring problem*. Suppose that $\overline{G}$ is the complementary graph to G, i.e. a graph containing edges between pairs of vertices, which were not present in G and vice versa. In a social network, an edge in $\overline{G}$ would mean that two persons do not know each other. Then, a solution to graph coloring problem for $\overline{G}$ represents a solution to CCP for G.

The best known approximation algorithm for graph coloring (and, thus, also for clique covering) is due to Haldórsson [13]. The approximation ratio, i.e. the ratio between the result of this algorithm and the optimum is at most $n(\log\log n)^2/(\log^3 n)$. Such an approximation ratio is not favorable for real-world applications. Therefore, *heuristic algorithms* are a suitable choice to solve both graph coloring and clique covering problems. For graph coloring, greedy heuristics are more scalable for large graphs. For smaller but structurally difficult problem instances, a large spectrum of local search and evolutionary algorithms is available [11].

It is however known that heuristics tend to perform better for some instances of a problem, while being less efficient for other types of instances. Therefore, structure of the studied graphs plays an important role in design of an efficient heuristic for CCP.

We have already indicated that CCP is practically interesting for real-world networks. Such networks often have a non-trivial structure and include social and biological networks [12], research citation networks, computer networks [25] or language networks [20]. Real-world networks are often *sparse*, i.e. they may contain a high number of vertices but the vertices are adjacent only to a limited number of other vertices. Let $m(n)$ be the number of edges as a function of the number of vertices of a network. For real-world networks, it often holds that $m(n) \prec n^2$ (in other notation, $m(n) = o(n^2)$), i.e. the number of edges grows much more slowly than the number of pairs of vertices. However, this means that for the complementary graph, the number of edges will grow quadratically. To put it into the context of social networks, a person knows only a limited number of other people, but the number of people not known will be much higher.

Even simple graph coloring heuristics have $\Omega(m)$ complexity (i.e. a complexity lower bounded by the number of edges), including greedy graph coloring [26], Brélaz's heuristic [2] and Leighton's recursive largest first heuristic [18]. For a dense graph, these are not well scalable. Therefore, we proposed our own specialized technique to solve CCP directly, which is well scalable for large sparse graphs and gives high-quality results for real-world networks.

3. Our Approach to Solve the Clique Covering Problem (CCP)

In this section, we introduce our original results. We first describe our greedy clique covering (GCC) algorithm and an iterated greedy (IG) algorithm used to improve the results obtained by GCC [4, 5]. Next, we present a technique to compute a lower bound for the optimal solution to CCP based on maximum independent set. To compute this lower bound, we proposed a randomized local search (RLS) algorithm, conceptually very similar to IG [6].

3.1 Greedy Clique Covering (GCC) and an Order-based Representation of CCP

GCC is an algorithm, which takes the vertices in a particular order and labels them. Then, the vertices, which have the same label, form a clique. The way how GCC chooses a label for a vertex is illustrated by Figure 1. Let us have a global information on how many vertices have which label. This information will be stored in array *sizes*. Let $\Gamma(v, c)$ be the number of vertices, which are neighbors of v and have label c. Then, vertex v can join the vertices with label c without violating the clique property, when $\Gamma(v, c) = sizes(c)$. When there are more values of c satisfying this condition, the one with the lowest index is chosen.

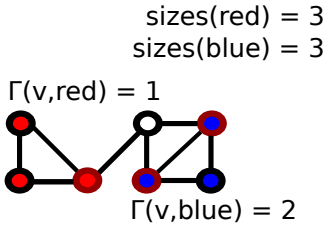

Figure 1: Illustration of the way how GCC chooses labels. The currently unlabeled vertex will receive label c if $\Gamma(v, c) = sizes(c)$. In this case, there are only 2 blue neighbors and 1 red neighbor of v. Therefore, the currently unlabeled vertex must receive its own new label.

Algorithm 1: Greedy Clique Covering (GCC)

	Greedy Clique Covering (GCC)
	Input: graph $G = [V, E]$
	permutation $P = [P_1, P_2, ..., P_n]$ of vertices in V
	Output: clique covering S of G
1	for $c = 1..n$
2	$sizes(c) = 0$
3	for $i = 1..n$
4	$j = P_i$
5	$c = find\_equal(\Gamma(v_j, c), sizes(c))$
6	$V_c = V_c \cup \{v_j\}$
7	return $S = \{V_1, V_2, ..., V_k\}$

Algorithm 1 describes how GCC can be implemented. The input is a permutation P of vertices of our graph G. In steps 1-2, *sizes* array is initialized. Then, the vertices are iteratively labeled. In step 4, vertex v_j is chosen for labeling, based on permutation P. In step 5, GCC finds the label for v_j in operation $find\_equal$. This can be done in such a way that neighbors of v are iterated and values in *sizes* are decremented, based on which label each neighbor has. If some value in $sizes(c)$ reaches 0, then apparently, c is a candidate label for v_j. The original values in *sizes* can then be restored by iterating the neighbors once again. In step 6, the labeling itself is performed. The advantage of this strategy is that for each vertex, we iterate only over its neighbors to choose a label. Therefore, the time complexity of GCC is $\mathcal{O}(m)$, since $\sum_{v \in V} \deg(v) = 2m$.

It is important to note that GCC performs differently for different input permutations. Therefore, GCC can be viewed as a mapping from the space of permutations of vertices to the search space of clique coverings. More formally, GCC is a mapping μ from the space of all permutations of n objects S_n to the space of clique coverings Φ, i.e. $\mu : S_n \rightarrow \Phi$, as shown by Figure 2. In evolutionary computation, such a mapping is often called *genotype-phenotype mapping*. The aim is to find a permutation P_{opt} such that $\mu(P_{opt}) = S_{opt}$, where S_{opt} is an optimal clique covering. It can be shown that for an arbitrary graph, there always is a permutation P_{opt} with this property.

Theorem 1. For an arbitrary graph G, there is a permutation, for which greedy clique covering will produce the optimal solution with $\vartheta(G)$ cliques.

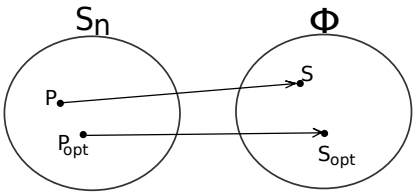

Figure 2: Illustration of GCC as a mapping. GCC can be viewed as function which, for a permutation $P \in S_n$, returns a clique covering $S \in \Phi$. Therefore, the problem is solved indirectly. Instead of searching for an optimal clique covering S_{opt}, one can search for an optimal permutation P_{opt}, for which GCC constructs S_{opt}.

Algorithm 2: Iterated Greedy (IG) Algorithm for CCP

	Iterated Greedy (IG) Algorithm for CCP
	Input: graph $G = [V, E]$
	Output: clique covering S of G
1	$P = random\_permutation(1, 2, ..., n)$
2	while stopping criterion is not met
3	$\{V_1, V_2, ..., V_k\} = GCC(G, P)$
4	with p_{rev} probability
5	$P = [V_k, V_{k-1}, ..., V_1]$
6	else
7	$P = random\_permutation(V_1, V_2, ..., V_k)$
8	if $\vartheta(G)$ is known and $k = \vartheta(G)$
9	return $S = \{V_1, V_2, ..., V_k\}$
10	return $S = \{V_1, V_2, ..., V_k\}$

With GCC, we established an *order-based representation of CCP*, i.e. the problem of searching for the optimal clique covering is now transformed into a problem of searching for an optimal ordering. As we have already indicated, the key property of our approach is its scalability. GCC performs in $\mathcal{O}(m)$ time, which is favorable for sparse graphs. This will allow one to use GCC repeatedly many thousands of times in a stochastic search algorithm and achieve reasonable results in a reasonable time even for large graphs.

3.2 Iterated Greedy (IG) Clique Covering

Iterated greedy (IG) is a stochastic local search technique [14], which combines a greedy algorithm with stochastic optimization of its component. It was used for the first time by Chvátal to solve the set covering problem [7], which suggests that it is suitable for covering problems.

Our main inspiration comes from the iterated greedy algorithm for graph coloring proposed by Culberson and Luo [9, 10]. This algorithm used block-based properties of greedy graph coloring algorithm. We now describe a way how to use such an idea analogically to improve the results obtained by GCC.

The way how our IG works is given by the pseudocode of Algorithm 2. In step 1, we generate a permutation of vertices uniformly at random. Next, an iterative procedure is performed. In step 3, we use GCC with permutation P to construct a clique covering $\{V_1, V_2, ..., V_k\}$. Then, one of two possible block-based mutations is performed. The

first is the *reverse mutation*, where the blocks $V_1, V_2, ..., V_k$ are put into the next permutation in the reverse order. The second one is the *random mutation*, by which the blocks are randomly shuffled. We note that the vertices within blocks are internally preserved. This is done in steps 4-7. If we know the optimal value $\vartheta(G)$, the improvement process is stopped in steps 8-9. Steps 2-9 are repeated until a stopping criterion is met.

One should note that by preserving the blocks internally, the number of cliques used to cover the graph can only become lower. This is because a new block simply cannot be created this way, because of the greedy nature of GCC. Therefore, although the random mutation operator performs at the level of blocks entirely at random, the search is not blind. The objective function is non-increasing during the run of IG, which is typical for local search algorithms.

3.3 Randomized Local Search (RLS) for a Lower Bound

In our experimental work, we quickly noticed that IG seems to perform well for real-world networks [4]. However, one usually does not know the optimal number of cliques $\vartheta(G)$ for a network with an unknown structure. Therefore, it is practically interesting to bound the optimum from below. This way, we can determine how well our algorithm performed. It can be easily shown that the clique covering number is bounded from below and from above according to Lemma 1.

Lemma 1. Let G be an undirected graph with minimum degree $\delta_{min}(G)$, clique covering number $\vartheta(G)$, maximum independent set size $\alpha(G)$ and maximum clique size $\omega(G)$. Then, $\vartheta(G)$ is bounded in the following way:

$$\max\left\{\alpha(G), \frac{n}{\omega(G)}\right\} \leq \vartheta(G) \leq n - \delta_{min}(G). \quad (1)$$

For real-world networks, the max operation in the lower bound will almost always evaluate to $\alpha(G)$, i.e. $\alpha(G)$ will be a tighter bound. The size of maximum independent set can be easily interpreted in the context of social networks. It is the maximum size of a group, in which nobody knows each other. Obviously, if $\alpha(G)$ is the size of such a set, it takes $\alpha(G)$ cliques to cover its vertices.

The only issue here is that maximum independent set problem is also NP-hard [16]. Therefore, it is a problem for heuristic algorithms, too. We designed a randomized local search algorithm (RLS) to estimate the maximum independent set. It is specified in pseudocode of Algorithm 3. We note that RLS_p^1 means that 1 mutation is performed per iteration and the algorithm searches in the space of permutations of vertices.

RLS_p^1 starts with a random permutation of vertices, generated in step 1. In step 3, greedy independent set procedure is used. This procedure takes the vertices in order determined by the current permutation P. For each vertex, we simply determine whether it can be added to the independent set. This occurs when the vertex is not adjacent to any vertex within the independent set. This way, some independent set is generated. In step 6, we choose a random vertex from the current permutation P and move it to the first position in the permutation. The other vertices are then shifted to the right. Greedy independent set procedure is used once again. The new permutation is accepted if and only if the new independent set is at least as large as the previous one. This is repeated until a stopping criterion is met.

Algorithm 3: RLS_p^1 Algorithm for the Maximum Independent Set Size

	RLS_p^1 Algorithm for the Maximum Independent Set Size
	Input: graph $G = [V, E]$ Output: the size $\alpha(G)$ of the estimated maximum independent set
1	$P = random\_permutation(1, 2, ..., n)$, $P^* = P$, $k^* = 1$
2	while *stopping_criterion* is not met
3	$k = \lvert greedy\_independent\_set(G, P)\rvert$
4	if $k \geq k^*$
5	$k^* = k$, $P^* = P$
6	$j = uniformly\_random(2, n)$
7	$P = jump(j, 1, P^*)$
8	return $\alpha(G) = k^*$

4. Overview of Experimental Results

The verification of our approach was conducted using both experimental and analytical methodologies. Let us first briefly present the experimental results we obtained for our approach. These results can be divided into three main parts:

- a case study of IG on graphs with planted cliques,
- experimental comparison of IG with Brélaz's heuristic and its modification,
- verification of quality of results obtained by IG and RLS on real-world networks.

In the first step, we conducted a case study of IG on *graphs with planted cliques*. A graph with planted cliques consists of embedded cliques with some predefined size, it is a special case of planted partition model of clustered graphs [8]. Planted cliques are connected uniformly at random. The main aim of our study was to determine how well IG is able to restore these planted cliques. On graphs, which were sparse enough between the planted cliques, IG performed very efficiently and was able to find the optimum. Additionally, we discovered that larger planted cliques took less time to discover than smaller cliques [4].

In the next step, we compared IG to Brélaz's heuristic (denoted by BRE) and its modification we called saturation-based greedy clique covering (SAT-GCC). Brélaz's heuristic was chosen because of its good tradeoff between scalability and quality of results, since it works in $\mathcal{O}(n^2)$ time [2]. SAT-GCC uses a strategy similar to Brélaz's heuristic but can be implemented to run faster. In SAT-GCC, after assignment of a vertex into a clique, for each of its neighbors, which are adjacent to all of the vertices of the current clique, its saturation is incremented. Vertices are ordered primarily based on highest saturation, secondarily based on lowest degree and possible ties are resolved at random.

Table 1: The comparison of the quality of results obtained by the Brélaz's heuristic (BRE), saturation-based greedy clique covering (SAT-GCC) and the iterated greedy (IG) heuristic using GCC on CCP. The numbers denote the approximations of the clique covering number $\vartheta(G)$ for each graph and each heuristic. These were computed for uniform random graphs, Leighton graphs, which contain embedded cliques of different sizes and graphs obtained by crawling a local social network. The best results are highlighted in bold.

G	BRE	SAT-GCC	IG
Erdős-Rényi uniform random graphs			
$unif1000\_0.1$	302	310	**242**
$unif5000\_0.1$	1241	1290	**1064**
$unif10000\_0.1$	2326	2403	**2030**
$unif20000\_0.01$	7640	7863	**6403**
Leighton graphs from DIMACS instances			
$le450\_15a$	85	89	**80**
$le450\_15b$	92	90	**82**
$le450\_15c$	72	76	**58**
$le450\_15d$	73	73	**59**
$le450\_25a$	**91**	94	**91**
$le450\_25b$	81	83	**80**
$le450\_25c$	61	62	**55**
$le450\_25d$	60	59	**51**
Social graphs			
$soc1000$	**759**	**759**	**759**
$soc2000$	**1471**	1475	**1471**
$soc10000$	6620	6671	**6618**
$soc20000$	12770	12899	**12764**

Table 1 shows the comparison of the numbers of cliques found by the three studied algorithms. The columns denote the three algorithms, the rows denote the instances and the values are the numbers of cliques found by a particular algorithm for the respective test graph. As test instances, we used Erdős-Rényi uniform random graphs, in which edges are put between each pair of vertices with constant probability. Leighton graphs are also pseudorandom and contain embedded cliques with several different sizes. These graphs were designed to model large scheduling problems [18]. However, searching for the original embedded cliques is also a relevant task. Last but not least, we used several extracts of a Slovak web-based social network. Table 1 illustrates the clear dominance of our IG algorithm over these well-scalable techniques. One can see that the difference in performance of Brélaz's heuristic and IG is more pronounced in random graphs and Leighton graphs. However, the most encouraging result is probably the fact that IG surpassed Brélaz's heuristic on large social graphs with 10000 and 20000 vertices. This gives a hint that IG might be a very suitable choice to solve CCP in large real-world networks.

Finally, we tested IG and RLS on set of real-world networks. For this purpose, we used the extracts of a web-based Slovak social network, denoted by Social network I, which were already used in experiments summarized in Table 1. Social network II is a neighborhood of a single user from another social network. Network science instances are taken from other sources. These instances include a language network describing adjective-noun adjacencies (*adjnoun*), a collaboration network for the field of network science (*netscience*), a network of friendships in a karate club (*zachary*), a network describing games in a season of an American college football league ($football$) and a snapshot of Internet on the level of autonomous systems ($as-22july06$)[1]. The last set of instances includes coappearance networks for several classical literary works, which describe whether two characters in a book encounter each other[2].

As we have already mentioned, the aim of using RLS is to find a lower bound for the clique covering number $\vartheta(G)$. This way, we are able to determine how good solutions our IG algorithm is in fact able to find. Table 2 summarizes the lower bounds $\vartheta_L(G)$ obtained by RLS and the numbers of cliques $\vartheta_U(G)$ used by IG to solve CCP for a particular graph. The table contains names and sources for the graphs, the values obtained by the algorithms and the average size of a clique $\frac{n}{\vartheta_U(G)}$ in the obtained solutions. The cases, for which we obtained $\vartheta_L(G) = \vartheta_U(G)$, are highlighted in bold. Naturally, for these cases, we can declare that *IG was able to find the optimal solutions.*

Using the bounds obtained by RLS, one can see that IG was able to solve CCP optimally for 13 out of 17 of these real-world networks. For the other 4 instances, the optimal value $\vartheta(G)$ was limited to a small interval. Interestingly, the instances where IG performed optimally, include instances from all categories. It was able to solve the problem for social networks, coappearance networks and a collaboration network. We also note that the results were performed with high number of successful runs over 30 independent runs of both IG and RLS.

5. Overview of Analytical Results

In the previous section, we have shown that IG performs well experimentally. However, this did not give us much insight into how the approach really works. Especially, one can wonder how the random decisions by block-based mutation operators are able to improve solutions. Insight into this is important to understand what are the advantages and disadvantages of IG. Therefore, we aimed also to conduct an analytical investigation on the behavior of IG [5].

At this point, we give an overview of the methods we used and the analytical results we obtained. As we have already indicated, IG is a stochastic local search algorithm. Therefore, methods of analysis for evolutionary algorithms are suitable to analyze its convergence properties and obtain bounds on its expected runtime [21].

There are two methods we used to analyze the runtime of IG on several chosen classes of graphs.

1. *Fitness levels (fitness-based partitions).* This method is based on partitioning of the search space into several levels, based on values of the objective function, such that an algorithm can only move to a better

[1]To the best of our knowledge, this Internet snapshot was not previously published. It is available on this site: `http://www-personal.umich.edu/~mejn/netdata/`.

[2]All these instances are available or a link to their source is provided at: `http://www.fiit.stuba.sk/~chalupa/benchmarks/ccp`

Table 2: Summary of the lower bounds $\vartheta_L(G)$ obtained by RLS and the numbers of cliques $\vartheta_U(G)$ used by IG on complex network instances. The results, for which we obtained that $\vartheta_L(G) = \vartheta_U(G)$, are highlighted in bold. There are 13 such networks, for which we showed this way that the result obtained by IG is actually the optimal clique covering [6].

source of G	file name	$\vartheta_L(G)$	$\vartheta_U(G)$	$\frac{n}{\vartheta_U(G)}$
Web-based social network extracts [4]				
Social network I.	$soc500$	**377**	**377**	1.33
Social network I.	$soc1000$	**759**	**759**	1.32
Social network I.	$soc2000$	1470	1471	1.36
Social network I.	$soc10000$	**6618**	**6618**	1.51
Social network I.	$soc20000$	**12764**	**12764**	1.57
Social network II.	$soc52$	**15**	**15**	3.47
Network science instances				
Adjective-noun adjacencies [22]	$adjnoun$	53	55	2.04
Network science collaborations [22]	$netscience$	**690**	**690**	2.30
Les Miserables network [17]	$lesmis$	**35**	**35**	2.20
Zachary Karate Club [27]	$zachary$	**20**	**20**	1.70
American College Football [12]	$football$	21	22	5.23
Snapshot of the Internet	$as-22july06$	19660	19661	1.17
Characters' coappearance networks from DIMACS coloring instances [15]				
Anna Karenina	$anna$	**80**	**80**	1.73
David Copperfield	$david$	**36**	**36**	2.42
Huckleberry Finn	$huck$	**27**	**27**	2.74
Iliad and Odyssey	$homer$	**341**	**341**	1.65
Jean Valjean	$jean$	**38**	**38**	2.11

fitness level or stagnate. If p_i is the minimum probability of improvement from i-th level to a better fitness level, then the expected waiting time for an improvement to a better fitness level is $1/p_i$. Consequently, runtime is determined by the sum of waiting times over all suboptimal fitness levels [21].

2. *Cover time of random walks.* This method is used to analyze the runtime of evolutionary algorithms on plateaus, i.e. areas consisting of solutions with equal values of objective function. On a plateau, a search algorithm behaves like a random walk. For a plateau with $|V|$ solutions and $|E|$ transformations between them, it holds that the expected time to visit each solution at least once is $2|E|(|V|-1)$ [1].

In runtime analysis of IG, we usually used fitness levels to upper bound the expected runtime of IG. A particular fitness level usually consisted of clique coverings with equal number of cliques. To estimate the expected time to obtain an improvement to a better fitness level, we used either the cover time of random walks or some specific knowledge about the structure of the graph.

Table 3 summarizes the analytical results we proved for GCC and IG. For the cases, where the algorithms behave suboptimally, we used metrics of approximation quality. Let f_{opt} be the optimal value of objective function for a minimization problem and let f be the worst possible value obtained by the algorithm. Then, we define approximation ratio as $R = f/f_{opt}$ and maximum discrepancy as $D = f - f_{opt}$.

Star graphs consist of one central vertex adjacent to several vertices with degree 1. Both GCC and IG behave optimally in linear time on this class of graphs. This is an example of a class of graphs, where block-based mutation is not needed, since GCC gives the optimum for any permutation. On *paths* (graphs consisting of vertices chained to a "line"), GCC achieves only 4/3-approximation, while IG behaves optimally in $\mathcal{O}(n^5)$ time. On *complements of bipartite graphs*, it depends on the number of edges between the partitions. If the graph is sparse enough, IG behaves optimally in $\mathcal{O}(n^3)$ time, while GCC may overestimate with high discrepancy. However, there is a complement of a bipartite graph, where IG can get stuck with probability lower bounded by 1/15. Finally, both GCC and IG have a similar worst-case result. There is a class of graphs, where GCC gives a suboptimal solution and IG is not able to improve it using block-based mutations with an overwhelming probability.

6. Conclusions

We proposed a new technique to solve the (vertex) *clique covering problem* (CCP). The aim of CCP is to partition the vertices of a network into minimum number of groups, which induce cliques (subgraphs with all pairs of vertices being adjacent). In the context of social networks, CCP can be viewed as a strict variant of *community detection*, where community is viewed as a group of persons, in which everybody knows each other.

The main contributions of our work were presented in three parts. Each part was published in a separate paper [4, 5, 6].

In the first part, we proposed our *greedy clique covering* (GCC) algorithm as a mapping of a permutation of vertices to a clique covering. Next, we extended it to an iterated greedy (IG) algorithm. The key advantage of GCC over existing approaches is that it works in $\mathcal{O}(m)$ time, where m is the number of edges of the graph. This makes GCC well-scalable and useable thousands of times

Table 3: Overview of the analytical results on GCC and IG on several studied graph classes. Quality of results and runtime of the algorithms are given, R denotes approximation ratio and D denotes maximum discrepancy between the found solution and the optimum.

graph class	GCC	IG
star graphs	optimal, $\mathcal{O}(n)$ time	optimal, $\mathcal{O}(n)$ time
paths	suboptimal, $R = 4/3$	optimal, $\mathcal{O}(n^5)$ time
complements of bipartite graphs: partitions of size $n/2^*$, $m_{out} < n/2$	suboptimal, $D = n/2 - 1$	optimal, $\mathcal{O}(n^3)$ time
complements of bipartite graphs: partitions of size $n/2^*$, $m_{out} \geq n/2$	suboptimal, $D = n/2 - 1$	can be suboptimal with probability $p \geq 1/15$
worst-case result	can be suboptimal with probability $1 - o(1)$	can be suboptimal with probability $1 - o(1)$

* Here, we assume that n is even, i.e. $n/2$ is an integer.

in an iterative optimization algorithm. We designed IG as such an algorithm. First experimental results showed that IG is able to perform well on graphs with planted cliques, where the aim was to restore the planted cliques. Additionally, IG experimentally surpassed the results of the well-known Brélaz's graph coloring heuristic applied to solve CCP.

Secondly, we studied the properties of GCC and IG analytically. We managed to show analytically, when the block-based mutation operators used by IG are able to improve the solutions and what are their disadvantages. We formulated a sufficient condition for an improvement by block-based mutation operator to occur. We have shown that IG is able to find the optimum in *polynomial time* on *path graphs* (chain graphs) and *sparse complements of bipartite graphs*. On the other hand, we showed that there is a class of graphs, where IG will tend to get stuck in a suboptimal solution with an overwhelming probability. Fortunately, such a class of graphs is highly artificial and unlikely to appear in real-world applications.

Last but on least, we experimentally demonstrated that IG is a suitable technique to solve CCP in *real-world networks*. For real-world networks, one usually does not know the optimal value in advance. Therefore, we proposed another heuristic to determine how far the solutions obtained by IG are from the optimum. We designed a randomized local search (RLS) algorithm for maximum independent set to accomplish this goal. IG and RLS were experimentally tested on social networks, on several graphs studied in network science, as well as on a set of coappearance networks for classical literary works. For 13 out of 17 graphs, IG found the optimum, while for the other 4 graphs, the optimum was bounded inside a very small interval.

Acknowledgement. This contribution was supported by Grant Agency VEGA SR under the grants 1/0553/12 and 1/0458/13.

References

[1] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovasz, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *Proc. of the 20th Annual Symposium on Foundations of Computer Sc., FOCS '79*, pp 218–223, 1979.

[2] D. Brélaz. New methods to color vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[3] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38(1):Article No. 2, 2006.

[4] D. Chalupa. On the efficiency of an order-based representation in the clique covering problem. In T. Soule and J. Moore, editors, *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, GECCO '12, pages 353–360, New York, NY, 2012. ACM.

[5] D. Chalupa. An analytical investigation of block-based mutation operators for order-based stochastic clique covering algorithms. In C. Blum and E. Alba, editors, *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, GECCO '13, pages 495–502, New York, NY, 2013. ACM.

[6] D. Chalupa. Construction of near-optimal vertex clique covering for real-world networks. *Computing and Informatics*, to appear.

[7] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[8] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. In D. S. Hochbaum, K. Jansen, J. D. P. Rolim, and A. Sinclair, editors, *Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems: Randomization, Approximation, and Combinatorial Algorithms and Techniques*, RANDOM-APPROX '99, pages 221–232, London, 1999. Springer.

[9] J. C. Culberson. Iterated greedy graph coloring and the difficulty landscape. Technical Report TR92-07, University of Alberta, 1992.

[10] J. C. Culberson and F. Luo. Exploring the k-colorable landscape with iterated greedy. In D. S. Johnson and M. Trick, editors, *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, pages 245–284, Providence, RI, 1995. American Mathematical Society.

[11] P. Galinier and A. Hertz. A survey of local search methods for graph coloring. *Computers & Operations Research*, 33(9):2547–2562, 2006.

[12] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[13] M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 1993.

[14] H. H. Hoos and T. Stützle. *Stochastic local search: Foundations & applications*. Elsevier, Amsterdam, 2004.

[15] D. S. Johnson and M. Trick. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*. American Mathematical Society, Providence, RI, 1996.

[16] R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, NY, 1972.

[17] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1993.

[18] F. T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, 84(6):489–503, 1979.

[19] J. Leskovec, K. J. Lang, and M. W. Mahoney. Empirical comparison of algorithms for network community detection. In J. F. M. Rappa, P. Jones and S. Chakrabarti, editors, *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 631–640, New York, NY, 2010. ACM.

[20] P. Náther and M. Markošová. Positional word web and its numerical and analytical studies. *Computing and Informatics*, 30(6):1287–1302, 2011.

[21] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization - Algorithms and Their Computational Complexity*. Springer, Berlin / Heidelberg, 2010.

[22] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(036104):036104–1–036104–19, 2006.

[23] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd ed.)*. Prentice Hall, Upper Saddle River, NJ, 2003.

[24] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[25] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Sparse graph mining with compact matrix decomposition. *Statistical Analysis and Data Mining*, 1(1):6–22, 2008.

[26] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.

[27] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

Selected Papers by the Author

D. Chalupa. Construction of Near-Optimal Vertex Clique Covering for Real-world Networks. *Computing and Informatics*, to appear.

D. Chalupa. An Analytical Investigation of Block-based Mutation Operators for Order-based Stochastic Clique Covering Algorithms. In C. Blum and E. Alba, editors, *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, GECCO 2013, Amsterdam, the Netherlands, 2013. ACM, New York, 495–502.

D. Chalupa. On the Efficiency of an Order-based Representation in the Clique Covering Problem. In T. Soule and J. Moore, editors, *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, GECCO 2012, Philadelphia, PA, USA, 2012. ACM, New York, 353–360.

D. Chalupa. Solving Clique Covering in Very Large Sparse Random Graphs by a Technique Based on k-Fixed Coloring Tabu Search. In M. Middendorf and C. Blum, editors, *Proceedings of the 13th European conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP 2013, Vienna, Austria, 2013. Springer, Berlin / Heidelberg, 238–249.

D. Chalupa, J. Pospíchal. Metaheuristically Optimized Multicriteria Clustering for Medium-Scale Networks. In V. Snášel, A. Abraham and E. S. Corchado, editors, *Soft Computing Models in Industrial and Environmental Applications, 7th International Conference*, SOCO '12, Ostrava, Czech Republic, Advances in Intelligent Systems and Computing, Vol. 188, 2013. Springer, Berlin / Heidelberg, 337–346.

D. Chalupa. Population-based and learning-based metaheuristic algorithms for the graph coloring problem. In N. Krasnogor and P. L. Lanzi, editors, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO 2011, Dublin, Ireland, 2011. ACM, New York, 465–472.

D. Chalupa. Adaptation of a Multiagent Evolutionary Algorithm to NK Landscapes. In C. Blum and E. Alba, editors, *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, GECCO 2013, Amsterdam, the Netherlands, 2013. ACM, New York, 1391–1398.

D. Chalupa and J. Pospíchal. On the Growth of Independent Sets in Scale-free Networks. In *Nostradamus 2014: Prediction, Modeling and Analysis of Complex Systems*, Ostrava, Czech Republic, Advances in Intelligent Systems and Computing, 2014. Springer, Berlin / Heidelberg, to appear.

D. Chalupa. On the Ability of Graph Coloring Heuristics to Find Substructures in Social Networks. *Information Sciences and Technologies Bulletin of ACM Slovakia*, 3(2):51–54, 2011.

D. Chalupa. An optimization strategy based on social insect behavior. In M. Bieliková, I. Černá, T. Gyimóthy, J. Hromkovič, K. Jeffery, R. Královič, M. Vukolić and S. Wolf, editors, *SOFSEM 2011: Student Research Forum*, Nový Smokovec, Slovakia, 2011. Okat, Bratislava, 35–50.

WLAN Power Save by Header Compression and Packet Overhearing Reduction

David Levy*
Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
david.levy@stuba.sk

Abstract

Cellular, laptop, tablet and handheld devices support WLAN technology. In addition to the spectral efficiency and security issues, power consumption is a vital issue in making the usage of these devices widespread. Efforts are underway in improving power conservation in those devices and thus increasing the duration between recharging the battery. Power consumption is mainly due to the transmission of WLAN packets and the receiver listening and receiving packets. Internet packets contain several headers for routing required in a complex network. Those networks are mostly wired networks using TCP/IP protocol. The WLAN transmitter is encapsulating all the TCP/IP headers and adding its own WLAN header. In some cases, the combined headers length represents a significant portion of the total frame length.

This paper focuses on reducing the combined headers length with different compression methods. Smaller headers imply smaller packets which take less time to transmit and therefore activate the power hungry WLAN transmitter for a shorter time. The receiver on time is also reduced using overhearing avoidance techniques. The performance of those new methods is evaluated using both simulation and analytical models. Both the simulation and the analytical WLAN power estimation were developed for this paper.

The simulation and the analytical models concord and show that if those methods are combined, they can provide up to 30% power consumption reduction for high load and many active stations present on the same channel.

*Recommended by thesis supervisor: Assoc. Prof. Ivan Kotuliak. Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on August 22, 2014.

Levy, D. WLAN Power Save by Header Compression and Packet Overhearing Reduction. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 6, No. 3 (2014) 9-18

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Architecture and Design, Wireless communication*; C.2.5 [**Computer Systems Organization**]: Computer-Communication Networks—*Local and Wide-Area Networks, Access schemes*; C.4 [**Computer Systems Organization**]: Performance of Systems; E.4 [**Data**]: Coding and Information Theory—*Data compaction and compression*; I.6.4 [**Computing Methodologies**]: Simulation and modeling—*Model Validation and Analysis*

Keywords

802.11, WLAN, power consumption, analytical model, header compression, unknown header, packet overhearing, throughput

1. Introduction

There are existing improvements to the initially released IEEE 802.11 protocol , with mandatory features only. The WLAN device, also called STA (Station), stays in a less power consuming state when it is assumed that there is no relevant data that is received from the network. Unfortunately these improvements save power for devices that have a rather low average throughput. This is the case for voice applications for example.

As the applications demand for throughput on hand-held devices is exploding (wireless display, gaming, video over IP), the existing methods cannot manage to save enough power anymore. There is a need to extend battery life of devices which, for a long period of time (several hours), have a sustained high throughput.

The goal in this paper is to save power at a STA that has high throughput, as this is not dealt with in the existing power save protocols. The environment is assumed such that also the other STAs on the channel have a rather high throughput.

Two complementing ideas will be developed in this paper:

- Header compression,
- Packet overhearing reduction.

Header compression is most efficient on short packets while packet overhearing avoidance is most effective for longer packets.

For header compression, the idea to reduce the time the STA stays in transmit or receive state is to reduce the frame size by reducing the header size. The header of a frame is, in fact, a series of encapsulated headers. Each header represents the overhead required by a specific protocol. The header represents therefore a substantial part of a typical frame. A typical encapsulation of a part of an Internet page, has the following headers: WLAN header, LLC header, IP header, TCP header and HTTP header. The frame overhead can range from 66 bytes for Voice over IP applications to more than 100 bytes for Internet applications.

The existing header compression methods compress only two known protocols concurrently. The novel compression scheme proposed in this paper will have a lower compression performance but will cover all the Open Systems Interconnection reference model layers, from Physical to Application. The compression scheme does not need to understand the upper layer protocols in order to compress the headers. Only the WLAN header is compressed with protocol knowledge for packet re-ordering and correct channel access.

In order to follow the CSMA/CA protocol which is part of the standard, all STAs stay in the listen state and hear all traffic, including frames that are not addressed to them. Not addressed frames are discarded after they were fully received. A study of Biswas in [3] shows that in a high traffic network, more than 30% of a device energy is spent in receiving packets that are not addressed to it. Therefore, emphasizing on reducing this energy can save substantial energy, overall, for the device. The novel idea in packet overhearing reduction is to reduce the time a STA stays in receive state for acquiring a frame that is not addressed to it.

In order to estimate the performance of the proposed methods, a novel analytical model of a WLAN STA power consumption was developed. The analytical model is derived from Bianchi's throughput analytical model of a WLAN STA developed in [2].

The organization of this paper is as follows: Section 2 will mention existing methods to estimate and save power in WLAN. Section 3 will describe the novel power consumption analytical model. Section 4 will describe the proposed header compression algorithms. Section 5 will explain the power overhearing reduction method. Section 6 will evaluate the performance of the two proposed methods. Section 7 will conclude this paper.

2. Prior Art

This section will mention Prior Art of WLAN power save methods, header compression methods, packet overhearing reduction and throughput analytical models.

There are several amendments to the 802.11 standard that provide mechanism to save power. They are well described in [9]. Those mechanisms try to have the STA stay longer in sleep state instead of listen state. Since the power consumption in the sleep state is much lower than in the listen state , if implemented correctly, these protocols provide a substantial battery life extension. Unfortunately these protocols save power for devices that have a rather low average throughput. In case of sustained high throughput, the device would have to stay anyway in idle state to be able to continuously transmit and/or receive following the CSMA/CA protocol.

The existing header compressions were developed mainly to increase the performance of the IP flows. The older compression methods [7] and [5] used the simple delta compression, sending only the difference in the value of the changing fields, to minimize the number of bits sent. The TCP recovery mechanism was used to recover from channel errors. Newer compression schemes do not use the TCP, anymore, to recover from error conditions. Instead, a dedicated feedback mechanism was defined and allowed compression of other headers such as UDP and RTP. The last development was the ROHC [4] which compresses more types of headers and offers also feedback mechanism for error recovery. ROHC defines some compression algorithms dedicated to specific headers to achieve very high compression efficiency. The compression methods compress only two known protocols concurrently. Unknown headers cannot be compressed and headers on other layers cannot be compressed either.

In order to cover all OSI layer headers, a general compression scheme shall be used, instead of a dedicated method for every layer. A comparison of several general compression algorithms was done in [8]. The non-adaptive Huffmann encoding [6] cannot be used as the headers are unknown. The adaptive Huffmann encoding requires too much processing time compared to the throughput of WLAN. The dictionary methods (as the Lempel-Ziv-Welch) or [12] are inefficient in the first kilo byte of the compressed message, but the headers field we need to compress is less than 200 bytes. The Run-Length Encoding (RLE) performs a lossless compression of input data based on sequences of identical values (runs). The algorithm is quite easy: each run, instead of being represented explicitly, is translated by the encoding algorithm in a pair (l, v) where l is the length of the run and v is the value of the run elements. The longer the run in the sequence to be compressed, the better is the compression ratio. An enhanced RLE was proposed in [1].

The WLAN standard proposes several methods to reduce device power consumption: Legacy power save mode, unscheduled Automatic Power Save Delivery and Power Save Multi-Poll. Other proposed methods go in the same direction to save power for a station which has low traffic [11], [10] and [13]. In all cases, low traffic is required to be able to reduce power consumption.

Biswas proposes, in [3], a way to avoid packet overhearing, relaying on the availability of the Clear To Send (CTS) packet. The address of the destination node in the CTS packet is checked. If it is not the one of the device, then the STA is forced to sleep state for the duration appearing in the CTS packet duration field. This corresponds to the duration of the immediately following data packet. The CTS mechanism was used for backwards compatibility and for hidden node avoidance. But in modern networks there is no need for backwards compatibility anymore and the receivers have much higher sensitivity than the transmitter, sharply reducing the effect of the hidden node. The CTS frame is seldom used as it reduces the system throughput.

In [2], Bianchi developed the WLAN throughput analytical model. His approach revolutionized the analytic study

of networks employing this protocol. His model studies the behavior of a single STA with a discrete time Markov chain. He obtains the probability of transmission of this STA and the probability of collision in steady state. The study is then generalized to all the STAs on the channel, to obtain the overall channel throughput.

3. Analytical Model

This section describes how the STA average power consumption was derived from Bianchi's model. The analytical model is validated with comparison to a WLAN Matlab model, which I also developed.

3.1 Derive power consumption from throughput model

The average slot duration, σ_{avg}, is defined in (1). The terms (denominator of the normalized throughput obtained by Bianchi) are the successful transfer time, failed transfer time due to collision and listen time respectively. Unless clearly defined, all parameters appearing in this section are identical to those defined in [2].

$$\sigma_{avg} = (1 - p_{tr})\sigma + p_{tr}p_s T_s + p_{tr}(1 - p_s)T_c \quad (1)$$

The fraction of time the network spends for a successful transfer, failed transfer and decrementing the back-off (listen time), is defined in equations (2), (3) and (4), respectively.

$$\sigma_s = \frac{p_{tr}p_s T_s}{\sigma_{avg}} \quad (2)$$

$$\sigma_c = \frac{p_{tr}(1 - p_s)T_c}{\sigma_{avg}} \quad (3)$$

$$\sigma_{bo} = \frac{(1 - p_{tr})\sigma}{\sigma_{avg}} \quad (4)$$

These equations need to be translated to fractions of time during which a single STA is in transmit, receive and listen states in order to estimate the average power consumption of a single STA.

The fraction of time a STA is in transmit state, τ_{tx}, is the time it spends to transmit successful packets, colliding packets and Acknowledgments to packets it successfully received. The fraction of time a STA is in receive state, τ_{rx}, is the time it spends to receive successful packets, Acknowledgments and colliding packets. The fraction of time a STA is in listen state, τ_{idle}, is the time it spends to sense the channel (back-off counter decrement), the inter-frame times during successful packet reception and the DIFS time (defined in 802.11 standard) after colliding packets. All those time fractions are function of σ_s, σ_c, σ_{bo}, the header length, the payload length, the number of active STAs on the channel and the time required to transmit successful and colliding packets (including eventual Acknowledgment and Inter Frame Spaces, as defined in 802.11).

The fraction of time in each state can now be used to calculate the average power consumption of a STA:

$$P_{avg} = \tau_{idle}P_{idle} + \tau_{tx}P_{tx} + \tau_{rx}P_{rx}, \quad (5)$$

with P_{idle}, the power consumption of a STA in listen state, P_{tx}, the power consumption of a STA in transmit state and P_{rx}, the power consumption in receive state. Those powers are usually provided by the device manufacturer in their data sheets.

3.2 Validation of the analytical model

The power consumption analytical model developed in section 3.1 needs to be validated. For this purpose I developed internally (in Matlab) a simulation that emulates a network where all the stations follow the CSMA/CA protocol, as defined in 802.11. The correct behavior of the simulator itself was checked by comparing the simulator normalized throughput to the analytical model defined in [2]. Once the throughput is correct, it is guaranteed that the behavior of all the stations on the simulator access the media as defined by the CSMA/CA protocol. Each STA is correctly switching power states (transmit, receive and listen). The power consumption estimation for each station in the simulation is then simply accumulating the energy based on the station power state. The power consumption analytical model was validated by comparing the results to this WLAN network simulator. There were only minor differences, validating the power estimation analytical model.

The developed analytic and Matlab models will be used in following chapters to estimate the performance of the proposed power save methods.

4. Header Compression Algorithm

In this section, I focus on decreasing header size by proposing generic compression algorithms.

In wireless networks, the channel introduces many errors (up to a PER of 10%). If those errors are undetected at the link layer, corrupted frames can be forwarded to the higher layers. When header compression is performed, corrupted headers could cause misinterpretation of data content at the receiver. A single corrupted frame forwarded at the higher layers can therefore cause the loss of several subsequent frames. To avoid this loss, as many errors as possible shall be detected. Due to the number of error bursts generated by the Viterbi decoder, the 32-bit WLAN FCS cannot be trusted for detecting all practical errors. A mechanism needs to be developed to avoid transfers of undetected corrupted packets to the higher layer.

I will use a 16 bit CRC to protect the compressed header alone. The header that can be compressed will be limited to 200 bytes to reduce the probability of multiple bursts on the protected compressed header. The average distance between error bursts, decreases exponentially with the SNR. Compression will be allowed only on links that are assumed to have a good SNR. The receiver first checks the FCS. In case of an invalid FCS the frame is rejected and no Acknowledgment (ACK) frame is sent. If the 16 bit CRC detects error that was not detected by the FCS, then the receiver shall

1. stop sending ACKs to subsequent frames until an uncompressed packet is received,
2. not forward any frame to the upper layer until an uncompressed frame is received.

The general flow is modified to include compression and decompression.

As shown in Figure 1, the compression has to be split to three stages. First, all layers above the WLAN MAC layer

are compressed before encryption. This is needed because the encrypted data looses the repetitive pattern that is visible between consecutive frames. After the MAC layer is added, the second compression stage can be executed on it. Finally, the Duration ID field which is part of the MAC header has to be modified to indicate the compressed packet duration.

The decompression is split to two stages, one for the MAC header and one for the rest of the headers. The frame reordering and decryption must take place before the second decompression stage.

I limit compression to WLAN data frames (excluding control and management frames) in an Infrastructure network. Each WLAN header field is compressed in a different way to optimize compression. By this I achieve close to 50% compression of the WLAN MAC header.

The transmitter separates flows based on the Access Category (AC). Each AC queue is treated independently. The compression consists in comparing the previously successfully transmitted frame with the frame to be sent on the same AC queue. The compressor calculates the difference between the two frames for every byte. The resulting difference is then compressed. The decompressor decompresses the header and rebuilds the newly received frame by adding to it the previously successfully received frame. Two problems need to be considered:

1. What is the optimum compression method, taking into account the limited resources and processing time?
2. The length of the generalized header is unknown, as the protocols are unknown. It is clear, that at a certain point, the application payload data will appear in the frame. From this point on, all bytes will differ, compared to the previous frame. The effect may be an increase in length, instead of the wished decrease, due to many differences between two consecutive frames. What is the optimum number of bytes to compress?

In order to decompress, the compressor must tell the decompressor the length of the compression code. This is pre-pended in the one byte Length field. The 16 bit CRC covers all compressed content, including the compressed WLAN header. The frame format is shown in Figure 2, where the untouched part of the frame is grayed out.

The Bit Stuffed (BS) RLE that was proposed in [1] is compared to a newly proposed method called Fixed Asymmetric (FA) RLE. The FA RLE is a simplification of the BS RLE and biased to compress better the case where no change happened between two consecutive frames. All runs are forced to have the same length. Then it is not more needed to transmit the number of repetitions, since it is *fixed* and known by the receiver.

Since the length of the unknown header is also unknown, algorithms to detect the boundary between the end of the header and the beginning of the payload are also created. The idea is to find the number of bytes to compress that yields the shortest compressed header. The algorithm calculates the header reduction for each additionally compressed header byte, using the BS RLE or the FA RLE,

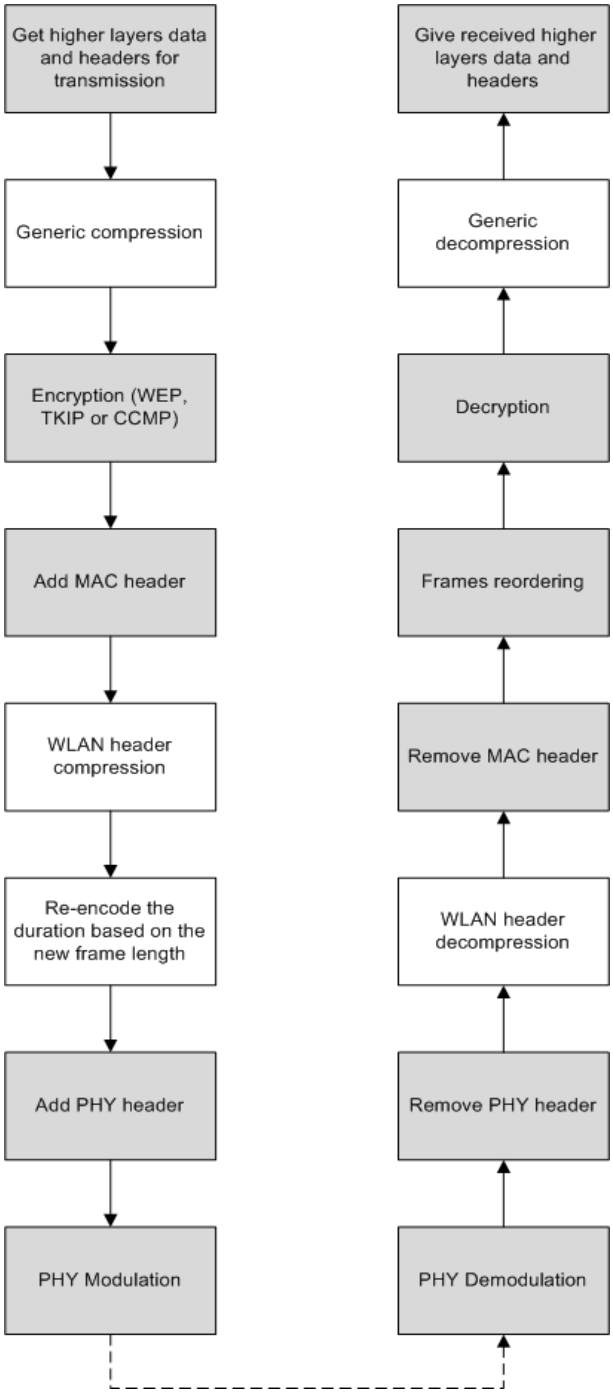

Figure 1: Flow with compression

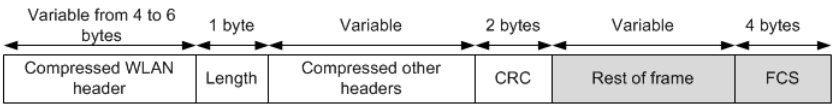

Figure 2: Compressed frame format

therefore it is called *Header Reduction* algorithm. I then obtain a graph that shows the header reduction against the number of compressed bytes (starting from the end of the WLAN MAC header). The peak in this graph is the optimal number of bytes to compress.

At the end of the unknown header length, the difference between consecutive frames will suddenly grow. Therefore, a simpler approach to find the boundary is to perform a moving average of the binary difference. The binary difference between consecutive frames for every byte is 1 if the bytes are different and 0 if they are equal. If the moving average is above a threshold, then the header-data boundary was reached. This algorithm is called the *End of Header Detection* algorithm.

Additionally, I use analytical models for performance evaluation. I developed an analytical model in the simple case, where a single STA is active. This model calculates both the energy and the throughput gained by header compression. The second analytical model for more active STAs is a modified version of the model presented in section 3.1. The only difference is that the header length for packets that are transmitted first time right (not colliding) is reduced.

5. Packet Overhearing Reduction

This section describes a method to reduce the WLAN inherent packet overhearing without the usage of the Request to Send (RTS)/CTS mechanism. For this purpose I introduce a new intermediate state called *semi-doze* state. In this state, a STA consumes less power than in listen/idle state, because it has some of its analog components switch off. But the slowest analog components are not switched off, so that it is possible to return to receive state in micro-seconds (and not milliseconds when waking up from doze state).

The proposed method consists in:

1. detecting early in packet reception that the frame is addressed to another device,
2. calculating the remaining duration of the packet. If this duration is longer than the time it takes for the STA to transition from receive to semi-doze state and back to receive state, then
3. changing to semi-doze state, and,
4. scheduling to order transition back to the receive state such that the STA will finish its transition shortly before the end of the packet.

Like in the method proposed in [3], the gain in power is P_{rx} - $P_{semi-doze}$, where $P_{semi-doze}$ is the power consumption in semi-doze state. The duration of this power gain is not the complete data packet τ_p, as one needs to take into consideration the transitions of the STA from state to state, as shown in Figure 3. Therefore, I talk of packet overhearing *reduction*. The duration of the energy gain is τ_g:

$$\tau_g = \tau_p - \tau_{res} \tag{6}$$

where τ_{res} is the duration of residual overhearing. It can be partitioned to $\tau_{res} = \tau_d + \tau_{sd} + \tau_r + \tau_m$ where τ_d is the time it takes the receiver to detect that the packet is addressed to another STA, τ_{sd} and τ_r are the transition time to semi-doze and and back to receive state, respectively. τ_m is a margin used to take into account the eventual STA to AP clock phase shift.

In order to check the performance of this method, the analytical model defined in section 3.1 is modified. It takes into account also the time a STA is in semi-doze state. Another analytical model is developed for the simple case where only the AP transmits to two STAs. The latter shows an asymptotic gain (for very long frames) as defined in equation 7.

$$G_{max} = \frac{1}{2}\frac{P_{semi-doze}}{P_{rx}} \tag{7}$$

6. Evaluation of proposed methods

The objective of this section is to evaluate the gains achieved by header compression and packet overhearing avoidance. The methods to achieve this objective are both analytical and by simulations. I will first measure on real captured frames the average header reduction that can be achieved by Header compression. This will be used to estimate the energy consumption and throughput gains when the frames have the reduced length. The packet overhearing reduction energy consumption gain will also be estimated. Finally, both methods will be combined to figure out how each of them contributes to save energy and how much can be saved in total. The performance will be estimated when varying the key parameters, as the payload length, the PHY rate and the actively transmitting number of STAs.

6.1 Header compression

6.1.1 Length reduction

Approximately one million frames were captured on heavy loaded networks with different types of applications. This was done on a public network (hot spot) as well as an office network AP. The frame headers were analyzed with a Packet Analyzer. All frames captured were sorted according to AC queue, source and destination and saved in separate files. The difference between two consecutive frames on the same file were calculated, byte by byte.

The captured frames were also compressed with the algorithms defined in section 4, using Perl scripts. The processing time of the Perl scripts was measured to be able to perform a relative comparison of the algorithms processing time.

I choose the algorithm parameters that provide the most header reduction for each case. I add the WLAN Header

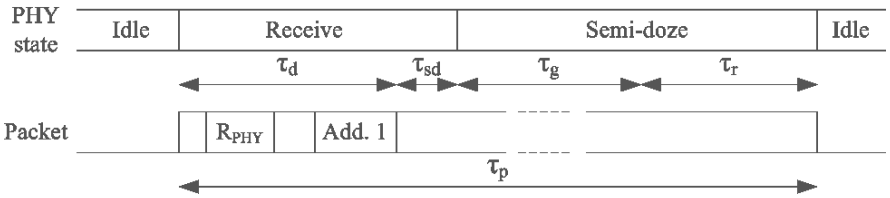

Figure 3: PHY states in a packet

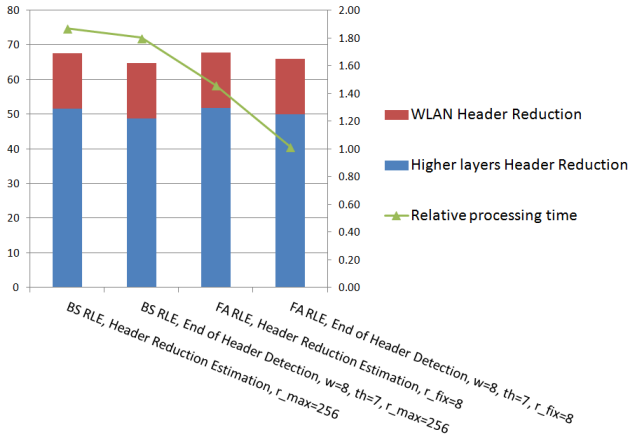

Figure 4: Overall best achieved Absolute Efficiency and processing time

reduction, resulting in the total achieved header reduction. The results are shown on Figure 4, where I depict also the relative processing time (excluding WLAN header compression, as it is identical for all cases). The Header Reduction algorithm detects the end of the header in a more efficient way than the End of Header Detection algorithm. I can see that there are only minor differences between the achieved header reduction; less than 3 bytes. But there are major differences in processing time: it takes almost twice longer to process the BS RLE with Header Reduction Estimation than the newly proposed FA RLE with End of Header Detection.

6.1.2 Power save

The simulations as well as the analytical model are in saturation mode: all STAs always have a new frame to transmit. I see that in this case the reduction in power consumption is negligible. The header compression only achieves to squeeze more payload per unit of time, which does not reduce the average active time. In real life, the user would like to transfer a file, with a finite size. I therefore will consider for the rest of this section the case where all STAs have to transfer a file of finite length.

I firstly vary the number of STAs that transfer their file simultaneously. The result is depicted on Figure 5. The simulation shows that the gain from using compression decreases when the number of active STAs increases. The number of collisions increase with the number of STAs. As the compression is done only on the first transmission attempt, the proportion of compressed frames decreases accordingly, bringing the gain to an asymptotic value of 0, for a big number of STAs. The analytical result was obtained using the simple single STA analytical model. The multiple STAs analytical model cannot be used as it assumes non-stop transfer instead of a finite length file

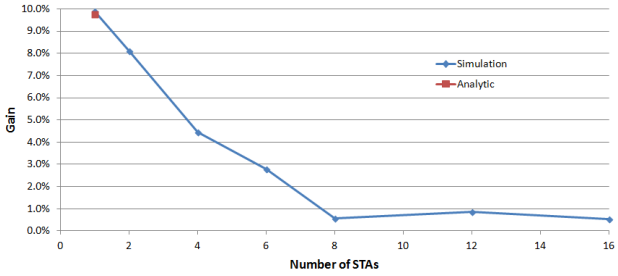

Figure 5: File Transfer with Header compression: power saved vs. number of STAs

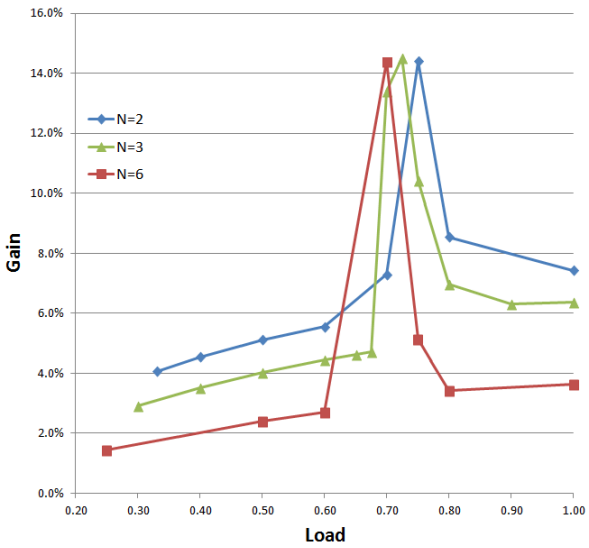

Figure 6: File Transfer with Header compression: power saved vs. load

transfer.

The next effect analyzed is the STAs load. It is possible to imagine, that the STAs do not have the file to transfer ready. The file needs to be preprocessed in chunks, before it is sent. Therefore, the data load seen by the transmitter may be lower. The results are depicted on Figure 6.

At low loads, the state of the STA is most of the time in idle. As the traffic is low, the impact of header compression on saved power is low. With the increasing traffic, the proportion of time in idle diminishes and so the gain achieved increases. At very high loads, the network is in saturation. There is no difference between a load that is slightly above saturation and 100% load; the same throughput is achieved and consequently the header compression power saved stays constant. There are many collisions, but thanks to the backoff counter range increase

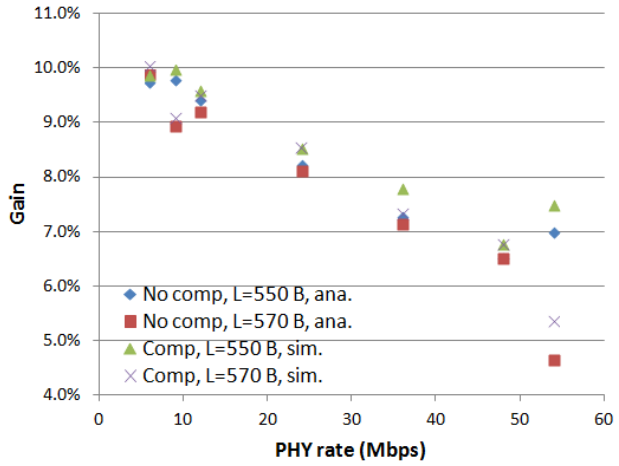

Figure 7: File Transfer with Header compression: power saved vs. PHY rate

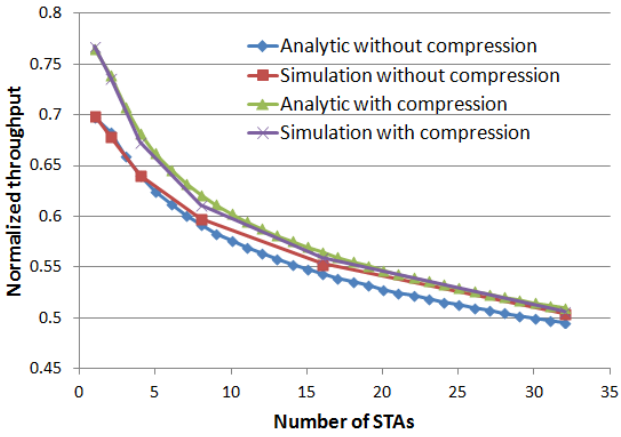

Figure 8: Throughput gain using header compression vs. number of transmitting STAs

(of the other STAs), still some frames are successfully transmitted at the first attempt, providing power saving. In order to understand the peak gain around a load of 0.7, it is necessary to check the throughput gain obtained by header compression on Figure 8. On this figure, it can be seen that if header compression is enabled, the maximum throughput is slightly greater. Therefore, for degenerate distribution (constant interval) packet generation and $N = 3$ STAs (for example), frames can be transferred without collision for a load up to 0.74 with header compression, while collisions start at a load of 0.68 without header compression. Therefore, for a load between 0.68 and 0.74, header compression saves power thanks to two reasons: shorter frames and less collisions. The second reason has a very high impact because frame retransmission consumes a lot of power.

The effect of the PHY rate is depicted in Figure 7. I can see a general behavior of reduced gain with increasing PHY rate. The cause is the OFDM symbol. For each PHY rate, a given number of bytes can be inserted per OFDM symbol. For example, only 3 data bytes are inserted per OFDM symbol, at 6Mbps, while 27 data bytes are inserted per OFDM symbol. Following the 802.11 standard, no partial OFDM symbol, may be transmitted. If the quantity of data bytes in a frame is not an integer number of OFDM symbols, then the last symbol is padded with zero bytes. For higher PHY rates, the average number of padded bytes becomes significant, compared to the header reduction achieved. Part of the bytes gained by header compression are reinserted due to the padding of the last symbol.

For example, at 54 Mbps, a frame of 650 bytes, requires 25 symbols (omitting the PHY header), with the last symbol containing only 2 data bytes and 25 pad bytes. The header compression reduces the header by 70 bytes. This brings the frame to 580 bytes, which require 22 symbols, with the last symbol half full. The achieved gain is 3 symbols. But if the frame length is slightly increased to 670 bytes, then the gain is only 2 symbols, and the last symbol after compression contains 21 padded zero bytes.

The same OFDM symbol padding causes the spread in gain between payload of 550 and 570 bytes, in Figure 7. The spread is wider at higher PHY rates, as the variation between no padding for the last symbol and almost complete padding reaches 26 bytes. This means that for some frame lengths, the header compression will be net 70 bytes, while in other extreme cases, it will only be 44 bytes, at 54Mbps.

6.1.3 Throughput gain

The first impact analyzed is the number of STAs. The results are depicted in Figure 8. I see a sharp decrease in throughput gain as the number of STAs increases. This is due to the increasing probability of collision and the fact that re-transmitted frames do not have their header compressed. When a single STA transmits, the normalized throughput is increased by 0.067 (representing almost 10%).

The effect of the frame length is depicted in Figure 9 for two active STAs. The general behavior is an increase of the throughput for longer frames, as the proportion of overhead to payload decreases. The throughput is increased by 0.075 for short frames (representing 24%), and slowly decreases to 0.035, for frames with 1400 Bytes of payload. This shows, that even for long frames, the header reduction creates a nice maximum throughput increase.

The effect of the header reduction is similar to the effect of the payload length, as it consists in varying the overall frame length. But the impact on the throughput is reduced because the variation on the frame length is limited.

The throughput gain is reduced with increasing PHY rate. The cause is similar to the one described in Section 6.1.2, for the power save gain: the OFDM symbol padding.

6.2 Packet overhearing reduction

This section compares the average power consumption of a standard STA and a STA that implements the packet overhearing reduction method.

Figure 10 shows the power consumption gain when using the packet Overhearing Reduction method in saturation mode with a PHY rate of 6Mbps. The power consumption decreases as the number of STAs increases. This can be explained by the larger average backoff counter value that

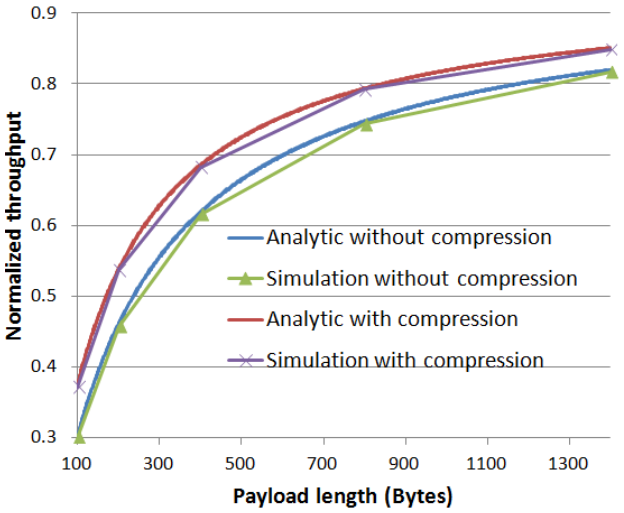

Figure 9: Throughput gain using header compression vs. payload length

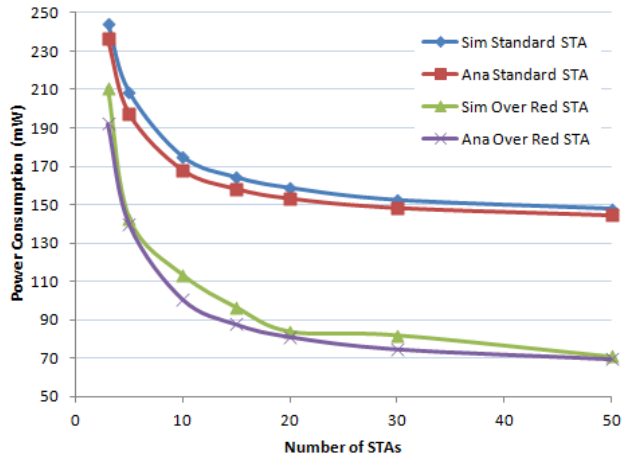

Figure 10: Power vs. N in saturation mode at PHY rate 6Mbps

the STAs use before they are allowed to transmit. Therefore, the STAs spend less time in transmit state (the most power consuming state) and more time in idle state. The amount of retransmissions, per STA, due to collisions in a channel with more STAs is higher. This gives more opportunities for a STA implementing Overhearing Reduction to be in semi-doze state. The power saved at lower PHY rate is larger thanks to the overall packet duration. At 6Mbps, the packets are longer, allowing the STA using Overhearing Reduction to stay longer in semi-doze state.

In Figure 11, the normalized throughput is fixed to 0.33, which is below the saturation throughput for the range of up to $N = 50$ STAs. For N small, the channel is far from saturation, therefore, the proportion of time a STA transmits is low. The standard STA spends therefore less time in receive state, which means also less time in semi-doze state for the STA implementing the Overhearing Reduction method. The result is less power save gain compared to the saturation mode for small N. As N increases, the channel approaches saturation and therefore, the power save gain approaches also the gain in saturation mode.

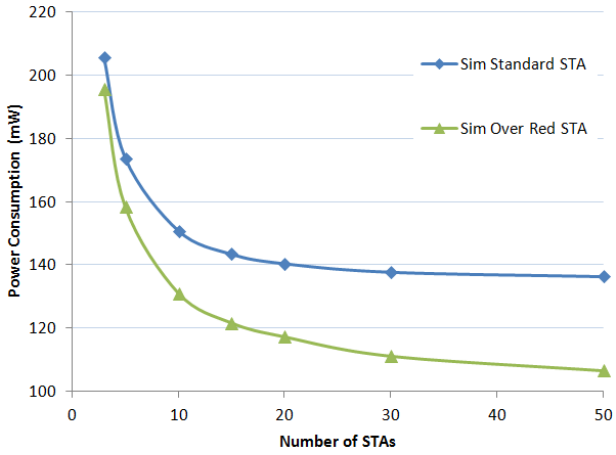

Figure 11: Power vs. N in non-saturation mode at PHY rate 24Mbps

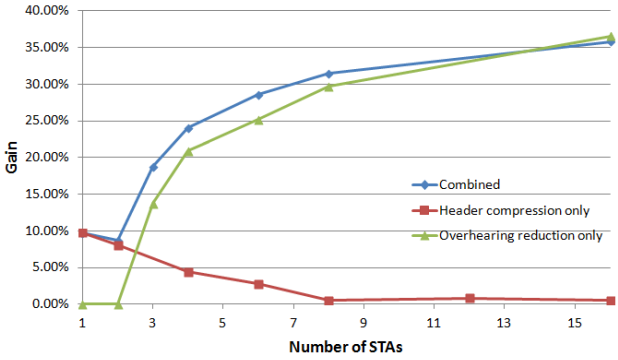

Figure 12: Combined methods, gain vs. number of STAs

6.3 Energy saved when combining both methods

In this section, I combine the methods of Header compression and packet overhearing Reduction. Packet overhearing reduction, by definition, cannot contribute to throughput increase. Therefore, I focus on power consumption analysis. As seen in Section 6.1.3, Header compression energy saving can be evaluated on finite size file transfer only. Therefore, this section will only analyze the energy saving on a file transfer.

The first impact analyzed is the number of STAs. The results are depicted in Figure 12. As seen in Figure 5, the energy saved from Header Compression decreases with the number of STAs actively transmitting. As seen in Fig 10, the gain increases with the number of STA. For $N = 1$ and $N = 2$, there is no overhearing, therefore, Overhearing Reduction is not possible. I can see that the methods complement each other. Where Overhearing Reduction cannot help, Header Compression provides its best performance. Where Header Compression cannot provide much energy saving, Overhearing Reduction takes over.

Based on the findings of Figure 12, when there are only one or two STAs active, only Header Compression brings energy saving. When there are many STAs active ($N > 8$), mainly Overhearing Reduction brings energy saving. For these cases, the advantages for Header Compression alone or Overhearing Reduction alone, are described in

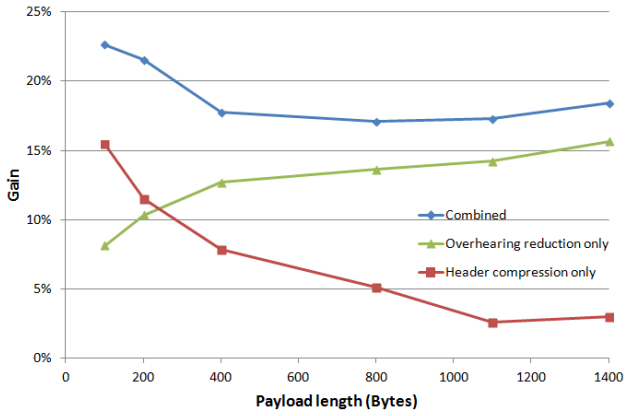

Figure 13: Combined methods, gain vs. payload length

Section 6.1.2 and Section 6.2 respectively. In this section, I will concentrate on the case where both methods contribute to the energy saving. I will, therefore, fix $N = 3$, to have approximately the same contribution from both methods, for the next simulations in this section.

The impact of the frame length is depicted in Figure 13. For Header Compression, the gain in energy decreases with the increased Payload length, as the proportion of the header to the payload decreases. For the Packet Overhearing Reduction, the longer the frame, the more energy can be saved, as the STA stays longer in semi-doze state, and less in receive state. I can see that also for Payload length, the methods complement each other, allowing for an almost constant gain over the complete Payload range.

Both methods have their gain reduced with reduced load. The reduced load translates to proportionally more time in PHY listen state compared to active states. Therefore, the gain achieved by header compression or packet overhearing reduction is negligible because it mainly reduces the time in PHY active states (transmit or receive).

7. Conclusion

The WLAN is a very popular way to transfer data from and to mobile devices at high rate without wires. In order to allow the user maximum flexibility, the interval between consecutive battery recharges should be increased as much as possible. This goal is achieved by reducing the energy consumption during data transfer. The goal of this paper was to define two methods to save energy during high data traffic. Namely, packet overhearing avoidance and all-layers header compression.

In order to estimate the energy saved by the proposed method, I developed an analytical model of a standard WLAN STA energy consumption. This model is derived from Bianchi's throughput model [2]. The analytical model was validated against an internally developed CSMA/CA simulator (in Matlab) and Bianchi's throughput model. I also modified this standard STA analytical model to support the header compression method and the packet overhearing reduction method.

I identified the cases to keep the probability low to have undetected errors in the headers. I defined an efficient compression of each WLAN MAC header field. I used the difference between consecutive frame of the same AC to compress unknown headers. The BS RLE algorithm was clearly defined and I introduced a new algorithm called the FA RLE algorithm. The new notion of boundary between unknown header and payload was introduced and three algorithms to identify this boundary were given. An analytical model was developed for the case of unidirectional communication between the AP and a single STA.

A method was described to reduce packet overhearing without the need of RTS/CTS frames. For this purpose, a new PHY state was introduced which is a compromise between the too slow doze state and the too much energy consuming idle state. I developed an analytical model considering the case where only the AP transmits frames to two STAs, meaning that no collision occurs. This model showed the asymptotic behavior of the energy gain.

The gains achieved by the two proposed methods was proven. A big amount of frames was captured on which the header compression algorithms were run. This gave the average expected frame size reduction that can be realistically achieved. I then used both the analytical model and the simulator to estimate the throughput gain and energy saved by reducing the frame size, thanks to header compression. The same tools were used to estimate the energy saved thanks to packet overhearing reduction. By combining both methods it could be seen how they complement each other. For small number of STAs transmitting, the header compression contributes more, while for longer frames, the packet overhearing contributes more. It is remarkable that at high load and low PHY rate, over a wide range of the critical remaining parameters, the achieved gain is above 15% if both methods are combined.

Acknowledgments. This work has been supported by Slovak National Research Grant Agency under project no. VEGA 1/0676/12. It is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005 and for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF.

References

[1] A. Amin, H. Qureshi, M. Junaid, M. Habib, and W. Anjum. Modified run length encoding scheme with introduction of bit stuffing for efficient data compression. *International Conference on Internet Technology and Secured Transactions*, 1:668–672, December 2011.

[2] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE J. Sel. Area Comm.*, 18(3):535–547, 2000.

[3] S. Biswas and S. Datta. Reducing overhearing energy in 802.11 networks by low-power interface idling. *IEEE International Conference on Performance, Computing and Communications*, pages 695–700, 2004.

[4] Bormann. *RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed*. Network Working Group, 2001.

[5] S. Casner and V. Jacobson. *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*. Network Working Group, 1999.

[6] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E.*, page 1098–1102, September 1952.

[7] V. Jacobson. *Compressing TCP/IP Headers for Low-Speed Serial Links*. Network Working Group, 1990.

[8] S. Kodituwakku and U. Amarasinghe. Comparison of lossless data compression algorithms for text data. *Indian Journal of Computer Science and Engineering*, 1(4):416–425, 2010.

[9] B. McFarland. How to use optional wireless power-save protocols to dramatically reduce power consumption. *EETimes magazine*, May 2008.

[10] R. Palit, K. Naik, and A. Singh. Impact of packet aggregation on energy consumption in smartphones. *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 589–594, July 2011.

[11] S. Tang, H. Yomo, Y. Kondo, and S. Obana. Wake-up receiver for radio-on-demand wireless lans. *EURASIP Journal on Wireless Communications and Networking*, February 2012.

[12] T. A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, June 1984.

[13] Z. Zeng, Y. Gao, and P. Kumar. Sofa: A sleep-optimal fair-attention scheduler for the power-saving mode of wlans. *International Conference on Distributed Computing Systems*, pages 87–98, June 2011.

Selected Papers by the Author

D. Levy, I. Kotuliak. WLAN power consumption analytical model. In *Wireless and Mobile Networking Conference*, pages 101–106, Bratislava, Slovakia, 2012.

D. Levy, I. Kotuliak. General all-Layers combined with Efficient WLAN MAC Layer Headers Compression. In *International Conference on Advances in Mobile Computing and Multimedia*, LNCS 5164, pages 123–132, Vienna, Austria, 2013.

D. Levy, I. Kotuliak. WLAN Power Saving Using Packet Overhearing Reduction. In *Telecommunication Systems Journal*, accepted for publication in Springer.

D. Levy. DSI3 Sensor to Master Current Threshold Adaptation for Pattern Recognition. In *Int. Conf. on Signal Processing Systems*, Sydney, Australia, 2013, obtained the "Excellent Paper Award". Published in *Int. Journal of Signal Processing Systems*, pages 141–145. Engineering and Technology Publishing, 2013.

D. Levy. DSI3 Sensor to Master Decoder using Symbol Pattern Recognition. In Society of Automobile Engineers, editors, *Int. J. Passeng. Cars – Electron. Electr. Syst.*. SAE, Warrendale (PA, USA), 2014.

One-Sided Random Context Grammars

Petr Zemek*
Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 1/2, 612 66 Brno, Czech Republic
izemek@fit.vutbr.cz

Abstract

This extended abstract of the doctoral thesis introduces the notion of a *one-sided random context grammar* as a context-free-based regulated grammar, in which a set of *permitting symbols* and a set of *forbidding symbols* are attached to every rule, and its set of rules is divided into the set of *left random context rules* and the set of *right random context rules*. A left random context rule can rewrite a nonterminal if each of its permitting symbols occurs to the left of the rewritten symbol in the current sentential form while each of its forbidding symbols does not occur there. A right random context rule is applied analogically except that the symbols are examined to the right of the rewritten symbol.

The thesis is divided into three parts. The first part gives a motivation behind introducing one-sided random context grammars and places all the covered material into the scientific context. Then, it gives an overview of formal language theory and some of its lesser-known areas that are needed to fully grasp the upcoming topics.

The second part forms the heart of the thesis. It formally defines one-sided random context grammars and studies them from many points of view. Generative power, relations to other types of grammars, reduction, normal forms, leftmost derivations, parsing-related and generalized versions all belong between the studied topics.

The final part of the thesis closes its discussion by adding remarks regarding its coverage. More specifically, these remarks concern application perspectives, bibliography, and open problem areas.

Categories and Subject Descriptors

F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Classes defined by grammars or automata, Operations on languages*

Keywords

formal language theory, regulated grammars, random context grammars, one-sided random context grammars, permitting grammars, forbidding grammars, reduction, generative power, normal forms, leftmost derivations, generalized versions, LL versions

1. Introduction

Formal languages, such as programming languages, are applied in a great number of scientific disciplines, ranging from biology through linguistics up to informatics (see [20]). As obvious, to use them properly, they have to be precisely specified in the first place. Most often, they are defined by mathematical models with finitely many rules by which they rewrite sequences of symbols, called strings.

Over its history, formal language theory has introduced a great variety of these language-defining models. Despite their diversity, they can be classified into two basic categories—generative and recognition language models. Generative models, better known as *grammars*, define strings of their language so their rewriting process generates them from a special start symbol. On the other hand, recognition models, better known as *automata*, define strings of their language by a rewriting process that starts from these strings and ends in a special set of strings, usually called final configurations.

Concerning grammars, the classical theory of formal languages has often classified all grammars into two fundamental categories—*context-free grammars* and *non-context-free grammars*. As their name suggests, context-free grammars are based upon context-free rules, by which these grammars rewrite symbols regardless of the context surrounding them. As opposed to them, non-context-free grammars rewrite symbols according to context-dependent rules, whose application usually depends on rather strict conditions placed upon the context surrounding the rewritten symbols, and such a way of context-dependent rewriting often makes them clumsy and inapplicable in practice. From this point of view, we obviously always prefer using context-free grammars, but they have their drawbacks, too. Perhaps most importantly, context-free grammars are significantly less powerful than non-context-free grammars. Considering all these pros and cons, it

*Recommended by thesis supervisor: Prof. Alexander Meduna. Defended at Faculty of Information Technology, Brno University of Technology on September 12, 2014.

Zemek, P.: One-Sided Random Context Grammars. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 6, No. 3 (2014) 19-28

comes as no surprise that modern formal language theory has intensively and systematically struggled to come with new types of grammars that are underlined by context-free rules, but which are more powerful than ordinary context-free grammars. Regulated versions of context-free grammars, briefly referred to as *regulated grammars* in the thesis, represent perhaps the most successful and significant achievement in this direction. They are based upon context-free grammars extended by additional regulating mechanisms by which they control the way the language generation is performed.

Over the last four decades, formal language theory has introduced an investigated many types of regulated grammars (see [3, 11, 18], Chapter 13 of [9], and Chapter 3 of the second volume of [20] for an overview of the most important results). Arguably, one of the most studied type of regulated grammars are random context grammars, which are central to the thesis.

Random Context Grammars

In essence, *random context grammars* (see Section 1.1 in [3]) regulate the language generation process so they require the presence of some prescribed symbols and, simultaneously, the absence of some others in the rewritten sentential forms. More precisely, random context grammars are based upon context-free rules, each of which may be extended by finitely many *permitting* and *forbidding nonterminal symbols*. A rule like this can rewrite the current sentential form provided that all its permitting symbols occur in the sentential form while all its forbidding symbols do not occur there.

Random context grammars are significantly stronger than ordinary context-free grammars. In fact, they characterize the family of recursively enumerable languages (see Theorem 1.2.5 in [3]), and this computational completeness obviously represents their indisputable advantage. Also, *propagating random context grammars*, which do not have any erasing rules—that is, rules with the empty string on their right-hand sides—are stronger than context-free grammars. However, they are strictly less powerful than context-sensitive grammars. Indeed, they generate a language family that is strictly included in the family of context sensitive languages (see Theorem 1.2.4 in [3]).

From a pragmatical standpoint, however, random context grammars have a drawback consisting in the necessity of scanning the current sentential form in its entirety during every single derivation step. From this viewpoint, it is highly desirable to modify these grammars so they scan only a part of the sentential form, yet they keep their computational completeness. *One-sided random context grammars*—the topic of the thesis—represent a modification like this.

One-Sided Random Context Grammars

Specifically, in every one-sided random context grammar, the set of rules is divided into the set of *left random context rules* and the set of *right random context rules*. When applying a left random context rule, the grammar checks the existence and absence of its permitting and forbidding symbols, respectively, only in the prefix to the left of the rewritten nonterminal in the current sentential form. Analogously, when applying a right random context rule, it checks the existence and absence of its permitting and forbidding symbols, respectively, only in the suffix to the right of the rewritten nonterminal. Otherwise, it works just like any ordinary random context grammar.

As the main result of the thesis, we demonstrate that propagating versions of one-sided random context grammars, which possess no erasing rules, characterize the family of context-sensitive languages, and with erasing rules, they characterize the family of recursively enumerable languages.

Furthermore, we discuss the generative power of several special cases of one-sided random context grammars. In a greater detail, we prove that *one-sided permitting grammars*, which have only permitting rules, are more powerful than context-free grammars; on the other hand, they are no more powerful than so-called scattered context grammars (see [10]). *One-sided forbidding grammars*, which have only forbidding rules, are equivalent to so-called selective substitution grammars (see [7]). Finally, *left forbidding grammars*, which have only left-sided forbidding rules, are only as powerful as context-free grammars.

Apart from the generative power of one-sided random context grammars and their special cases, we investigate the following aspects of these grammars. First, we establish four normal forms of one-sided random context grammars, in which all rules satisfy some prescribed properties or format. Then, we study a reduction of one-sided random context grammars with respect to the number of nonterminals and rules. After that, we place three leftmost derivation restrictions on one-sided random context grammars and investigate their generative power. We also study generalized versions of one-sided random context grammars, in which strings of symbols rather than single symbols can be required or forbidden. Finally, we study one-sided random context grammars from a more practical viewpoint by investigating their parsing-related variants.

To give a summary, the thesis is primarily and principally meant as a theoretical treatment of one-sided random context grammars, which represent a modification of random context grammars. Apart from this theoretical treatment, however, we also cover some application perspectives to give the reader ideas about their applicability in practice.

Motivation

Taking into account the definition of one-sided random context grammars and all the results sketched above, we see that these grammars may fulfill an important role in the language theory and its applications for the following four reasons.

(I) From a practical viewpoint, one-sided random context grammars examine the existence of permitting symbols and the absence of forbidding symbols only within a portion of the current sentential form while ordinary random context grammars examine the entire current sentential form. As a result, the one-sided versions of these grammars work in a more economical and, therefore, efficient way than the ordinary versions. Moreover, one-sided random context grammars provide a finer control over the regulation process. Indeed, the designer of the grammar may select whether the presence or absence of symbols is examined to the left or to the right. In the case of

ordinary random context grammars, this selection cannot be done since they scan the sentential forms in their entirety.

(II) The one-sided versions of propagating random context grammars are stronger than ordinary propagating random context grammars. Indeed, the language family defined by propagating random context grammars is properly included in the family of context-sensitive languages (see Theorem 1.2.4 in [3]). One-sided random context grammars are as powerful as ordinary random context grammars. These results come as a surprise because one-sided random context grammars examine only parts of sentential forms as pointed out in (I) above.

(III) Left forbidding grammars were introduced in [4], which also demonstrated that these grammars only define the family of context-free languages (see Theorem 1 in [4]). It is more than natural to generalize left forbidding grammars to one-sided forbidding grammars, which are stronger than left forbidding grammars (see Theorem 8 in this paper). As a matter of fact, even *propagating left permitting grammars*, introduced in [2], are stronger than left forbidding grammars because they define a proper superfamily of the family of context-free languages (see Theorem 10 in this paper). We also generalize left permitting grammars to one-sided permitting grammars and study their properties.

(IV) In the future, one might find results achieved in the thesis useful when attempting to solve some well-known open problems. Specifically, recall that every propagating scattered context grammar can be turned to an equivalent context-sensitive grammar (see Theorem 3.21 in [10]), but it is a longstanding open problem whether these two kinds of grammars are actually equivalent—the *PSC = CS problem* (see [10]). If in the future one proves that the propagating versions of one-sided permitting grammars and one-sided random context grammars are equivalent, then so are propagating scattered context grammars and context-sensitive grammars (see Theorem 9 in this paper), so the PSC = CS problem would be solved.

Organization

This extended abstract of the thesis is divided into five sections. After this introductory Section 1, Section 2 briefly reviews the needed notions from formal language theory. Then, Section 3 defines one-sided random context grammars and illustrates them by examples.

Section 4 represents the heart of this extended abstract. Indeed, it gives an overview of the established results and investigated topics concerning one-sided random context grammars. Generative power, relations to other types of grammars, reduction, normal forms, leftmost derivations, parsing-related and generalized versions all belong between the studied topics.

Section 5 closes the text by making several final remarks concerning the covered material with a special focus on its future developments. It concerns application perspectives of one-sided random context grammars, bibliographic comments, and open problem areas.

2. Preliminaries

We assume that the reader is familiar with formal language theory (see [20]). For a set Q, $\mathrm{card}(Q)$ denotes the cardinality of Q, and 2^Q denotes the power set of Q. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. For $x \in V^*$, $|x|$ denotes the length of x and $\mathrm{alph}(x)$ denotes the set of symbols occurring in x.

A *random context grammar* (see Section 1.1 in [3]) is a quadruple, $G = (N, T, P, S)$, where N and T are two disjoint alphabets of *nonterminals* and *terminals*, respectively, $S \in N$ is the *start symbol*, and $P \subseteq N \times (N \cup T)^* \times 2^N \times 2^N$ is a finite relation, called the set of *rules*. Set $V = N \cup T$. Each rule $(A, x, U, W) \in P$ is written as $(A \to x,\ U,\ W)$. The *direct derivation relation* over V^*, symbolically denoted by $\Rightarrow_G$, is defined as follows: if $u, v \in V^*$, $(A \to x,\ U,\ W) \in P$, $U \subseteq \mathrm{alph}(uAv)$, and $W \cap \mathrm{alph}(uAv) = \emptyset$, then $uAv \Rightarrow_G uxv$. U is called the *permitting context* and W is called the *forbidding context*. Let $\Rightarrow_G^*$ denote the reflexive-transitive closure of $\Rightarrow_G$. The *language of* G is denoted by $L(G)$ and defined as $L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}$.

Let $G = (N, T, P, S)$ be a random context grammar. Rules of the form $(A \to \varepsilon, U, W)$ are called *erasing rules*. If $(A \to x,\ U,\ W) \in P$ implies that $|x| \geq 1$, then G is a *propagating random context grammar*. If $(A \to x,\ U,\ W) \in P$ implies that $W = \emptyset$, then G is a *permitting grammar*. If $(A \to x,\ U,\ W) \in P$ implies that $U = \emptyset$, then G is a *forbidding grammar*. By analogy with propagating random context grammars, we define a *propagating permitting grammar* and a *propagating forbidding grammar*, respectively.

Denotation of Language Families

Throughout the rest of this paper, the language families under discussion are denoted in the following way. **RC**, **P**, and **F** denote the language families generated by random context grammars, permitting grammars, and forbidding grammars, respectively. The notation with the upper index $-\varepsilon$ stands for the corresponding propagating family. For example, $\mathbf{RC}^{-\varepsilon}$ denotes the family of languages generated by propagating random context grammars. **CF**, **CS**, and **RE** denote the families of context-free languages, context-sensitive languages, and recursively enumerable languages, respectively. $\mathbf{SC}^{-\varepsilon}$, **S**, and $\mathbf{S}^{-\varepsilon}$ denote the language families generated by propagating scattered context grammars (see [5]), selective substitution grammars (see [8, 19]), and propagating selective substitution grammars—that is, selective substitution grammars without erasing rules—, respectively.

3. Definitions and Examples

Next, we formally define one-sided random context grammars and their variants. In addition, we illustrate them by examples.

Definition 1. A *one-sided random context grammar* is a quintuple

$$G = \big(N, T, P_L, P_R, S\big)$$

where N and T are two disjoint alphabets, $S \in N$, and

$$P_L, P_R \subseteq N \times \big(N \cup T\big)^* \times 2^N \times 2^N$$

are two finite relations. Set $V = N \cup T$. The components V, N, T, P_L, P_R, and S are called the *total alphabet*, the

alphabet of *nonterminals*, the alphabet of *terminals*, the set of *left random context rules*, the set of *right random context rules*, and the *start symbol*, respectively. Each $(A, x, U, W) \in P_L \cup P_R$ is written as

$$(A \to x, U, W)$$

For $(A \to x,\ U,\ W) \in P_L$, U and W are called the *left permitting context* and the *left forbidding context*, respectively. For $(A \to x,\ U,\ W) \in P_R$, U and W are called the *right permitting context* and the *right forbidding context*, respectively. □

When applying a left random context rule, the grammar checks the existence and absence of its permitting and forbidding symbols, respectively, only in the prefix to the left of the rewritten nonterminal in the current sentential form. Analogously, when applying a right random context rule, it checks the existence and absence of its permitting and forbidding symbols, respectively, only in the suffix to the right of the rewritten nonterminal. The following definition states this formally.

Definition 2. Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a one-sided random context grammar. The *direct derivation relation* over V^* is denoted by $\Rightarrow_G$ and defined as follows. Let $u, v \in V^*$ and $(A \to x,\ U,\ W) \in P_L \cup P_R$. Then,

$$uAv \Rightarrow_G uxv$$

if and only if

$$(A \to x, U, W) \in P_L, U \subseteq \text{alph}(u), \text{ and } W \cap \text{alph}(u) = \emptyset$$

or

$$(A \to x, U, W) \in P_R, U \subseteq \text{alph}(v), \text{ and } W \cap \text{alph}(v) = \emptyset$$

Let $\Rightarrow_G^*$ denote the reflexive-transitive closure of $\Rightarrow_G$. □

The language generated by a one-sided random context grammar is defined as usual—that is, it consists of strings over the terminal alphabet that can be generated from the start symbol.

Definition 3. Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a one-sided random context grammar. The *language of* G is denoted by $L(G)$ and defined as

$$L\big(G\big) = \big\{w \in T^* \mid S \Rightarrow_G^* w\big\}$$ □

Next, we define several special variants of one-sided random context grammars.

Definition 4. Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a one-sided random context grammar. Rules of the form $(A \to \varepsilon,\ U,\ W)$ are called *erasing rules*. If $(A \to x,\ U,\ W) \in P_L \cup P_R$ implies that $|x| \geq 1$, then G is a *propagating one-sided random context grammar*. If $(A \to x,\ U,\ W) \in P_L \cup P_R$ implies that $W = \emptyset$, then G is a *one-sided permitting grammar*. If $(A \to x, U, W) \in P_L \cup P_R$ implies that $U = \emptyset$, then G is a *one-sided forbidding grammar*. By analogy with propagating one-sided random context grammars, we define a *propagating one-sided permitting grammar* and a *propagating one-sided forbidding grammar*, respectively. □

Definition 5. Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a one-sided random context grammar. If $P_R = \emptyset$, then G is a *left random context grammar*. If $P_R = \emptyset$ and $(A \to x,\ U,\ W) \in P_L$ implies that $W = \emptyset$, then G is a *left permitting grammar* (see [2]). If $P_R = \emptyset$ and $(A \to x,\ U,\ W) \in P_L$ implies that $U = \emptyset$, then G is a *left forbidding grammar* (see [4]). Their propagating versions are defined analogously as the propagating version of one-sided random context grammars. □

Next, we illustrate the above definitions by three examples.

Example 1. Consider the following one-sided random context grammar

$$G = \big(\{S, A, B, \bar{A}, \bar{B}\}, \{a, b, c\}, P_L, P_R, S\big)$$

where P_L contains the next four rules

$$(S \to AB, \emptyset, \emptyset) \qquad (\bar{B} \to B, \{A\}, \emptyset)$$
$$(B \to b\bar{B}c, \{\bar{A}\}, \emptyset) \qquad (B \to \varepsilon, \emptyset, \{A, \bar{A}\})$$

and P_R contains the next three rules

$$(A \to a\bar{A}, \{B\}, \emptyset) \qquad (A \to \varepsilon, \{B\}, \emptyset)$$
$$(\bar{A} \to A, \{\bar{B}\}, \emptyset)$$

It is rather easy to see that every derivation that generates a nonempty string of $L(G)$ is of the form

$$\begin{aligned} S &\Rightarrow_G AB \\ &\Rightarrow_G a\bar{A}B \\ &\Rightarrow_G a\bar{A}b\bar{B}c \\ &\Rightarrow_G aAb\bar{B}c \\ &\Rightarrow_G aAbBc \\ &\Rightarrow_G^* a^nAb^nBc^n \\ &\Rightarrow_G a^nb^nBc^n \\ &\Rightarrow_G a^nb^nc^n \end{aligned}$$

where $n \geq 1$. The empty string is generated by

$$S \Rightarrow_G AB \Rightarrow_G B \Rightarrow_G \varepsilon$$

Based on the previous observations, we see that G generates the non-context-free language $\{a^nb^nc^n \mid n \geq 0\}$. □

Example 2. Consider $K = \{a^nb^mc^m \mid 1 \leq m \leq n\}$. This non-context-free language is generated by the one-sided permitting grammar

$$G = \big(\{S, A, B, X, Y\}, \{a, b, c\}, P_L, \emptyset, S\big)$$

with P_L containing the following seven rules

$$(S \to AX,\ \emptyset,\ \emptyset) \qquad (X \to bc,\ \emptyset,\ \emptyset)$$
$$(A \to a,\ \emptyset,\ \emptyset) \qquad (X \to bYc,\ \{B\},\ \emptyset)$$
$$(A \to aB,\ \emptyset,\ \emptyset) \qquad (Y \to X,\ \{A\},\ \emptyset)$$
$$(B \to A,\ \emptyset,\ \emptyset)$$

Notice that G is, in fact, a propagating left permitting grammar. Observe that $(X \to bYc,\ \{B\},\ \emptyset)$ is applicable if B, produced by $(A \to aB,\ \emptyset,\ \emptyset)$, occurs to the left of X in the current sentential form. Similarly, $(Y \to X,\ \{A\},\ \emptyset)$ is applicable if A, produced by $(B \to A,\ \emptyset,\ \emptyset)$, occurs to the left of Y in the current sentential form. Consequently, we see that every derivation that generates $w \in L(G)$ is

of the form

$$\begin{array}{lll} S & \Rightarrow_G & AX \\ & \Rightarrow_G^* & a^u AX \\ & \Rightarrow_G & a^{u+1} BX \\ & \Rightarrow_G & a^{u+1} BbYc \\ & \Rightarrow_G & a^{u+1} AbYc \\ & \Rightarrow_G^* & a^{u+1+v} AbYc \\ & \Rightarrow_G & a^{u+1+v} AbXc \\ & \vdots & \\ & \Rightarrow_G^* & a^{n-1} Ab^{m-1} Xc^{m-1} \\ & \Rightarrow_G & ab^{m-1} Xc^{m-1} \\ & \Rightarrow_G & a^n b^m c^m = w \end{array}$$

where $u, v \geq 0$, $1 \leq m \leq n$. Hence, $L(G) = K$. □

Example 3. Consider the one-sided forbidding grammar

$$G = \big(\{S, A, B, A', B', \bar{A}, \bar{B}\}, \{a, b, c\}, P_L, P_R, S\big)$$

where P_L contains the following five rules

$$(S \to AB, \emptyset, \emptyset) \qquad (B' \to B, \emptyset, \{A'\})$$
$$(B \to bB'c, \emptyset, \{A, \bar{A}\}) \qquad (\bar{B} \to \varepsilon, \emptyset, \{\bar{A}\})$$
$$(B \to \bar{B}, \emptyset, \{A, A'\})$$

and P_R contains the following four rules

$$(A \to aA', \emptyset, \{B'\}) \qquad (A' \to A, \emptyset, \{B\})$$
$$(A \to \bar{A}, \emptyset, \{B'\}) \qquad (\bar{A} \to \varepsilon, \emptyset, \{B\})$$

Notice that every derivation that generates a nonempty string of $L(G)$ is of the form

$$\begin{array}{lll} S & \Rightarrow_G & AB \\ & \Rightarrow_G & aA'B \\ & \Rightarrow_G & aA'bB'c \\ & \Rightarrow_G & aAbB'c \\ & \Rightarrow_G & aAbBc \\ & \Rightarrow_G^* & a^n Ab^n Bc^n \\ & \Rightarrow_G & a^n \bar{A} b^n Bc^n \\ & \Rightarrow_G & a^n \bar{A} b^n \bar{B} c^n \\ & \Rightarrow_G & a^n b^n \bar{B} c^n \\ & \Rightarrow_G & a^n b^n c^n \end{array}$$

where $n \geq 1$. The empty string is generated by

$$S \Rightarrow_G AB \Rightarrow_G \bar{A}B \Rightarrow_G \bar{A}\bar{B} \Rightarrow_G \bar{B} \Rightarrow_G \varepsilon$$

Based on the previous observations, we see that G generates the non-context-free language $\{a^n b^n c^n \mid n \geq 0\}$. □

Denotation of Language Families

Throughout the rest of this paper, the language families under discussion are denoted in the following way. **ORC**, **OP**, and **OF** denote the language families generated by one-sided random context grammars, one-sided permitting grammars, and one-sided forbidding grammars, respectively. **LRC**, **LP**, and **LF** denote the language families generated by left random context grammars, left permitting grammars, and left forbidding grammars, respectively.

The notation with the upper index $-\varepsilon$ stands for the corresponding propagating family. For example, $\mathbf{ORC}^{-\varepsilon}$ denotes the family of languages generated by propagating one-sided random context grammars.

4. Results

In this section, we give an overview of the established results and studied topics concerning one-sided random context grammars.

The present section is divided into four subsections. Section 4.1 studies the generative power of one-sided random context grammars. Section 4.2 establishes four normal forms of these grammars. Section 4.3 investigates their descriptional complexity. Finally, Section 4.4 briefly mentions other topics related to one-sided random context grammars that have been investigated in the thesis.

4.1 Generative Power

First, we have investigated the generative power of one-sided random context grammars. In the thesis, it is proved that one-sided random context grammars characterize the family of recursively enumerable languages, and that their propagating versions characterize the family of context-sensitive languages.

Theorem 1. $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$ *and* $\mathbf{ORC} = \mathbf{RE}$

Since $\mathbf{RC}^{-\varepsilon} \subset \mathbf{CS}$ and $\mathbf{RC} = \mathbf{RE}$ (see [3]), we have that one-sided random context grammars are equally powerful as random context grammars, while propagating one-sided random context grammars are more powerful than propagating random context grammars.

Theorem 2. $\mathbf{RC}^{-\varepsilon} \subset \mathbf{ORC}^{-\varepsilon} \subset \mathbf{RC} = \mathbf{ORC}$

Next, we consider one-sided forbidding grammars. In the thesis, it is proved that they have the same power as selective substitution grammars (see [8, 19]).

Theorem 3. $\mathbf{OF}^{-\varepsilon} = \mathbf{S}^{-\varepsilon}$ *and* $\mathbf{OF} = \mathbf{S}$

It is not known whether one-sided forbidding grammars or selective substitution grammars characterize the family of recursively enumerable languages. Also, it is not known whether these grammars without erasing rules characterize the family of context-sensitive languages.

Moreover, the thesis proves the following two results concerning the generative power of one-sided forbidding grammars, where the set of left random context rules coincides with the set of right random context rules.

Theorem 4. *A language K is context-free if and only if there is a one-sided forbidding grammar, $G = (N, T, P_L, P_R, S)$, satisfying $K = L(G)$ and $P_L = P_R$.*

Theorem 5. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided forbidding grammar satisfying $P_L = P_R$. Then, there is a propagating one-sided forbidding grammar H such that $L(H) = L(G) - \{\varepsilon\}$.*

One-sided forbidding grammars are at least as powerful as forbidding grammars. This is stated in the next theorem.

Theorem 6. $\mathbf{F}^{-\varepsilon} \subseteq \mathbf{OF}^{-\varepsilon}$ *and* $\mathbf{F} \subseteq \mathbf{OF}$

In terms of left forbidding grammars and their power, [4] proves that they are no more powerful than context-free grammars.

Theorem 7. $\mathbf{LF}^{-\varepsilon} = \mathbf{LF} = \mathbf{CF}$

From Theorems 6 and 7 above and from the fact that $\mathbf{CF} \subset \mathbf{F}^{-\varepsilon}$ (see [3]), we obtain the following theorem, which relates the language families generated by left forbidding grammars, one-sided forbidding grammars, and forbidding grammars.

Theorem 8. $\mathbf{LF}^{-\varepsilon} = \mathbf{LF} \subset \mathbf{F}^{-\varepsilon} \subseteq \mathbf{OF}^{-\varepsilon} \subseteq \mathbf{OF}$

Finally, the following two theorems relate the language families generated by propagating one-sided permitting grammars and propagating left permitting grammars to other families of languages.

Theorem 9. $\mathbf{CF} \subset \mathbf{OP}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$

Theorem 10. $\mathbf{CF} \subset \mathbf{LP}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$

Recall that it is not known whether propagating scattered context grammars characterize the family of context-sensitive languages—that is, whether the inclusion $\mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS}$ above is, in fact, an identity (see [10]).

4.2 Normal Forms

Formal language theory has always struggled to turn grammars into *normal forms*, in which grammatical rules satisfy some prescribed properties or format because they are easier to handle from a theoretical as well as practical standpoint. Concerning context-free grammars, there exist two famous normal forms—the Chomsky and Greibach normal forms. In the former, every grammatical rule has on its right-hand side either a terminal or two nonterminals. In the latter, every grammatical rule has on its right-hand side a terminal followed by zero or more nonterminals. Similarly, there exist normal forms for general grammars, such as the Kuroda, Penttonen, and Geffert normal forms. In this section, we present four normal forms for one-sided random context grammars.

In the first normal form, the set of left random context rules coincides with the set of right random context rules.

Theorem 11. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $P'_L = P'_R$. Furthermore, if G is propagating, so is H.*

The second normal form represents a dual normal form to that in Theorem 11. Indeed, every one-sided random context grammar can be turned into an equivalent one-sided random context grammar with the sets of left and right random context rules being disjoint.

Theorem 12. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $P'_L \cap P'_R = \emptyset$. Furthermore, if G is propagating, so is H.*

The third normal form represents an analogy of the well-known Chomsky normal form for context-free grammars. However, since one-sided random context grammars with erasing rules are more powerful than their propagating versions, we allow the presence of erasing rules in the transformed grammar.

Theorem 13. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $(A \to x, U, W) \in P'_L \cup P'_R$ implies that $x \in N'N' \cup T \cup \{\varepsilon\}$. Furthermore, if G is propagating, so is H.*

In the fourth normal form, every rule has its permitting or forbidding context empty.

Theorem 14. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $(A \to x, U, W) \in P'_L \cup P'_R$ implies that $U = \emptyset$ or $W = \emptyset$. Furthermore, if G is propagating, so is H.*

4.3 Reduction

Recall that one-sided random context grammars characterize the family of recursively enumerable languages (see Theorem 1). Of course, it is more than natural to ask whether the family of recursively enumerable languages is characterized by one-sided random context grammars with a limited number of nonterminals or rules. The present section gives an affirmative answer to this question.

The next theorem states that ten nonterminals suffice to generate any recursively enumerable language by a one-sided random context grammar.

Theorem 15. *For every recursively enumerable language K, there exists a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$ and* $\mathrm{card}(N) = 10$.

The number of nonterminals can be also limited in terms of one-sided random context grammars satisfying the normal form from Theorem 11.

Theorem 16. *For every recursively enumerable language K, there exists a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$, $P_L = P_R$, and* $\mathrm{card}(N) = 13$.

To approach the reduction of the number of nonterminals in a finer way, the notion of a *right random context nonterminal* is introduced in the thesis. It is defined as a nonterminal that appears on the left-hand side of a right random context rule. The thesis has demonstrated how to convert any one-sided random context grammar to an equivalent one-sided random context grammar with two right random context nonterminals. This result has been proved also for propagating one-sided random context grammars.

Let us first define the above-mentioned measure formally.

Definition 6. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. If $(A \to x, U, W) \in P_R$, then A is a *right random context nonterminal*. The *number of right random context nonterminals* of G is denoted by $\mathrm{nrrcn}(G)$ and defined as

$$\mathrm{nrrcn}(G) = \mathrm{card}\big(\{A \mid (A \to x, U, W) \in P_R\}\big) \qquad \square$$

The next two theorems state that two right random context nonterminals suffice to keep the power of one-sided random context grammars unchanged.

Theorem 17. *For every recursively enumerable language K, there is a one-sided random context grammar H such that $L(H) = K$ and* $\mathrm{nrrcn}(H) = 2$.

Theorem 18. *For every context-sensitive language J, there is a propagating one-sided random context grammar H such that $L(H) = J$ and* $\mathrm{nrrcn}(H) = 2$.

By analogy with Definition 6, we may define a *left random context nonterminal* and their number in one-sided random context grammars. Then, in the thesis, it is shown that Theorems 17 and 18 can be reformulated in terms of left random context nonterminals and their number. The thesis also proves that we may limit both the total number of right and left random context nonterminals at the same time.

Finally, apart from reducing the overall number of nonterminals and right random context nonterminals, a reduction of the number of right random context rules has been investigated. Recall that a right random context rule is a rule that checks the presence and absence of symbols to the right of the rewritten nonterminal (see Definition 1).

Theorem 19. *For every recursively enumerable language K, there exists a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$ and* $\mathrm{card}(P_R) = 2$.

That is, we know that two right random context rules suffice to keep the generative power of one-sided random context grammars unchanged.

The next theorem says that it is possible to simultaneously reduce both the number of nonterminals and the number of right random context rules.

Theorem 20. *For every recursively enumerable language K, there exists a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$,* $\mathrm{card}(N) = 13$, $\mathrm{nrrcn}(H) = 2$, *and* $\mathrm{card}(P_R) = 2$.

4.4 Other Topics of Investigation

We conclude Section 4 by briefly mentioning other topics related to one-sided random context grammars that have been investigated in the thesis.

Leftmost Derivations

By analogy with the three well-known types of leftmost derivations in regulated grammars (see [3]), three types of leftmost derivation restrictions placed upon one-sided random context grammars have been defined and studied in the thesis. In the *type*-1 *derivation restriction*, during every derivation step, the leftmost occurrence of a nonterminal has to be rewritten. In the *type*-2 *derivation restriction*, during every derivation step, the leftmost occurrence of a nonterminal which can be rewritten has to be rewritten. In the *type*-3 *derivation restriction*, during every derivation step, a rule is chosen, and the leftmost occurrence of its left-hand side is rewritten. In the thesis, the following three results are demonstrated.

(I) One-sided random context grammars with type-1 leftmost derivations characterize the family of context-free languages.

(II) One-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of recursively enumerable languages.

(III) Propagating one-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of context-sensitive languages.

Generalized One-Sided Random Context Grammars

We may generalize the concept of one-sided context from symbols to strings. Obviously, as one-sided random context grammars already characterize the family of recursively languages, such a generalization cannot increase their power. However, a generalization likes this makes sense in terms of variants of one-sided random context grammars. In the thesis, one-sided forbidding grammars that can forbid strings instead of single symbols are studied, and it has been proved that they are computationally complete, even if all strings are formed by at most two symbols.

LL One-Sided Random Context Grammars

In the previous sections, have introduced and studied one-sided random context grammars from a purely theoretical viewpoint. From a more practical viewpoint, however, it is also desirable to make use of them in such grammar-based application-oriented fields as syntax analysis (see [1]). An effort like this obviously gives rise to introducing and investigating their parsing-related variants, such as LL versions.

LL one-sided random context grammars, introduced in the thesis, represent ordinary one-sided random context grammars restricted by analogy with LL requirements placed upon LL context-free grammars. That is, for every positive integer k, (1) LL(k) one-sided random context grammars always rewrite the leftmost nonterminal in the current sentential form during every derivation step, and (2) if there are two or more applicable rules with the same nonterminal on their left-hand sides, then the sets of all terminal strings of length k that can begin a string obtained by a derivation started by using these rules are disjoint. The class of LL grammars is the union of all LL(k) grammars, for every $k \geq 1$.

We have introduced and investigated LL versions of one-sided random context grammars. We have proved that they generate the family of LL context-free languages. Taking a finer look at this generation, we have also demonstrated that the generation of languages by LL one-sided random context grammars can be more succinct than that by LL context-free grammars.

5. Concluding Remarks

This concluding section makes several final remarks concerning the material covered in the thesis with a special focus on its future developments. First, it suggests application perspectives of one-sided random context grammars (Section 5.1). Then, it chronologically summarizes the concepts and results achieved in most significant studies on the subject of the thesis (Section 5.2). Finally, this section lists the most important open problems resulting from the study of the thesis (Section 5.3).

5.1 Application Perspectives

As already stated in Section 1, the thesis is primarily and principally meant as a theoretical treatment of one-sided

random context grammars. Nevertheless, to demonstrate their possible practical importance, we make some general remarks regarding their applications in the present section.

Taking the definition and properties of one-sided random context grammars into account, we see that they are suitable to underly information processing based on the existence or absence of some information parts. Therefore, in what follows, we pay major attention to this application area.

Molecular Genetics
We believe that one-sided random context grammars can formally and elegantly simulate processing information in molecular genetics, including information concerning macromolecules, such as DNA, RNA, and polypeptides. For instance, consider an organism consisting of DNA molecules made by enzymes. It is a common phenomenon that a molecule m made by a specific enzyme can be modified unless molecules made by some other enzymes occur either to the left or to the right of m in the organism. Consider a string w that formalizes this organism so every molecule is represented by a symbol. As obvious, to simulate a change of the symbol a that represents m requires random context occurrences of some symbols that either precede or follow a in w. As obvious, one-sided random context grammars can provide a string-changing formalism that can capture this random context requirement in a very succinct and elegant way. To put it more generally, one-sided random context grammars can simulate the behavior of molecular organisms in a rigorous and uniform way.

Computer Science
Considering the fact that one-sided random context grammars have a greater power than context-free grammars, we may immediately think of applying them in terms of syntax analysis of complicated non-context-free structures during language translation. However, as one-sided random context grammars are computationally complete (see Theorem 1), Rice's theorem (see Section 9.3.3 in [6]) implies that we cannot use them to parse all recursively enumerable languages. Therefore, we should focus on variants of one-sided random context grammars that are not computationally complete, such as propagating one-sided random context grammars.

At the end of Section 4.4, we have mentioned LL versions of one-sided random context grammars, which may be suitable for syntax analysis. Even though they are equally powerful as context-free grammars, they still may be useful since for some languages, they can describe languages in a more economical way. See the thesis for more details.

Linguistics
In terms of linguistics, one-sided random context grammars may be used for generating or verifying that the given texts contain no forbidding passages, such as vulgarisms or classified information. More specifically, generalized one-sided forbidding grammars, which are one-sided forbidding grammars that can forbid the occurrences of strings, are suitable to formally capture such applications.

Another application area of one-sided random context grammars may be syntax-oriented linguistics. Observe that many common English sentences contain expressions and words that mutually depend on each other although they are not adjacent to each other in the sentences. For example, consider the following sentence: *He sometimes goes to bed very late.* The subject (*he*) and the predicator (*goes*) are related. Therefore, we cannot rewrite *goes* to *go* because of the subject. One-sided random context grammars form a suitable formalism to capture and verify such dependencies.

Application-oriented topics like the ones outlined in this section obviously represent a future investigation area concerning one-sided random context grammars.

5.2 Bibliographical and Historical Remarks
This section gives an overview of the crucially important studies published on the subject of the thesis from a historical perspective.

One-sided random context grammars were introduced and investigated in [12]. Their special variants, left permitting and left forbidding grammars, were originally introduced in [2] and [4], respectively. The generative power of one-sided forbidding grammars and their relation to selective substitution grammars were studied in [14]. The nonterminal complexity of one-sided random context grammars was investigated in [13]. A reduction of the number of right random context rules was the topic of [17]. Several normal forms of these grammars were established in [21]. Leftmost derivations were studied in [15]. The generalized version of one-sided forbidding grammars was introduced and investigated in [16]. A list of open problems concerning these grammars appears in [22]. Finally, the LL versions of one-sided random context grammars appear in the thesis for the first time.

5.3 Open Problem Areas
We finish the thesis by summarizing open problems concerning one-sided random context grammars.

(I) What is the generative power of left random context grammars? What is the role of erasing rules in this left variant? That is, are left random context grammars more powerful than propagating left random context grammars?

(II) What is the generative power of one-sided forbidding grammars? We only know that these grammars are equally powerful as selective substitution grammars (see Theorem 3). Thus, by establishing the generative power of one-sided forbidding grammars, we would establish the power of selective substitution grammars, too.

(III) By Theorem 15, ten nonterminals suffice to generate any recursively enumerable language by a one-sided random context grammar. Is this limit optimal? In other words, can Theorem 15 be improved?

(IV) Recall that propagating one-sided random context grammars characterize the family of context-sensitive languages (see Theorem 1). Can we also limit the overall number of nonterminals in terms of this propagating version like in Theorem 15?

(V) What is the generative power of one-sided forbidding grammars and one-sided permitting grammars?

Moreover, what is the power of left permitting grammars? Recall that every propagating scattered context grammar can be turned to an equivalent context-sensitive grammar (see Theorem 3.21 in [10]), but it is a longstanding open problem whether these two kinds of grammars are actually equivalent—the *PSC = CS problem*. If in the future one proves that propagating versions of one-sided permitting grammars and one-sided random context grammars are equivalent, then so are propagating scattered context grammars and context-sensitive grammars (see Theorem 9), so the PSC = CS problem would be solved.

(VI) By Theorem 17, any recursively enumerable language is generated by a one-sided random context grammar having no more than two right random context nonterminals. Does this result hold with one or even zero right random context nonterminals? Notice that by proving that no right random context nonterminals are needed, we would establish the generative power of left random context grammars.

(VII) By Theorem 19, any recursively enumerable language is generated by a one-sided random context grammar having no more than two right random context rules. Does this result hold with one or even zero right random context rules? Again, notice that by proving that no right random context rules are needed, we would establish the generative power of left random context grammars.

Acknowledgements. This work was supported by the following grants: BUT FIT FIT-S-14-2299, European Regional Development Fund in the IT4Innovations Centre of Excellence (MŠMT CZ1.1.00/02.0070), and Visual Computing Competence Center (TE01010415).

References

[1] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, Boston, 2nd edition, 2006.

[2] E. Csuhaj-Varjú, T. Masopust, and G. Vaszil. Cooperating distributed grammar systems with permitting grammars as components. *Romanian Journal of Information Science and Technology*, 12(2):175–189, 2009.

[3] J. Dassow and G. Păun. *Regulated Rewriting in Formal Language Theory*. Springer, New York, 1989.

[4] F. Goldefus, T. Masopust, and A. Meduna. Left-forbidding cooperating distributed grammar systems. *Theoretical Computer Science*, 20(3):1–11, 2010.

[5] S. A. Greibach and J. E. Hopcroft. Scattered context grammars. *Journal of Computer and System Sciences*, 3(3):233–247, 1969.

[6] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, 3rd edition, 2006.

[7] H. C. M. Kleijn. *Selective Substitution Grammars Based on Context-Free Productions*. PhD thesis, Leiden University, Netherlands, 1983.

[8] H. C. M. Kleijn. Basic ideas of selective substitution grammars. In *Trends, Techniques, and Problems in Theoretical Computer Science*, volume 281 of *Lecture Notes in Computer Science*, pages 75–95. Springer, 1987.

[9] C. Martín-Vide, V. Mitrana, and G. Păun, editors. *Formal Languages and Applications*. Springer, Berlin, 2004.

[10] A. Meduna and J. Techet. *Scattered Context Grammars and their Applications*. WIT Press, Southampton, 2010.

[11] A. Meduna and P. Zemek. *Regulated Grammars and Their Transformations*. Faculty of Information Technology, Brno University of Technology, Brno, CZ, 2010.

[12] A. Meduna and P. Zemek. One-sided random context grammars. *Acta Informatica*, 48(3):149–163, 2011.

[13] A. Meduna and P. Zemek. Nonterminal complexity of one-sided random context grammars. *Acta Informatica*, 49(2):55–68, 2012.

[14] A. Meduna and P. Zemek. One-sided forbidding grammars and selective substitution grammars. *International Journal of Computer Mathematics*, 89(5):586–596, 2012.

[15] A. Meduna and P. Zemek. One-sided random context grammars with leftmost derivations. In *LNCS Festschrift Series: Languages Alive*, volume 7300, pages 160–173. Springer Verlag, 2012.

[16] A. Meduna and P. Zemek. Generalized one-sided forbidding grammars. *International Journal of Computer Mathematics*, 90(2):127–182, 2013.

[17] A. Meduna and P. Zemek. One-sided random context grammars with a limited number of right random context rules. *Theoretical Computer Science*, 516(1):127–132, 2014.

[18] A. Meduna and P. Zemek. *Regulated Grammars and Automata*. Springer, New York, 2014.

[19] G. Rozenberg. Selective substitution grammars (towards a framework for rewriting systems). Part 1: Definitions and examples. *Elektronische Informationsverarbeitung und Kybernetik*, 13(9):455–463, 1977.

[20] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages, Volumes 1 through 3*. Springer, New York, 1997.

[21] P. Zemek. Normal forms of one-sided random context grammars. In *Proceedings of the 18th Conference STUDENT EEICT 2012*, volume 3, pages 430–434, Brno, CZ, 2012. Brno University of Technology.

[22] P. Zemek. One-sided random context grammars: Established results and open problems. In *Proceedings of the 19th Conference STUDENT EEICT 2013*, volume 3, pages 222–226, Brno, CZ, 2013. Brno University of Technology.

Selected Publications by the Author

A. Meduna, P. Zemek. *Regulated Grammars and Automata.* Springer, New York, 2014.

A. Meduna, P. Zemek. One-sided random context grammars with a limited number of right random context rules. *Theoretical Computer Science*, 516(1): 127–132, 2014.

A. Meduna, P. Zemek. Left random context ET0L grammars. *Fundamenta Informaticae*, 123(3): 289–304, 2013.

A. Meduna, P. Zemek. Generalized one-sided forbidding grammars. *International Journal of Computer Mathematics*, 90(2): 127–182, 2013.

P. Zemek. One-sided random context grammars: Established results and open problems. In *Proceedings of the 19th Conference STUDENT EEICT 2013, Volume 3*, pages 222–226. Brno University of Technology, Brno, CZ, 2013.

A. Meduna, P. Zemek. One-sided random context grammars with leftmost derivations. In *LNCS Festschrift Series: Languages Alive*, Volume 7300, pages 160–173, Springer, 2012.

A. Meduna, P. Zemek. Nonterminal complexity of one-sided random context grammars. *Acta Informatica*, 49(2): 55–68, 2012.

A. Meduna, P. Zemek. One-sided forbidding grammars and selective substitution grammars. *International Journal of Computer Mathematics*, 89(5): 586–596, 2012.

P. Zemek. Normal forms of one-sided random context grammars. In *Proceedings of the 18th Conference STUDENT EEICT 2012, Volume 3*, pages 430–434. Brno University of Technology, Brno, CZ, 2012.

A. Meduna, P. Zemek. One-sided random context grammars. *Acta Informatica*, 48(3): 149–163, 2011.

A. Meduna, P. Zemek. *Regulated Grammars and Their Transformations*. Faculty of Information Technology, Brno University of Technology, Brno, CZ, 2010.

Instructions to the authors

Publishing procedure

All contributions are web-published. A contribution is published without unnecessary delay right after it has been accepted. Contributions are published on the fly in the current issue. It is at the discretion of the Editor-in-chief to determine, when the current issue is closed and a subsequent new one is open. There will be at least two issues in a year but it is left up to the Editor-in-chief to adjust periodicity of the Bulletin to actual needs.

Extended abstracts of theses is the primary type of article in the Bulletin. Each extended abstract will by annotated by identifying the thesis supervisor, who must recommend it for publication and stands for the Editorial Board in a role similar to a reviewer. We offer publishing extended abstracts on the Bulletin's web before the thesis is defended. This preliminary publishing is a specific service to the academic community. As soon as we learn about successful defence, the extended abstract gains the status of accepted paper and will be included in the forthcoming issue. The accepted paper will be annotated with the date of successful defence and name of the insitution where the defence took place.

It is the policy of the Bulletin to offer a free access to all its articles on the web. Moreover, the publisher will seek opportunities to promote as wide as possible access and/or indexing of the articles. All the past issues remain accessible on the web as part of the web portal of the Bulletin. Closed issues will be made available also in a printable form, free for downloading and printing by anyone interested.

Policy on Originality

It is the policy of the Bulletin that Slovak University of Technology be the sole, original publisher of articles. Manuscripts that have been submitted simultaneously to other magazines, journals or to conferences, symposia, or workshops without the prior written consent of the Editor-in-Chief will be rejected outright and will not be reconsidered. Publication of expanded versions of papers that have been disseminated via proceedings or newsletters is permitted only if the Editor-in-Chief judges that there is significant additional benefit to be gained from journal publication. A conference chairperson can arrange with the Editor-in-Chief to publish selected papers from conferences, symposia, and workshops, after suitable reviewing. The papers must meet the editorial requirements for research articles. Acknowledgement of the originating conference will appear as a credit when the paper is published in the Bulletin.

Manuscript information for extended abstracts of doctoral dissertations

All contributions are submitted electronically. Send your manuscript as LaTeXsources and .pdf files by e-mail to `editor.acm@fiit.stuba.sk`. Paper's length should be 6-12 pages. Please, use LaTeXstyle, which is available to download at bulletin web-page `http://slovakia.acm.org/bulletin/`.

Some remarks to the provided style:

- **Headings and Abstract**
  The heading must contain the title, full name, and address of the author(s), thesis supervisor, abstract of about 100-200 words.
- **Categories and Subject Descriptors**
  Define category and subject descriptors according to ACM Computing Classification System (see `http://www.acm.org/about/class/1998/`).
- **Keywords**
  Please specify 5 to 10 keywords.
- **Equations**
  You may want to display math equations in three distinct styles: inline, numbered or non-numbered display (we recommend the numbered style). Please make sure that your equations are clearly formulated and described.
- **Figures and tables**
  Figures and tables cannot be split across pages, the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures/table, use the environment figure/table to enclose the figure and its caption.
- **References**
  Please use BibTeX to automatically produce the bibliography. If possible use abbreviations like: Proceedings – Proc., International – Int., Conference – Conf., Journal – J.
- **Selected papers by the author**
  This section is used for thesis extended abstracts. Please write down all publications which are related to your thesis.

Published by Slovak University of Technology Press,
Vazovova 5, 812 43 Bratislava, IČO: 00397687
on behalf of the ACM Slovakia Chapter
ISSN 1338-1237 (printed edition)
ISSN 1338-6654 (online)
Registration number: MK SR EV 3929/09