Link Building

Martin Olsen

PhD Dissertation
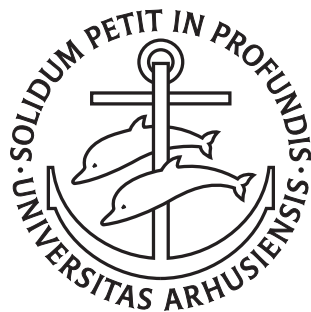
Link Building

A Dissertation
Presented to the Faculty of Science
of Aarhus University
in Partial Fulfilment of the Requirements for the
PhD Degree

by
Martin Olsen
December 16, 2009

Preface

This PhD dissertation is based on seven papers in total among which four are published as single author papers. One of these papers [72] is an extended journal version of one of the other papers [69]. Two papers are submitted – a single author paper and a paper with co-authors Josep Freixas, Xavier Molinero and Maria Serna from the Polytechnic University of Catalonia, Barcelona – and one paper with co-author Tasos Viglas, University of Sydney, is under preparation. The following chronologically ordered list shows where the content of the papers appear in the dissertation:

[70] M. Olsen
    Communities in Large Networks: Identification and Ranking
    Proc. Fourth Workshop on Algorithms and Models for the Web-Graph,
    WAW 2006
    Section 1.4 and Chapter 5

[69] M. Olsen
    Nash Stability in Additively Separable Hedonic Games Is NP-Hard
    Proc. The 3rd conference on Computability in Europe, CiE 2007
    Section 1.5 and Chapter 6

[71] M. Olsen
    The Computational Complexity of Link Building
    Proc. Computing and Combinatorics, 14th Annual International Conference,
    COCOON 2008
    Section 1.3, Section 2.1, Section 2.4.1, Section 3.1 and Section 4.1

[72] M. Olsen
    Nash Stability in Additively Separable Hedonic Games and Community
    Structures
    Theory of Computing Systems, 2009 (Extended version of [69])
    Section 1.5 and Chapter 6

[41] J. Freixas, X. Molinero, M. Olsen and M. Serna
    On the Complexity of Problems on Simple Games
    submitted
    Section 1.6 and Chapter 7

[68] M. Olsen
Maximizing PageRank with new Backlinks
submitted
Section 1.3, Section 2.1, Section 2.3, Section 3.2 and Section 4.2.1

[73] M. Olsen and T. Viglas
MILP for Link Building (working title)
in preparation
Section 1.3, Section 2.1, Section 4.2.2, Section 4.2.3 and Section 4.3

There are parts of the dissertation that are not listed above and there are parts listed more than once.

Abstract

Google uses the PageRank algorithm to compute an estimate of the popularity of each page based solely on the link structure of the web graph – these estimates are the so-called *PageRank values*. A page will achieve one of the top spots of a search result if it has a high PageRank value and matches the search criteria for the actual Google search. For a given page t and $k \in \mathbb{Z}^+$ we study the problem of computing k new links pointing to t – so-called backlinks to t – producing the maximum increase in the PageRank value of t. The problem of obtaining optimal new backlinks in order to achieve good search engine rankings is known as *Link Building* and this problem attracts much attention from the *Search Engine Optimization* (SEO) industry. In this dissertation we concentrate on the problem of *identifying* optimal new backlinks and refer to this problem as *the Link Building problem*. We show that no FPTAS exists for Link Building under the assumption NP$\neq$P and that Link Building is W[1]-hard. On the more positive side we show how to solve the case with fixed $k = 1$ using time corresponding to a small and constant number of PageRank computations using a randomized scheme and we show that Link Building is a member of the complexity class APX. We also show how to use Mixed Integer Linear Programming to solve the problem for smaller graphs and values of k.

We show how the Link Building problem is related to the problem of detecting community structures in networks. We present a community definition justified by a formal analysis of a very simple model of the evolution of a directed graph $G(V, E)$ and show that the problem of deciding whether a community $C \neq V$ exists such that $R \subseteq C$ for a given set of representatives R is NP complete. In spite of the intractability result we show that a fast and simple parameter free greedy approach performs well when detecting communities in a crawl of the Danish part of the web graph.

We present results from a branch of game theory dealing with so-called *Hedonic Games* and argue that community structures can be viewed as *Nash equilibriums* for Hedonic Games and in this way we provide a link to the other topics in the dissertation. To be more specific we show that computing Nash equilibriums in *Additively Separable Hedonic Games* is NP-hard. Finally, we present results from another branch of game theory concerning what is known as *Simple Games*. For several properties we study the computational complexity of deciding whether or not a given simple game has the property. Some of the proof techniques used in this final part of the dissertation are used several other places in the dissertation.

Acknowledgements

Contents

Chapter 1

Introduction

The founders of Google introduced the PageRank algorithm [12, 76] that computes an estimate of the popularity of each page based solely on the link structure of the web graph – these estimates are the so-called *PageRank values*. A page will achieve one of the top spots of a search result if it has a high Page-Rank value and matches the search criteria for the actual Google search. The PageRank algorithm – or variants of the algorithm – can be used to assign a measure of popularity to the nodes in any directed graph. As an example it can also be used to rank scientific journals and publications [11, 22] based on citation graphs.

For a company, it is extremely important that its web page appears at the top – or close to the top – of results when potential customers do a Google search. The problem of obtaining optimal new backlinks[1] in order to achieve good search engine rankings is known as *Link Building* and this problem attracts much attention from the *Search Engine Optimization* (SEO) industry. The main focus of this dissertation is the Link Building problem but we also present results on detection of community structures in networks and argue how this field is related to Link Building. Moreover, we present results from a branch of game theory concerning so-called *Hedonic Games* and establish a connection from these results to community structures. Finally, we present results from another branch of game theory – *Simple Games* – where the link to the other material in the dissertation is common proof techniques.

This introductory chapter is organized as follows: In the next section, we will describe the real world context of the Link Building problem. The main objective of the dissertation is presented in Section 1.2. Sections 1.3 to 1.6 describe the related work and the headlines of the contribution of this dissertation. Section 1.7 gives an outline of the subsequent chapters presenting the details of our contributions.

1.1   Search Engine Optimization (SEO)

The objective of *Search Engine Optimization* – abbreviated SEO – is to improve the search engine visibility for a given web page or set of web pages. To put it more simply: The objective is to make the page(s) appear among the first search results when users query the search engines Yahoo, Google, etc. It is actually possible to pay the search engines to make a link to your web page appear on the page of search results for a given word in the query issued by the user – this is referred to as *paid placement*. As an example, the page of search results for a Google search contains so-called *sponsored links* where Google is paid a fee each time a user clicks on the link – referred to as *Pay Per Click* (PPC). The problem of settling the price of a sponsored link is a research topic by its own. The page of search results also contains non sponsored links that Google considers to be valuable links for the user – these links are referred to as the *organic results*. SEO is about achieving a top spot among the organic results whereas designing a good strategy for paid placement is the objective of

---

[1]A backlink to a page t is a link pointing to t from another page. To be more precise, a backlink is an element in $V \times \{t\}$ where $G(V, E)$ is the directed graph under consideration.
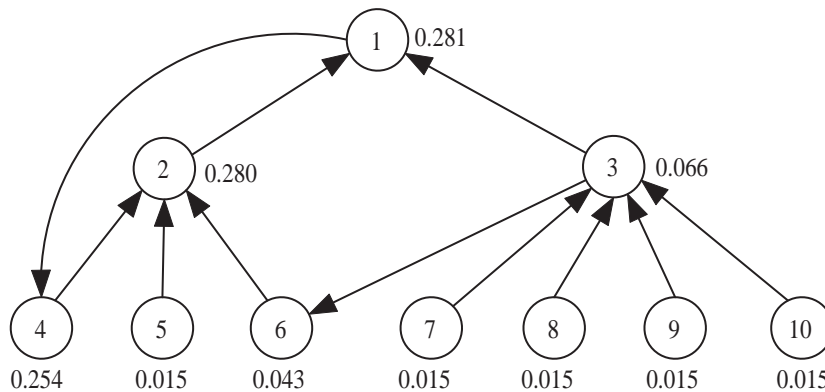
Figure 1.1: A directed graph with PageRank values to the right of or below the nodes.

Search Engine Marketing (SEM).

Companies are predicted to spend almost 9 billion dollars on SEO in 2012 [74] but many non professionals are also doing SEO in an attempt to improve the search engine visibility of their web pages. A search on amazon.com on "search engine optimization" reveals a lot of SEO books (Examples [10, 54, 59, 81]) and there is – not surprisingly – also a lot of online material on the subject (some of the major sites on the topic are www.seomoz.org, www.webmasterworld.com and searchengineland.com). It seems that most of the books and the online material offer practical SEO advice for a target audience consisting of webmasters and SEO consultants assisting web masters. There are even several well attended conferences for SEO professionals (Examples: searchmarketing-expo.com, www.searchenginestrategies.com, and www.pubcon.com). Google had a share of roughly 70% of all US search engine queries in April 2009 according to www.hitwise.com. Google is by far the most popular search engine so we will focus on Google – and especially the PageRank algorithm introduced by the founders of Google – in this dissertation.

1.1.1 Ranking Factors

When Google ranks the web pages for a given query, Google considers what is known as *on page* and *off page* factors. The on page factors for a page are directly controlled by the owners of the page – basically the content of the page – whereas the off page factors are not controlled or only indirectly controlled by the owners of the page. An argument for emphasizing off page factors is that the lack of control of these factors for the owners of a page makes it harder to manipulate or "spam" the rankings. Google is pretty secretive regarding the semantics of the ranking mechanism used so re-engineering of the mechanism is a research topic [9, 37] and it is also a hot online topic.

The common assumption is that the PageRank algorithm [12, 76] introduced by the founders of Google plays a major role in the ranking mechanism (also backed up by the re-engineering results in [9, 37]). The PageRank algorithm

considers only the link structure of the web graph and assigns a score to each web page estimating the *reputation* of the page. The fundamental principle behind the PageRank computation is that the reputation score of a page is divided evenly among the outbound links of the page and distributed to the targets of the outbound links meaning that the score of a page is the sum of the scores "flowing" along the backlinks of the page. In other words, a page is reputable if reputable pages link to it. As an example, a page with 1000 backlinks can be less reputable than a page with only one reputable page linking to or "voting" for it if the page voting for it receives a lot of votes from other pages. The scores – or PageRank values – are thus recursively defined but can easily and efficiently be computed with a simple iterative scheme even for billions of pages with a minor adjustment to the "flow model": In the adjusted model, a fixed fraction $\alpha < 1$ of the reputation score is divided evenly among the outlinks while a fraction of $1 - \alpha$ of the reputation score is divided evenly among *all* pages. Figure 1.1 shows a small example with $\alpha = 0.85$ and the PageRank values normalized so that the sum of the values are 1. Now let us as an example see how the PageRank value of page 1 is related to the other PageRank values where we let π_u denote the PageRank value of page u:

$$\pi_1 = 0.85\pi_2 + \frac{0.85\pi_3}{2} + \frac{(1 - 0.85) \cdot 1}{10} = 0.281$$

It is possible to get a rough impression of the PageRank value of a page by installing the Google toolbar in a browser. The toolbar displays a PageRank number as an integer in the interval 0-10 when the page is visited – the exact relation between the "real" PageRank value (a member of $\mathbb{R}$) and the PageRank value shown in the toolbar is kept as a secret by Google. We will present a more formal treatment of the PageRank algorithm in Section 2.1 including an introduction to the iterative scheme for computing the PageRank values.

A page will do well in the ranking if it is *relevant* considering the actual query and reputable as illustrated by the following simplified identity:

$$\text{Relevance score} + \text{Reputation score} = \text{Ranking score}$$

The details of the Google ranking mechanism is also kept as a secret but the general assumption is that obtaining a high PageRank value is very important for a page that wants to do well in a Google ranking. Obtaining backlinks from reputable pages can have a dramatic effect on the PageRank value of a page but it may come as a surprise that adjusting the structure of the outbound links of a page can increase the PageRank value of the page with roughly a factor 3.6 in the optimal case[2] [4] – so PageRank is actually partly an on page ranking factor. Identifying optimal new backlinks – The Link Building Problem – is the key problem for this dissertation.

1.1.2 Non-Academic Advice for Link Building

We will now briefly cover what seems to be the most dominant general advice on Link Building presented in the literature and online material targeted at a

---

[2]The precise factor is $\frac{1}{1-\alpha^2}$ which is 3.6 for for the typical value $\alpha = 0.85$.

general audience. As mentioned above, the amount of SEO literature and SEO online material is huge so the coverage is based on only a small fraction of the material available: Books: [10, 54, 59, 81], online articles: [36, 50].

Identifying Good Backlinks

According to the literature and online material, the two dominant characteristics for a link (u, t) with good ranking potential for t are the following:

1. u is reputable

2. u and t are related

Some intuition backing up these advice could be the following: Being recommended by a highly ranked computer scientist is the best thing that can happen for a computer scientist aiming for a top spot in the computer science ranking. Getting a recommendation from an expert on a completely other field is probably less valuable. You can also look at it in another way: Some of the presumably many visitors of u will probably use the link (u, t) to visit t which would not be the case if u and t were not related. These are good arguments that (u, t) is a link with good ranking potential for t assuming a well-functioning search engine. It is worth noting that (u, t) would be a good backlink for t *even in a world without search engines* so the objective of Link Building is not solely to obtain good search engine rankings but also to establish links to t on pages visited by many people that might be interested in visiting t.

Three more down to earth arguments offered in the literature and online are the following looking at Google:

1. The PageRank value of u is relatively high and some of the PageRank value will now "flow" to t resulting in a hopefully significant increase in the PageRank value of t.

2. The link (u, t) will confirm that t is a page dealing with the common theme for u and t increasing the relevance score for t on the common theme. Google will probably have more confidence in this confirmation compared to information gained from on page factors.

3. There is maybe a risk that Google ignores or assigns a smaller weight to the link (u, t) compared to the other links on u if u and t are not related.

So how do you identify the links with the characteristics presented above? A typical advice is to query the search engines using queries on the topic for t. The pages in the top of the search results are the u's to go for. More sophisticated techniques use information on the web graph topology: You could go for obtaining links from highly ranked u's linking to your competitors [36] or use commercial link analysis software [50]. As an example, the tool *LinkScape* offers users the ability to "Judge the quality of potential links" to their sites according to the LinkScape homepage[3].

---

[3]www.seomoz.org/linkscape

How to Obtain Backlinks

The Link Building process consists of two steps: 1) Identify optimal backlinks and 2) Obtain the backlinks identified. We will focus on the first step of the process in this dissertation. The second step receives a lot of attention in the literature and online which suggests that it is – at least in many cases – actually possible to obtain given backlinks. Three backlink acquisition approaches for obtaining (u, t) described online and in the literature [10] are the following:

- Link Exchange. Offer the owners of u that you will establish a link to u in exchange (not necessarily with origin t). Maybe you can add some content to t that is relevant and interesting for the visitors of u?

- Embedded Links. Create some good content (Applet, plain HTML, ...) containing the link (u, t) and offer it for free to u.

- Buying Links. Maybe you can simply buy (u, t) from the owners of u. There are even online services for buying/selling links with www.textlinkbrokers.com as an example. It should be noted that Google attempts to take counter measures to paid links as can be seen on the blog[4] of Matt Cutts. Matt Cutts is the head of Google's Web spam team.

1.1.3   White Hat and Black Hat SEO

One obvious way of attempting to "spam" the search engines is to build *Link Farms* that are networks of artificial pages linking to real pages. Link farms are created with the only purpose to improve the rankings of the real pages. In this way, it is possible to obtain a lot of artificial backlinks but the search engines try hard to detect and ignore the link farms. Detection of link farms and spam pages is a computer science research topic [44, 85]. Building link farms is one of the techniques labeled as *Black Hat SEO* [61] as opposed to *White Hat SEO* encompassing "ethical" SEO techniques accepted by the search engines. This dissertation is focusing on the identification of optimal backlinks (u, t) where u is a *real* page. This problem is the equivalent in cyber space to the real world problems of identifying optimal media for commercials for a company or locating optimal spots for physical signs. Whether the Link Building problem is related to white hat SEO or black hat SEO is left to the judgment of the reader of this dissertation.

1.2   The Main Objective of the Dissertation

As we have seen up till now, the Link Building problem is seen by many people as an important problem and there is even commercial link analysis tools available. The purpose of this dissertation is to investigate the Link Building problem and related problems from a computer science perspective. As an example, we will analyze the computational complexity of the problem. As can be seen from Section 1.1.2 locating related pages – or *communities* of pages –

---

[4]www.mattcutts.com/blog/text-links-and-pagerank

on a specific topic is a problem related to Link Building so this problem will also be considered in the dissertation. The work on communities led the author of the dissertation to problems concerning so-called *Hedonic Games* and some of the proof techniques showed also to be applicable for so-called *Simple Games* – this dissertation also contains results from these branches of game theory.

In the four next sections, we will present related work and the headlines for the contribution for each of the four main topics for this dissertation: Link Building, Communities in Networks, Hedonic Games and Simple Games. We will also explain in more detail how the topics are related. The details of the contributions are covered in the subsequent chapters.

1.3 Link Building

Given any directed graph $G(V, E)$ we can compute a PageRank value π_v for every $v \in V$. The details of the computation of π_v are presented in Chapter 2. In this dissertation, we will primarily look at the PageRank values obtained after adding a set of links E' to $G(V, E)$. We will let $\tilde{\pi}_v(E')$ denote the Page-Rank value of v in $G(V, E \cup E')$. The argument E' may be omitted if E' is clear from the context. We will now formally define the Link Building problem where we assume that G is weighted but we will also consider the unweighted case in this dissertation.

Definition 1.1 *The* LINK BUILDING *problem:*

- *Instance: A triple (G, t, k) where $G(V, E)$ is a weighted directed graph with positive integer weights on the edges, $t \in V$ and $k \in \mathbb{Z}^+$.*

- *Solution: A set $S \subseteq V \setminus \{t\}$ with $|S| = k$ maximizing $\tilde{\pi}_t(S \times \{t\})$.*

The theoretical results in this dissertation are based on the original formulation of the PageRank algorithm [12, 76] but the PageRank semantics used by Google has changed according to Matt Cutts [26]. Matt Cutts is not specific in [26] but the link analysis used by Google might have been adjusted in order to take counter measures against link spamming/link farms [44]. As mentioned in Section 1.1.1, some fixed fraction of the PageRank score is distributed uniformly on all pages following the classic formulation of PageRank and this distribution might also have been changed in an attempt to *personalize* the PageRank computation and make it *topic sensitive* [48]. Matt Cutts recently used what he refers to as the "classic PageRank" to explain the link analysis used by Google which justifies using this model even though it is not a "perfect analogy", again using the words of Matt Cutts [26].

In this dissertation we will typically try to maximize the PageRank *value* of a node but we will also briefly consider the problem of achieving the maximum improvement in the *ranking* of the node in which case we also have to take the values of the competitors of the node into consideration.

1.3.1 Related Work

We will now present work directly related to the Link Building problem. Langville and Meyer [58] deal with the problem of updating PageRank efficiently without starting from scratch. Avrachenkov and Litvak [4] study the effect on Page-Rank if a given page establishes one or more links *to* other pages. Avrachenkov and Litvak show that an optimal linking strategy for a page is to establish links only to pages in the *community* of the page. When Avrachenkov and Litvak speak about a web community they mean "... a set of Web pages that a surfer can reach from one to another in a relatively small number of steps". It should be stressed that Avrachenkov and Litvak look for optimal links in $\{t\} \times V$ for a given page t where V is the nodes in the directed graph under consideration and that they conclude that t "... cannot significantly manipulate its PageRank value by changing its outgoing links". Kerchove *et al.* [28] study the more general problem of maximizing *the sum* of PageRank values for a set of pages T by adding links from $T \times V$. In this dissertation, we will mainly look for optimal links in $V \times \{t\}$ which could cause a significant increase in the PageRank value of t.

1.3.2 Contribution

We now summarize the contributions of the dissertation with respect to the Link Building problem. We list references to chapters/sections and papers covering the details in parentheses.

- We develop Theorem 2.1 expressing among other things how the topology of the graph determines the PageRank potential for a set of new backlinks to t (Section 2.3, [68]).

- Lower Bounds (Chapter 3)

  - We consider the variant of the Link Building problem where the objective is to maximize the minimum PageRank value for a given set of nodes $T \subseteq V$ by adding k new links from $V \times V$. This problem is shown to be NP-hard. The max–min formulation is admittedly a bit artificial but the first results on intractability were obtained using this model of the problem so we include these results in the dissertation (Section 3.1, [71]).

  - Compared to the max–min formulation we present stronger intractability results for the more realistic formulation of the Link Building problem from Definition 1.1. Based on Theorem 2.1 on the topology influence we show that no FPTAS exists for this problem under the assumption NP$\neq$P and we also show that this problem is W[1]-hard. We also consider the computational complexity of the variant of Link Building where we are allowed to add or remove links with source t besides adding k new backlinks to t and the variant where we for each page p have a cost $c(p) \in \mathbb{Z}^+ \cup \{+\infty\}$ for obtaining the link (p, t) and where the objective is to maximize the PageRank value of t for

a given budget $B \in \mathbb{Z}^+$ – the cost models the price or the difficulty of obtaining (p, t) as discussed in Section 1.1.2 (Section 3.2, [68]).

- Upper Bounds (Chapter 4)

  – We look at the simplest case of the problem where we want to find *one* new optimal backlink for a given node t – in other words, $k = 1$ is *fixed* in Definition 1.1. We present a simple randomized algorithm solving this case with a time complexity corresponding to a *small* and *constant* number of PageRank computations as opposed to the brute force approach using $|V|$ PageRank computations computing π_t in $G(V, E \cup \{(u, t)\})$ for every $u \in V$. Results of experiments with the algorithm on artificial computer generated graphs and a crawl of the Danish part of the web graph are also reported (Section 4.1, [71]).

  – We use Theorem 2.1 on the topology influence to characterize sets of backlinks with a high PageRank potential for t (Section 4.2.1, [68]).

  – We analyze the naive Link Building approach where the solution is the k u-nodes with the maximum values of π_t in $G(V, E \cup \{(u, t)\})$ – the graph obtained after adding the link (u, t). Let $\tilde{\pi}_t^N$ denote the PageRank value of t obtained by the naive approach and let $\tilde{\pi}_t^*$ denote the optimal value. Based on Theorem 2.1 we systematically construct a graph with $\tilde{\pi}_t^* \approx 13.8 \tilde{\pi}_t^N$ proving that the naive approach is indeed naive (Section 4.2.2, [73]).

  – We prove that the unweighted case of LINK BUILDING is a member of the complexity class APX by presenting a greedy polynomial time algorithm guaranteeing $\tilde{\pi}_t^* \leq \frac{1}{1-\alpha^2} \frac{e}{e-1} \tilde{\pi}_t^G$ where $\tilde{\pi}_t^G$ denotes the Page-Rank value of t obtained by the greedy algorithm. The worst case factor on the right hand side is roughly 5.7 for $\alpha = 0.85$ which is considerably smaller than the factor obtained by the naive approach for a specific graph (Section 4.2.3, [73])

  – We show how to attack the Link Building problem by using Mixed Integer Linear Programming (MILP). We present an integer linear program solving the Link Building problem as defined by Definition 1.1 and we show how to construct an integer linear program for solving the problem of "beating" specific nodes in the *ranking* induced by the PageRank values. We also show how to construct an integer linear program for the problem of achieving the highest improvement in the ranking for a given budget (Section 4.3, [73]).

1.4 Communities in Networks

We now turn to the field of identification of members of communities in networks. A community in a graph $G(V, E)$ is a set of somewhat isolated nodes linking heavily to each other – for example a set of pages in the web graph related to a particular topic. The purpose of the techniques presented in this

dissertation is not to partition the network into several communities. The purpose is to isolate and rank the members of a *single* community C given a set $R \subseteq C$ of representatives.

1.4.1   Relation to the other Topics

As we saw in Section 1.1.2, the objective of a Link Building campaign might be to obtain backlinks to t from highly ranked pages related to t – in other words, to obtain links from highly ranked pages in the *community* given by the representative t. Please note that there might be several communities containing t – as an example, the author of this dissertation is a computer scientist but he is also a member of the local soccer club – so we would typically use several hand-picked representatives to "define" the community we are going for. In Section 5.3, we report results on experiments where we have successfully identified and ranked Danish computer science sites and chess pages using only a few representatives. It should be noted that information on the *content* of the pages is only used in the process of hand-picking the representatives.

Another possible use for the community detection techniques is to use the techniques in a preprocessing step for the MILP approach for Link Building as explained in more detail in Section 4.3.

1.4.2   Related Work

Before the discussion of related work on communities we would like to introduce some notation used in this dissertation. We define the *relative attention* that u shows v as $w_{uv} = \frac{m(u,v)}{outdeg(u)}$ where $m(u,v)$ is the multiplicity of link (u,v) in E. If $outdeg(u) = 0$ then $w_{uv} = 0$. For $C \subseteq V$ we let $w_{uC} = \sum_{c \in C} w_{uc}$, i.e. the attention that u shows the set of nodes C.

The detection of community structure in networks has been subject to a great deal of research [60,67]. Newman and Girvan [67] present a class of *divisive* algorithms for detecting community structures in networks. An algorithm in this class iteratively removes the edge with the highest score of some *betweenness* measure. The betweenness measure is recalculated after each edge removal. One way of measuring the betweenness is to count the number of shortest paths that runs through an edge. A so-called *modularity measure* is used to calculate the quality of the current partition each time a new group of nodes is isolated by the edge removal procedure.

Bagrow *et al.* [6] present a "local" method for detecting the community given by a single representative. A breadth first search from the representative stops when the number of edges connecting the visited nodes with un-visited nodes drops in a special way and reports the visited nodes as a community. Bagrow *et al.* repeat this procedure for each node and analyzes the overlap of the communities in order to eliminate problems with what the authors call "spill-over" of the breadth first search.

Formal definitions of communities are provided by Flake and different co-authors in [38] and [39]. According to [38], a community in an *undirected* graph with edges of unit capacity is a set of nodes C such that for all $v \in C$, v has at

least as many edges connecting to nodes in C as it does to nodes in $\bar{C} = V - C$. Using the notion of relative attention extended to undirected graphs, this is $\forall v \in C : w_{vC} \geq \frac{1}{2}$. Flake *et al.* show in [38] how to identify a community containing a set of representatives as an *s-t* minimum cut in a graph with a virtual source s and virtual sink t. They show how the method can process only the neighborhood of the representatives yielding a local method with time complexity dependent on the size of the neighborhood. It is not possible for a node within a distance of more than two from the representatives to join the community for this "local" variant of their method.

The web graph is treated as a weighted *undirected* graph in [39] with an edge between page i and page j if and only if there is a link from page i to j or vice versa. Edge $\{i, j\}$ has weight $w_{ij} + w_{ji}$ following our definitions of attention. The graph is expanded with a virtual node t connected to all nodes with edges with the same weight α and the *community* of page s is defined by means of an *s-t* minimum cut. The members of such a community can be identified with a maximum flow algorithm.

The definitions in [38] and [39] are not based on a model of the evolution of a graph. It should also be noted that it seems impossible for a universally popular member to be a member of a small community by the definitions in [38] and [39]. A relatively high in-degree of a member will prevent it from being on the community side of a minimum cut. In fact, any member v of a relatively small community in a relatively large network is risking being forced to leave the community if v attracts some attention from non community members if the community definition is based on minimum cuts and the graph is undirected.

Andersen *et al.* [1] and Andersen and Lang [3] have presented some very interesting approaches to identifying communities containing specific nodes. In both papers, random walks are used to identify the communities. The graphs are assumed to be *unweighted* and *undirected* where this dissertation deals with *directed* graphs. The results in [1] have recently been generalized to directed graphs by Andersen *et al.* [2].

1.4.3 Contribution

The results related to detection and ranking of members of communities were published by the author of this dissertation in [70]. The details can also be found in Chapter 5 of this dissertation. The contribution on this topic can be summarized as follows:

- We present a community definition justified by a formal analysis of a very simple model of the evolution of a directed graph.

- The problem of deciding whether a community $C \neq V$ exists such that $R \subseteq C$ for a given set of representatives R is shown to be NP complete.

- In spite of the intractability result, we show that a fast and simple parameter free greedy approach performs well when detecting communities in the Danish part of the web graph. The time complexity of the approach is only dependent on the size of the found community and its immediate

surroundings. Our method is "local" as the method in [6] but it does not use breadth first searches. We also show how to use a computationally inexpensive local variant of PageRank to rank the members of the communities and compare the ranking with the PageRank for the total graph.

1.5   Hedonic Games

We now turn our attention to *Hedonic Games*. The introduction to this branch of game theory will be a little more formal compared to the preceding sections in an attempt to clarify the concepts and contribution related to this topic.

In a *Coalition Formation Game*, a set of players splits up in coalitions so that each player belongs to exactly one coalition. Each player prefers certain partitions[5] of the players to other partitions. If all players are satisfied with the partition in some formalized sense - or not able to move - the partition is said to be *stable*. A stable partition is called an *equilibrium*. For an overview of the field of *Coalition Formation Games*, we refer to the report [45] by Hajdukova.

A given notion of stability can have limitations in terms of computability. For some types of games it might be impossible to effectively compute equilibriums on a computing device under the assumption NP≠P. If a real world system is modeled using Coalition Formation Games and equilibriums with such limitations you should not expect to be able to calculate the equilibriums using a computer if the model is large. It is also an interesting question whether a real system is able to find an equilibrium if a computer cannot find it effectively. This is the motivation for analyzing the computational complexity for a given notion of stability as also pointed out by Daskalakis and Papadimitriou in [27] and Chen and Rudra in [21]. In this dissertation, we prove limitations for the notion of *Nash stability* in *Additively Separable Hedonic Games*.

The players of a Hedonic Game form coalitions so that each player belongs to exactly one coalition and the players only care about which other players team up with them. In order to define the game, we specify for each player i which coalitions player i prefers to be a member of:

Definition 1.2 *A* Hedonic Game *is a pair* $(N, \preceq)$ *where* $N = \{1, 2, \ldots, n\}$ *is the set of* players *and* $\preceq = (\preceq_1, \preceq_2, \ldots, \preceq_n)$ *is the* preference profile *specifying for each player* $i \in N$ *a reflexive, complete and transitive* preference relation $\preceq_i$ *on the set* $N_i = \{S \subseteq N : i \in S\}$.

In an *additively separable* Hedonic Game, we are given a function $v_i : N \to \mathbb{R}$ for each player $i \in N$ where $v_i(j)$ is the *payoff* of player i for belonging to the same coalition as player j:

Definition 1.3 *A Hedonic Game* $(N, \preceq)$ *is* additively separable *if there exists a* utility function $v_i : N \to \mathbb{R}$ *for each* $i \in N$ *such that*

$$\forall S, T \in N_i : T \preceq_i S \Leftrightarrow \sum_{j \in T} v_i(j) \leq \sum_{j \in S} v_i(j) \ .$$

---

[5]A partition of a set N is a collection of non empty disjoint subsets of N with union N.

Changing the value $v_i(i)$ has no effect on $\preceq_i$ so we assume $v_i(i) = 0$.

1.5.1 Stability Concepts

In this dissertation, we will focus on one type of stability: *Nash stability*. A partition Π of N is *Nash stable* if it is impossible to find a player p and a coalition $T \in \Pi \cup \{\emptyset\}$ such that p strictly prefers $T \cup \{p\}$ to the coalition of p in Π – in which case p would be better off by joining T:

Definition 1.4 *The partition* $\Pi = \{S_1, S_2, \ldots, S_K\}$ *of* N *is Nash stable if and only if*

$$\forall i \in N, \forall S_k \in \Pi \cup \{\emptyset\} : S_k \cup \{i\} \preceq_i S_\Pi(i) \ . \tag{1.1}$$

where $S_\Pi(i)$ *denotes the set in the partition* Π *that* i *belongs to.*

We will briefly mention the three other main stability concepts for Hedonic Games: *individual stability, contractual individual stability* and *core stability*. A partition Π is individually stable if it is impossible to find a player p and a coalition $T \in \Pi \cup \{\emptyset\}$ such that 1) p is better off by joining T and 2) No player in T would be worse off if p joined T. A partition Π is contractually individually stable if we cannot find a player p and a coalition $T \in \Pi \cup \{\emptyset\}$ satisfying 1) and 2) above and the following condition: 3) No player in $S_\Pi(p)$ would be worse off if p left $S_\Pi(p)$. This shows that Nash stability implies individual stability and that individual stability implies contractual individual stability.

The concepts of Nash stability and core stability are on the other hand independent in the sense that none of the concepts imply the other one [45]. A partition Π is core stable if no $X \subseteq N$ exists such that all players in X strictly prefer X to their coalition in Π. We refer to [45] for more details.

1.5.2 Relation to the other Topics

A community structure of a network is a partition of the nodes into communities. In other words, it is a partition of the nodes into groups so that there are many connections between nodes belonging to the same group and few connections between nodes belonging to different groups. We will link community structures to equilibriums so that the limitations proven in this dissertation of the stability concepts formally indicate that computing community structures is hard.

1.5.3 Related Work

Sung and Dimitrov [82] show that the problem of deciding whether a *given* partition is core stable in an Additively Separable Hedonic Game is co-NP complete – the corresponding problem concerning Nash stability is clearly solvable in polynomial time. Cechlarova and Hajdukova [16, 17] study the problem of computing core stable partitions in Hedonic Games where the players compare the best (or worst) members in two coalitions when evaluating the coalitions. Actually, different variants of core stability are considered by Cechlarova and Hajdukova.

Ballester has shown in [7] that the problem of deciding whether a Nash stable partition exists in a Hedonic Game with *arbitrary* preferences is NP-complete. On the other hand, Bogomolnaia and Jackson show in [51] that a Nash stable partition exists in every Additively Separable Hedonic Game with *symmetric* preferences. The preferences are symmetric if $\forall i, j \in N : v_i(j) = v_j(i)$. If v_{ij} is the common value for $v_i(j)$ and $v_j(i)$ in a symmetric game then Bogomolnaia and Jackson show that any partition Π maximizing $f(\Pi) = \sum_{S \in \Pi} \sum_{i,j \in S} v_{ij}$ is Nash stable.

Burani and Zwicker introduce the concept of *descending separable* preferences in [13]. Burani and Zwicker show that descending separable preferences guarantees the existence of a Nash stable partition. They also show that descending separable preferences do not imply and are not implied by additively separable preferences.

As opposed to Newman and Girvan [67], a formal definition of a *community* appears in [38] by Flake *et al.* as also mentioned in Section 1.4.2. Using the terminology from coalition formation games, a *community* is a subset of players $C \subseteq N$ in an additively separable game with symmetric preferences such that $\forall i \in C : \sum_{j \in C} v_{ij} \geq \sum_{j \in N-C} v_{ij}$. In other words, each player in C gets at least half the total possible payoff by belonging to C. Flake show with different co-authors in [39] that the problem of deciding whether it is possible to partition N into k communities is NP-complete. Such a partition is Nash stable but a Nash stable partition is not necessarily a partition into communities. The proof techniques used in this dissertation with respect to hedonic games are similar to those used in [39].

1.5.4   Contribution

The results related to Hedonic Games were published by the author of this dissertation in [69, 72] and the details appear in Chapter 6 of this dissertation – [72] is a journal version of [69]. A significant difference between the two versions is that [72] contains considerations relating community structures and equilibriums of Hedonic Games.

- Compared to Ballester [7], we restrict our attention to Additively Separable Hedonic Games and show that the problem of deciding whether a Nash stable partition exists in such a game is NP-complete.

- We relate the field of detection of community structures to Nash stable partitions in Additively Separable Hedonic Games and argue that community structures in networks can be viewed as Nash stable partitions.

- The link to community structures motivates looking at the computational complexity of computing equilibriums in games with symmetric and positive preferences. We show that the problem of deciding whether a *non-trivial* Nash stable partition exists in an Additively Separable Hedonic Game with *non-negative* and *symmetric* preferences is NP-complete. This result also applies to individually stable partitions since individually stable partitions are Nash stable and vice versa in such games.

1.6 Simple Games

Simple games can be viewed as models of voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo.

Definition 1.5 *A simple game* Γ *is a pair* (N, W) *in which* $N = \{1, \ldots, n\}$ *for some positive integer* n, *and* W *is a collection of subsets of* N *that satisfies* $N \in W$, $\emptyset \notin W$, *and the* monotonicity *property:* $S \in W$ *and* $S \subseteq R \subseteq N$ *implies* $R \in W$.

The members of W are the *winning* subsets/coalitions. The intuition is that a set S is a winning coalition *iff* the bill or amendment passes when the players in S are precisely the ones who vote for it. A simple game can be specified in several ways as illustrated by the following examples:

- An explicit listing of W

- An explicit listing of the *minimal* elements of W

- An explicit listing of the *loosing* subsets $L = 2^N \setminus W$

- An explicit listing of the *maximal* elements of L

- A *quota* $q \in \mathbb{R}^+$ and a *weight function* $w : N \to \mathbb{R}^+$ such that S is winning exactly when the sum of weights of S meets or exceeds q. A *weighted* game is a simple game that can be specified by a quota and a set of weights – it should be noted that the weighted games form a proper subset of the simple games.

1.6.1 Relation to the other Topics

The proof techniques used to prove some of the intractability results for the other topics of the dissertation are also used to prove intractability results for weighted games. This was the reason that the author of the dissertation joined Freixas, Molinero and Serna from Polytechnic University of Catalonia in Barcelona on some work on computational complexity related to simple games.

1.6.2 Related Work and Contribution

There are several properties related to simple games. We have already seen that a simple game can be *weighted*. Another example is that a simple game can be *decisive* if $\forall S \in 2^N : S \in W \Leftrightarrow N \setminus S \in L$ – exactly one of S or $N \setminus S$ is winning for every $S \subseteq N$. The main focus of the work on simple games is to study the computational complexity of deciding whether or not a simple game has a certain property. We obtain results for several properties combined with the different ways of representing a simple game as listed above. The work on simple games is only loosely connected to the Link Building problem so we refer to Chapter 7 for a thorough coverage of the related work and contribution on this topic. Chapter 7 is based on [41].

1.7   Outline

Chapter 2 contains a deeper introduction to the Link Building problem and the PageRank algorithm. Chapters 3 to 7 cover the contribution of the dissertation in details. The lower and upper bounds for the Link Building problem are the subjects of Chapters 3 and 4 respectively. Detection and ranking of community members in networks is the theme of Chapter 5 and the results related to Hedonic Games are presented in Chapter 6. Finally, Chapter 7 is devoted to Simple games.

Chapters 3 and 4 are dependent upon Chapter 2. These are largely the dependencies among the subsequent chapters so the reader of the dissertation can safely skip one or more of them and concentrate on the chapters covering topics which the reader finds interesting.

Chapter 2

Link Building and the PageRank Algorithm

In this chapter, we will briefly present the mathematics behind the Page-Rank algorithm. We will also present a theorem predicting the effect on the PageRank vector of adding a set of new links pointing to the same page to the directed graph under consideration. Finally, we will try to improve the readers understanding of the subtleties of the PageRank algorithm and the Link Building problem through examples.

2.1   Mathematical Background

This section gives the mathematical background for the PageRank algorithm. We refer to [53] for more details on Finite Markov Chains in general and to [57] for more details on the PageRank algorithm. All vectors throughout this dissertation are column vectors.

Let $G(V, E)$ denote a directed graph. We allow multiple occurrences of $(u, v) \in E$ in this dissertation implying a *weighted* version of the PageRank algorithm as described in [11] but we will also present results for the *unweighted* version where multiple links from one node to another count as one. We let $|V| = n$ and $|E| = m$. The nodes V and links E could as an example represent the pages and links in the web graph respectively. A *random surfer* visits the nodes in V according to the following rules: When visiting u, the surfer picks a link $(u, v) \in E$ uniformly at random and visits v. If u is a sink[1] then the next node to visit is chosen uniformly at random from V. The sequence of nodes visited by the random surfer is a Finite Markov Chain with state space V and transition probability matrix $P = \{p_{uv}\}$ given by $p_{uv} = \frac{m(u,v)}{\text{outdeg}(u)}$ where $m(u, v)$ is the multiplicity or weight of link (u, v) in E and outdeg(u) is the out degree of u. If outdeg$(u) = 0$ then $p_{uv} = \frac{1}{n}$.

Now we modify the behavior of the random surfer so that he behaves as described above with probability $\alpha < 1$ when visiting u but *zaps* with probability $1 - \alpha$ to a node v chosen uniformly at random from V. Zapping is *always* done with probability $1 - \alpha$ – even when visiting a sink. The sinks can be thought of as linking to *all* nodes in the graph. Throughout this dissertation, we will assume that α is a fixed constant and that $\alpha = 0.85$, unless otherwise stated, which is the value used in most of the initial experiments performed by the founders of Google [76]. If E is the matrix with all 1's then the transition probability matrix Q for the modified Markov Chain is given by $Q = \frac{1-\alpha}{n}E + \alpha P$. The powers $w^T Q^i$ converge to the same probability distribution π^T for any initial probability distribution w on V as i tends to infinity – implying $\pi^T Q = \pi^T$. In fact, any Markov Chain with a transition probability matrix Q satisfying that Q^N has no zero entries for some N has this convergence property [53]. Our Q matrix has no zero entries due to zapping so in this case, we can use $N = 1$. The vector $\pi = \{\pi_v\}_{v \in V}$ is known as the PageRank vector. Computing $w^T Q^i$ can be done in time $O((n+m)i)$ and according to [57] 50 - 100 iterations provide a useful approximation for π for $\alpha = 0.85$. Two interpretations of π are the following:

---

[1]A sink is a node not linking to any node.

- π_v is the probability that a random surfer visits v after i steps for large i.

- π_v can be seen as a measure of how reputable or popular v is. The identity $\pi^T Q = \pi^T$ shows that the PageRank values "flow" along the links as described in Section 1.1.1 – a node is popular/reputable if it is pointed to by popular/reputable nodes. There is only one probability distribution satisfying $\pi^T Q = \pi^T$ if Q^N has no zero entries for some N [53] so it is not possible to find another probability distribution satisfying the "flow conservation properties". The PageRank vector π^T is referred to as the unique *stationary* probability distribution for Q.

The matrix $I - \alpha P$ is invertible where I is the identity matrix, and entry z_{uv} in $Z = (I - \alpha P)^{-1}$ is the *expected* number of visits – preceding the first zapping event – to node v for a random surfer starting at node u [4, 53]. If $u = v$ then the initial visit is also included in the count. The entries in Z induce a sort of distance measure on the nodes in V: Two nodes u and v that are "close" to each other will have relatively large entries z_{uv} and z_{vu}. The following identity expresses the connection between π and Z [4] where e is the vector with all entries equal to 1 – the identity can be deduced from $\pi^T Q = \pi^T$ by using $\pi^T E = e^T$:

$$\pi^T = \frac{1 - \alpha}{n} e^T Z \ . \tag{2.1}$$

As stated earlier, we will typically look at the PageRank vector for the graph we obtain if we add a set of links E' to $G(V, E)$. We will let $\tilde{\pi}_v(E')$ denote the PageRank value of v in $G(V, E \cup E')$. The argument E' may be omitted if E' is clear from the context.

2.1.1 List of Symbols

We now provide a list of the most important symbols used in this chapter and Chapters 3 and 4. The list also contains brief explanations of the symbols and the list is intended to be used for later reference.

$G(V, E)$**:** The directed graph under consideration with $n = |V|$ and $m = |E|$ where V denotes the set of nodes/vertices and E denotes the directed edges/links.

$P = \{p_{uv}\}_{u,v \in V}$**:** An $n \times n$ matrix with $p_{uv} = \frac{\mathrm{m}(u,v)}{\mathrm{outdeg}(u)}$ where $\mathrm{m}(u, v)$ is the multiplicity of link (u, v) in E and $\mathrm{outdeg}(u)$ is the out degree of u. If $\mathrm{outdeg}(u) = 0$ then $p_{uv} = \frac{1}{n}$. P contains transition probabilities modeling the behavior of a random surfer that is not zapping.

$\alpha \in [0, 1)$**:** A fixed constant known as the "damping factor" for the PageRank computation. A random surfer zaps with probability $1 - \alpha$ and goes to a node in V chosen uniformly at random. Unless otherwise stated, we will assume $\alpha = 0.85$ in this dissertation.

$Z = \{z_{uv}\}_{u,v \in V} = (I - \alpha P)^{-1}$: An $n \times n$ matrix. z_{uv} is the *expected* number of visits to node v before zapping for a random surfer starting at node u. If $u = v$ then the initial visit counts. A sink can be thought of as linking to all other nodes so $z_{uu} > 1$ if u is a sink.

E: An $n \times n$ matrix with all 1'es.

$Q = \frac{1-\alpha}{n} E + \alpha P$: An $n \times n$ matrix with transition probabilities for a random surfer following a link with probability α and zapping with probability $1 - \alpha$.

$\pi = \{\pi_v\}_{v \in V}$: The PageRank vector π is the unique probability distribution satisfying $\pi^T Q = \pi^T$ so π is the stationary probability distribution for the random surfer model. The connection to Z is expressed by the following identity where e is a column vector will all 1'es:

$$\pi^T = \frac{1 - \alpha}{n} e^T Z \ .$$

The PageRank value π_v is the probability for visiting v after i steps and it is also the *expected* fraction of visits to v for large i regardless of the starting node [53]. So regardless of the initial distribution w of the random surfers, we will obtain a distribution close to π after a large number of steps:

$$w^T Q^i \to \pi^T \text{ for } i \to \infty \ . \tag{2.2}$$

If $\alpha = 0.85$ we will obtain a good approximation even after 50-100 steps. Using (2.2) is an efficient way to compute π and it is referred to as *the power method*[2].

$\tilde{\pi}_v(E')$: $\tilde{\pi}_v(E')$ is the PageRank value of v in $G(V, E \cup E')$ – the graph obtained after adding the links E' to $G(V, E)$. The argument E' may be omitted if E' is clear from the context.

r_{uv}: The symbol r_{uv} appears for the first time in Section 4.2.3 and it is defined as the probability for reaching node v before zapping for a random surfer starting at node u. These are some useful identities expressing how π, r_{uv} and z_{uv} are related [4]:

$$z_{uv} = r_{uv} z_{uu} \text{ if } u \neq v \ .$$

$$z_{uu} = \frac{1}{1 - r_{uu}} \ .$$

$$\pi_t = \frac{1 - \alpha}{n} z_{tt} \left(1 + \sum_{u \neq t} r_{ut}\right) \ .$$

Please note that $r_{uv} > 0$ for all v if u is a sink.

---

[2]The power method is a well-known method from mathematics for computing *dominant* eigenvectors and π^T is the unique dominant eigenvector for Q [56].

2.2 The Link Exchange Example

We will try to increase the readers understanding of the PageRank algorithm and its subtleties by means of examples. The examples will appear in this section and Section 2.4. We will start by presenting a link exchange scheme having a dramatic *negative* effect on the PageRank value for one of the participants in the scheme. The scheme is a very simple scheme where two pages agree to establish links to each other. The fact that such a scheme can be harmful for one of the participants may come as a surprise – the SEO literature mentioned in Section 1.1.2 does not deal with such subtleties.

The link exchange scheme is shown in Figure 2.1 where the two nodes 1 and 11 with dashed links have agreed to link to each other. Node 1 is a popular node and the probability for returning to node 1 before zapping for a random surfer visiting node 1 is at its maximum prior to the link exchange. Node 11 is a "low life" node with a relatively big out degree. The dashed link to node 1 will only attract a few "new" random surfers to node 1, but the probability for returning to node 1 before zapping will decrease dramatically when node 1 establishes the new link to node 11. A direct computation shows that $\tilde{\pi}_1 \approx 0.49\pi_1$ – the PageRank value of node 1 after the exchange is roughly half of the PageRank value prior to the exchange. This example shows that modifying the outgoing links on a page can have a negative effect on the PageRank value of the page – in Section 2.4 we will see that a page can also benefit from adjusting the structure of the outgoing links. The lack of memory for the random surfers is in the opinion of the author of this dissertation the reason that some people might find this example counter intuitive.

2.3 The Effect of Receiving Links

The main focus of this dissertation is the problem of computing an optimal set of new links pointing to the same page as formalized in Definition 1.1. Before presenting examples dealing with this problem, we will develop a theorem expressing how the topology of the graph affects the PageRank potential for a new set of backlinks for a page.

Avrachenkov and Litvak [4] study the effect on PageRank of adding new links *with the same origin* to the web graph. Avrachenkov and Litvak establish a theorem that expresses the new PageRank vector $\tilde{\pi}$ by means of the "old" Page-Rank vector π and the "old" version of Z. We present Theorem 2.1 showing the effect of adding new links *pointing to the same page*. Without loss of generality, we assume that each of the pages 2 to $k + 1$ establish a link *to* page 1. The techniques used in the proof are similar to the techniques used in [4].

Theorem 2.1 shows how to express the *increase* (or *decrease*) in the Page-Rank value for the page p as a product of two factors: Roughly, the first factor concerns the PageRank values of the nodes involved and the second factor $c = M^{-1}q$ concerns the "distances" between the nodes involved in the update.

Theorem 2.1 *Let each of the pages 2 to $k + 1$ create a link to page 1. If $\tilde{\pi}_p$*
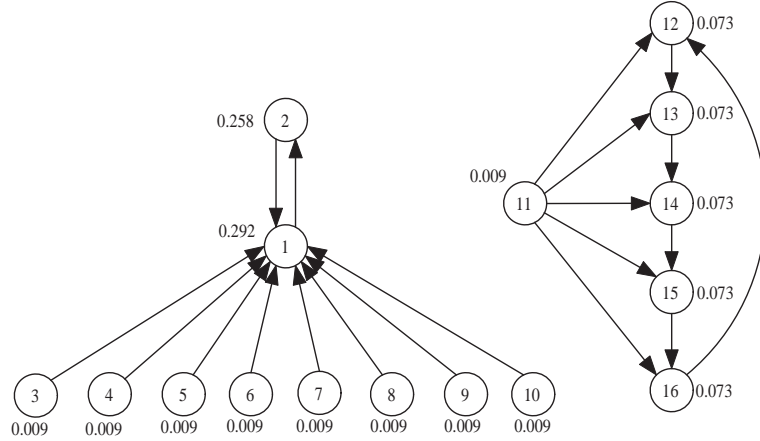
(a) The graph before the link exchange. The graph consists of two components.
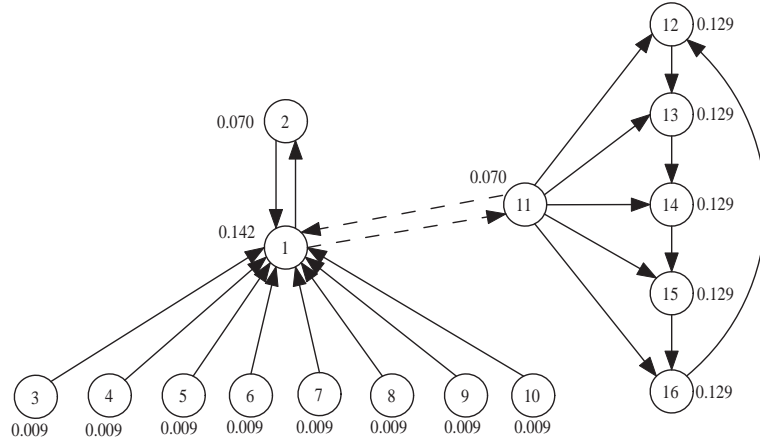
(b) The dashed links indicate a link exchange scheme which is harmful for node 1. Node 1 will obtain a PageRank value which is roughly the half of the original value if the dashed links are added to the graph.

Figure 2.1: A Link Exchange example. The PageRank values are shown beside the nodes.

denotes the updated PageRank value for page p for $p \in \{1, \ldots, n\}$ then we have:

$$\tilde{\pi}_p = \pi_p + \begin{bmatrix} \pi_2 & \pi_3 & \ldots & \pi_{k+1} \end{bmatrix} M^{-1} q \ .$$

where $M = \{m_{ij}\}$ is a $k \times k$ matrix and q is a k-dimensional column vector given by

$$m_{ij} = \delta_{ij} k_{i+1} + z_{i+1j+1} - \alpha z_{1j+1} \ .$$

$$q_i = \alpha z_{1p} - z_{i+1p} + \delta_{i+1p} \ .$$

Here $k_i = outdeg(i)$ prior to the update and $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Proof. Let e_i denote the n-dimensional column vector with a 1 at coordinate i and 0's elsewhere and e denote the n-dimensional column vector with all

1'es. Let b_i denote the k-dimensional column vector with all 0'es except a 1 at coordinate i. Let $\tilde{P}$ denote the updated version of the matrix P. Then we have $\tilde{P} = P + \Delta_P$ where

$$\Delta_P = \sum_{i=2}^{k+1} e_i \frac{1}{k_i + 1}(e_1^T - e_i^T P) \ .$$

The corresponding change of $I - \alpha P$ is

$$(I - \alpha \tilde{P}) - (I - \alpha P) = -\alpha \Delta_P \ .$$

We will use the Woodbury formula [47] to compute $\tilde{Z} = (I - \alpha \tilde{P})^{-1}$ – the updated version of Z. In order to do this, we find matrices S, T and U with dimensions $n \times k$, $k \times k$ and $k \times n$ respectively such that

$$-\alpha \Delta_P = STU \ .$$

We will use

$$S = -\sum_{i=2}^{k+1} e_i b_{i-1}^T \ .$$

$$T = \sum_{i=2}^{k+1} b_{i-1} b_{i-1}^T \frac{1}{k_i + 1} \ .$$

$$U = \sum_{i=2}^{k+1} \alpha b_{i-1}(e_1^T - e_i^T P) \ .$$

According to the Woodbury formula, we have the following

$$\tilde{Z} = Z - ZS(T^{-1} + UZS)^{-1}UZ \ . \tag{2.3}$$

Since $(I - \alpha P)Z = I$, we have that $\alpha PZ = Z - I$ and consequently

$$UZ = \sum_{i=2}^{k+1} b_{i-1}(\alpha e_1^T Z - e_i^T(Z - I)) \ .$$

Now we can calculate UZS:

$$UZS = \sum_{i=2}^{k+1}\sum_{j=2}^{k+1} b_{i-1}(e_i^T(Z - I) - \alpha e_1^T Z)e_j b_{j-1}^T$$

$$= \sum_{i=2}^{k+1}\sum_{j=2}^{k+1} b_{i-1}(z_{ij} - \delta_{ij} - \alpha z_{1j})b_{j-1}^T \ .$$

The entry in row i and column j in the $k \times k$ matrix $M = T^{-1} + UZS$ is

$$m_{ij} = \delta_{ij}(k_{i+1} + 1) + z_{i+1j+1} - \delta_{ij} - \alpha z_{1j+1}$$

$$= \delta_{ij}k_{i+1} + z_{i+1j+1} - \alpha z_{1j+1} \ .$$
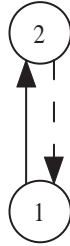
Figure 2.2: Node 1 can gain a lot by obtaining the dashed link from node 2.

Now we multiply (2.3) with $\frac{1-\alpha}{n}e^T$ from the left and e_p from the right. By using (2.1), we get

$$\tilde{\pi}_p = \pi_p - \pi^T SM^{-1}UZe_p \ .$$

The i'th entry in the k-dimensional column vector $q = UZe_p$ is

$$q_i = \alpha z_{1p} - z_{i+1p} + \delta_{i+1p} \ .$$

The i'th entry in the k-dimensional row vector $-\pi^T S$ is π_{i+1}. $\qquad\square$

Theorem 2.1 shows that knowing the degrees and the entries of Z and π for p and the nodes involved in the update is sufficient for calculating $\tilde{\pi}_p$. Informally, the PageRank values and the degrees of all nodes involved and the "distances" between them is sufficient information to predict the effect of an update.

2.4 Introductory Examples of Link Building

We now present some examples of Link Building problems. In the first example we will show that obtaining a link from an apparently unimportant node can have a dramatic effect on the PageRank value especially if the link is obtained in conjunction with links from important nodes. If node 1 only links to node 2 and node 2 is a sink as shown in Figure 2.2 then we might achieve $\tilde{\pi}_1 \approx \frac{1}{1-\alpha}\pi_1 = 6.7\pi_1$ by adding the reverse link $(2,1)$ and the PageRank value of node 2 will increase with roughly the factor $\frac{1}{1-\alpha^2} = 3.6$ (for $\alpha = 0.85$). This can be seen by using Theorem 2.1 in a graph with a big strongly connected component not containing the nodes 1 and 2 such that $z_{12} \approx \alpha$, $z_{21} \approx 0$, $z_{11} \approx 1$, $z_{22} \approx 1$ and $\pi_2 = (1+\alpha)\pi_1$ (It can also be seen by using Proposition 2.1 in [4] – See (4.10) in Section 4.2.3). Even in the case where node 1 is popular prior to the link modification, node 1 (and node 2) will benefit a lot if node 1 obtains the link $(2,1)$. Obtaining the link $(2,1)$ can more than triple the effect of a modification so once again the lack of memory for the random surfers plays a major role.

2.4.1 The Hexagon Examples

We now present some examples of link building problems involving a small graph where the nodes are organized as a hexagon connected with one link to a clique consisting of two nodes as shown in Figure 2.3a. Our objective is to identify new links pointing to node 1 maximizing $\tilde{\pi}_1$ – the PageRank value for node 1

(a) The original graph.

(b) One optimal new link.

(c) Two optimal new links.

(d) Two new links from the most popular nodes prior to the modification.
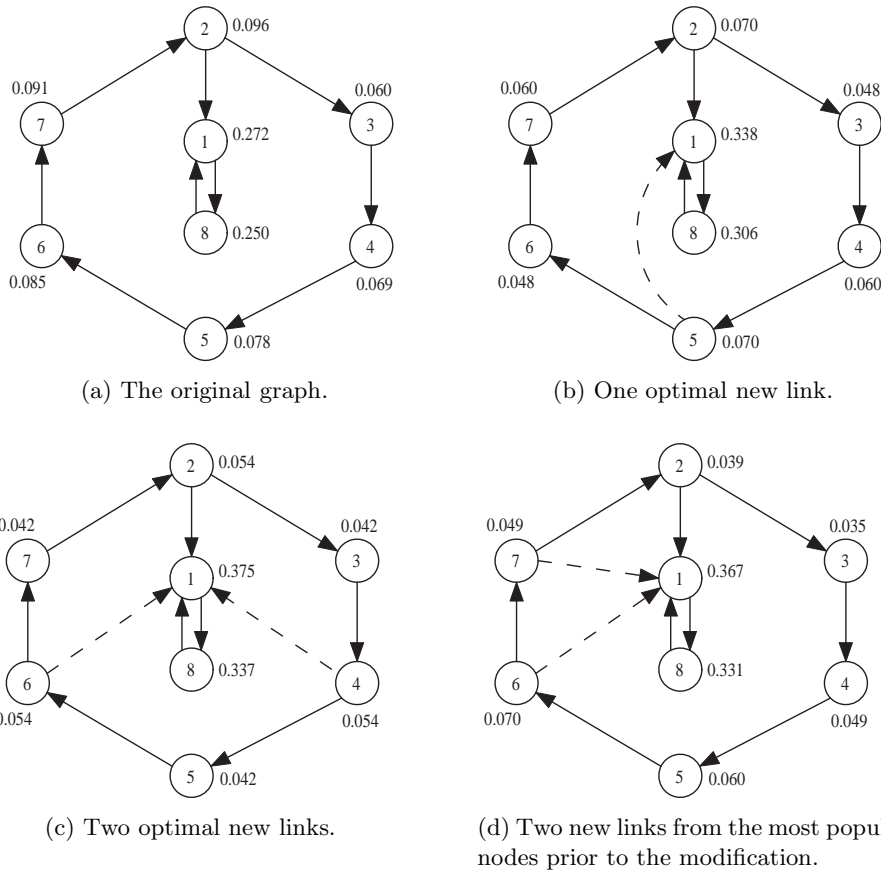
Figure 2.3: Link Building examples. The PageRank values for the *modified* graphs are shown besides the nodes.

after insertion of the links. We will use the unweighted version of PageRank in these examples. Figure 2.3b shows an optimal new link if we only look for one new link and Figure 2.3c shows an optimal set of two new links. The two most popular nodes in the set $\{3, \ldots, 7\}$ prior to the modification are the nodes 6 and 7. The examples show that adding links from the most popular nodes are not necessarily the optimal solution – even in the case where the most popular nodes have a low out degree. If we naively add the links $(6, 1)$ and $(7, 1)$ as shown in Figure 2.3d then we get the identity $\tilde{\pi}_1 = \pi_1 + 0.482\pi_6 + 0.594\pi_7 = 0.367$ by using Theorem 2.1. The optimal new links are $(4, 1)$ and $(6, 1)$ as shown in Figure 2.3c with corresponding identity $\tilde{\pi}_1 = \pi_1 + 0.665\pi_4 + 0.665\pi_6 = 0.375$. The coefficients in this identity are high compared to the naive approach which means that the price of the increase of π_1 is relatively low. The problem with the naive approach is that the topology of the network is ignored: the popular pages 6 and 7 are only a few clicks away from page 1 (z_{61} and z_{71} are high) and page 7 is only one click away from page 6 (z_{67} is high). We will analyze the characteristics of a "good" set of new backlinks more closely in Section 4.2.1.
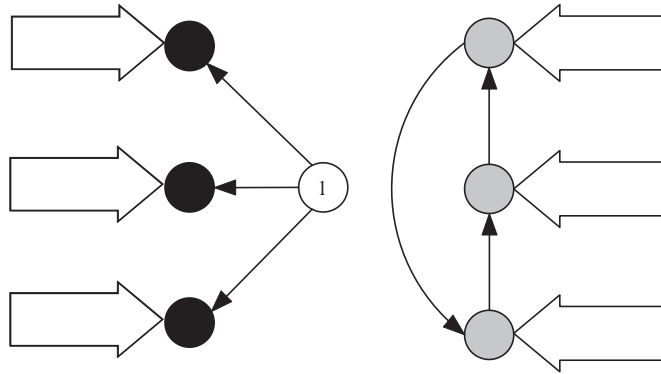
Figure 2.4: A directed graph where node 1 will gain a lot more by obtaining links from the black nodes compared to obtaining links from the grey nodes (assuming that all the grey and black nodes are solutions for the simple case $k = 1$)

2.4.2   Naive Link Building is Indeed Naive

In the final example in this chapter, we present a graph where the naive approach of choosing the k nodes with maximum values of $\tilde{\pi}_1(\{(u, 1)\})$ is shown to produce a very poor solution to the Link Building problem – for this approach we simply compute π_1 in $G(V, E \cup \{(u, 1)\})$ for each u and we choose the k u-nodes producing the biggest values of π_1. The details of the analysis and the construction of the graph can be found in Section 4.2.2 and part of the graph is shown in Figure 2.4 where the big arrows symbolize that the grey and black nodes have other nodes linking to them such that $\tilde{\pi}_1(\{(u, 1)\})$ is only slightly bigger for the grey nodes compared to the black nodes. If node 1 obtains links from all 3 black nodes, the PageRank value of node 1 will be roughly 6 times bigger compared to the PageRank value node 1 will achieve if node 1 obtains links from all 3 grey nodes. If the graph contains k grey nodes and k black nodes, this factor will tend to roughly 14 as k tends to infinity. *So naively picking "strong" nodes for the simple case $k = 1$ will lead to a PageRank value for node 1 which is roughly $\frac{1}{14}$ times the optimal value if the number of black and grey nodes is big!* The reason is that the grey nodes are strong candidates for the case $k = 1$ because of the cycle that boosts the PageRank for the participating nodes. Adding links from *all* the grey nodes will "ruin" the cycle. The black nodes will, on the other hand, become stronger if they all link to node 1 in which case the random surfers will revisit node 1 many times.

Chapter 3

Lower Bounds for Link Building

In this chapter we present intractability results for Link Building. In Section 3.1 we consider the variant MAX-MIN PAGERANK where the goal is to maximize the minimum PageRank value for a given set of nodes $T \subseteq V$ by adding k new links from $V \times V$. As mentioned in Section 1.3.2 then the first intractability results were obtained using this model of the problem. We show that MAX-MIN PAGERANK is NP-hard – this result was published in [71].

Section 3.2 starts with a brief introduction to the complexity classes PTAS, FPTAS and W[1]. After the introduction we turn our attention to the more realistic formulation of the Link Building problem presented as Definition 1.1 and prove stronger intractability results compared to the max–min formulation. Using Theorem 2.1 we show that no FPTAS exists for LINK BUILDING under the assumption NP≠P and we also show that LINK BUILDING is W[1]-hard. We also consider the computational complexity of the variant of Link Building where we are allowed to add or remove links with source t besides adding k new backlinks to t. Finally, we examine the variant where we for each page p have a cost $c(p) \in \mathbb{Z}^+ \cup \{+\infty\}$ for obtaining the link (p, t) and where the objective is to maximize the PageRank value of t for a given budget $B \in \mathbb{Z}^+$. The cost models the price or the difficulty of obtaining (p, t) as discussed in Section 1.1.2. These results are presented in [68].

3.1   MAX-MIN PAGERANK is NP-hard

A natural question to ask for a set of pages T and numbers x and k is the following: "Is it possible for all the pages in T to achieve a PageRank value greater than x by adding k new links anywhere in the web graph?". This is an informal way to phrase the decision version of the following optimization problem:

Definition 3.1 *MAX-MIN PAGERANK problem:*

- *Instance: A weighted directed graph $G(V, E)$ with positive integer weights on the edges, a subset of nodes $T \subseteq V$ and a number $k \in \mathbb{Z}^+$.*

- *Solution: A set $S \subseteq \{(u, v) \in V \times V : u \neq v\}$ with $|S| = k$ maximizing $\min_{t \in T} \tilde{\pi}_t(S)$.*

We allow multiple occurrences of (u, v) in S.

Please note that the solution to the MAX-MIN PAGERANK problem is a set of *edges* as opposed to the LINK BUILDING problem from Definition 1.1 where the solution is a set of *nodes*. The MAX-MIN PAGERANK problem is solvable in polynomial time if k is a fixed constant in which case we can simply calculate $\tilde{\pi}(S)$ for all possible S. If k is part of the input then the problem is NP-hard which is formally stated by the following theorem:

Theorem 3.1 *MAX-MIN PAGERANK is NP-hard.*
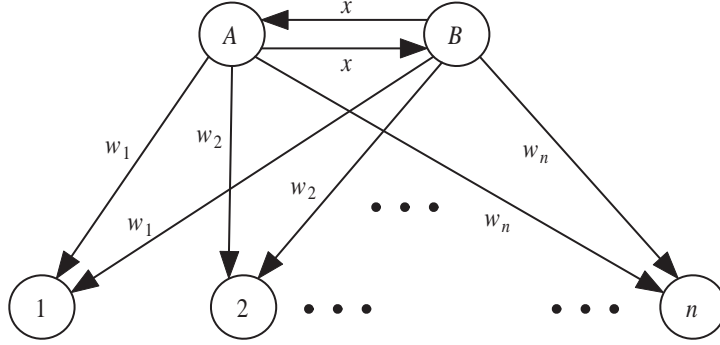
Figure 3.1: A directed graph with weights indicating the number of occurrences of the links.

Theorem 3.1 is proved by reduction from the NP-complete balanced version of the PARTITION problem [43, page 223]. The rest of this section gives the proof in detail.

In order to prove that MAX-MIN PAGERANK is NP-hard when k is part of the input we need three lemmas concerning the graph in Figure 3.1 where the weight of a link is the number of occurrences in E. The intuition behind the lemmas and the proof is the following: The nodes A and B are identical twins devoted to each other – the number of links x between them is big – and they share the same view on the world by assigning the same weight w_i to any other node i in the network. Suppose that you would like to maximize $\min(\tilde{\pi}_A, \tilde{\pi}_B)$ with n new links. The best you can do is to add one new link from every node in $\{1, \ldots, n\}$ to either A or B such that $\tilde{\pi}_A = \tilde{\pi}_B$. It turns out that we have to split the friends of A and B in two groups of equal cardinality and weight to achieve $\tilde{\pi}_A = \tilde{\pi}_B$ and let one group link to A and the other group link to B. Splitting the friends is a well known NP-complete problem [43, page 223].

In the following we let $N = \{1, \ldots, n\}$ and $W = \sum_{i=1}^{n} w_i$. We will write $\tilde{\pi}_{AB}(E')$ as a shorthand for $\tilde{\pi}_A(E') + \tilde{\pi}_B(E')$. We will now formally introduce the term *sum-optimal* and justify this definition in the two subsequent lemmas.

Definition 3.2 *A set of links E' is called* sum-optimal *if*

$$\forall i \in N : (i, A) \in E' \lor (i, B) \in E' \ .$$

In Lemma 3.1 we show that we achieve the same value for $\tilde{\pi}_A + \tilde{\pi}_B$ for all sum-optimal sets of n links. In Lemma 3.2 we show that we will achieve a lower value of $\tilde{\pi}_A + \tilde{\pi}_B$ for any other set of links.

In Lemma 3.3 we show that we can achieve $\tilde{\pi}_A = \tilde{\pi}_B$ for a sum-optimal set of n links *if and only if* we can split the friends of A and B in two groups of equal cardinality and weight. The three lemmas show that we can identify such a potential split by maximizing $\min(\tilde{\pi}_A, \tilde{\pi}_B)$.

Lemma 3.1 *Consider the graph in Figure 3.1. If E_1' and E_2' denote two arbitrary sum-optimal sets of n links then we have the following:*

$$\tilde{\pi}_{AB}(E_1') = \tilde{\pi}_{AB}(E_2') \ . \tag{3.1}$$

Proof. Let E' be an arbitrary sum-optimal set of n links. The only nodes that link to the nodes in N are A and B and A and B both use a fraction of $\frac{W}{W+x}$ of their links on N. Since no node in N is a sink and the sum of PageRank values of the nodes in N is $1 - \tilde{\pi}_{AB}(E')$ we have the following:

$$1 - \tilde{\pi}_{AB}(E') = (1 - \alpha)\frac{n}{n+2} + \alpha\tilde{\pi}_{AB}(E')\frac{W}{W+x} \ . \tag{3.2}$$

From (3.2) we obtain an expression for $\tilde{\pi}_{AB}(E')$ that proves (3.1):

$$\tilde{\pi}_{AB}(E') = \frac{1 - (1 - \alpha)\frac{n}{n+2}}{1 + \alpha\frac{W}{W+x}} \ .$$

$\square$

Lemma 3.2 *Let x satisfy the following inequality:*

$$x > \frac{W(n+2)^2}{n(1-\alpha)} - W \ . \tag{3.3}$$

If E' is an arbitrary sum-optimal set of n links and L is an arbitrary set of links which is not sum-optimal then we have that

$$\tilde{\pi}_{AB}(E') > \tilde{\pi}_{AB}(L) \ . \tag{3.4}$$

Proof. There has to be at least one node $u \in N$ that does not link to A and does not link to B since L is not sum-optimal. A fraction of $1 - \alpha$ of the Page-Rank value of u is spread uniformly on all nodes. No matter whether u is a sink or not then it will spread at least a fraction $\frac{n}{n+2}$ of the remaining part of its PageRank value to the other nodes in N. The PageRank value of u is greater than $\frac{1-\alpha}{n+2}$ which enables us to establish the following inequality:

$$1 - \tilde{\pi}_{AB}(L) > (1 - \alpha)\frac{n}{n+2} + \alpha\frac{1-\alpha}{n+2} \cdot \frac{n}{n+2} \ . \tag{3.5}$$

From (3.3) we get $\frac{(1-\alpha)n}{(n+2)^2} > \frac{W}{W+x}$. Now we use (3.2), (3.5) and $\tilde{\pi}_{AB}(E') < 1$ to conclude that $1 - \tilde{\pi}_{AB}(L) > 1 - \tilde{\pi}_{AB}(E')$ that proves (3.4). $\square$

Lemma 3.3 *Let E' denote an arbitrary sum-optimal set of n links and let x satisfy*

$$x > \frac{\alpha W(n+2)}{1-\alpha} - W \ . \tag{3.6}$$

Let $A_{\leftarrow} = \{i \in N : (i, A) \in E'\}$. The set $A_{\leftarrow}$ consists of the nodes in N that link to A. We define $W_{A_{\leftarrow}} = \sum_{i \in A_{\leftarrow}} w_i$. We also define $B_{\leftarrow}$ and $W_{B_{\leftarrow}}$ accordingly.

The following two statements are equivalent where E' is omitted as an argument for $\tilde{\pi}_A$ and $\tilde{\pi}_B$:

1. $W_{A_{\leftarrow}} = W_{B_{\leftarrow}} \wedge |A_{\leftarrow}| = |B_{\leftarrow}|$.

2. $\tilde{\pi}_A = \tilde{\pi}_B$.

Proof. Let $\tilde{\pi}_{A\leftarrow}$ and $\tilde{\pi}_{B\leftarrow}$ denote the sum of PageRank values for the two sets $A_\leftarrow$ and $B_\leftarrow$ respectively. Following the same line of reasoning as used in the proof of Lemma 3.1 we have the following:

$$\tilde{\pi}_A = \frac{1-\alpha}{n+2} + \alpha\tilde{\pi}_{A\leftarrow} + \alpha\frac{x}{x+W}\tilde{\pi}_B \tag{3.7}$$

$$\tilde{\pi}_B = \frac{1-\alpha}{n+2} + \alpha\tilde{\pi}_{B\leftarrow} + \alpha\frac{x}{x+W}\tilde{\pi}_A \tag{3.8}$$

$$\tilde{\pi}_{A\leftarrow} = |A_\leftarrow|\frac{1-\alpha}{n+2} + \alpha\frac{W_{A\leftarrow}}{W+x}(\tilde{\pi}_A + \tilde{\pi}_B) \tag{3.9}$$

$$\tilde{\pi}_{B\leftarrow} = |B_\leftarrow|\frac{1-\alpha}{n+2} + \alpha\frac{W_{B\leftarrow}}{W+x}(\tilde{\pi}_A + \tilde{\pi}_B) \ . \tag{3.10}$$

$1 \Rightarrow 2$: Assume that $W_{A\leftarrow} = W_{B\leftarrow}$ and $|A_\leftarrow| = |B_\leftarrow|$ for a sum-optimal set E' consisting of n links. By using (3.9) and (3.10) we conclude that $\tilde{\pi}_{A\leftarrow} = \tilde{\pi}_{B\leftarrow}$. By solving (3.7) and (3.8) we get that $\tilde{\pi}_A = \tilde{\pi}_B$.

$2 \Rightarrow 1$: Assume that $\tilde{\pi}_A = \tilde{\pi}_B$ for a sum-optimal set E' of n links. In this case we can conclude that $\tilde{\pi}_{A\leftarrow} = \tilde{\pi}_{B\leftarrow}$ by using (3.7) and (3.8). If $x > \frac{\alpha W(n+2)}{1-\alpha} - W$ then $\frac{1-\alpha}{n+2} > \alpha\frac{W}{W+x}$. This means that the last term in (3.9) and (3.10) are smaller than $\frac{1-\alpha}{n+2}$. We conclude that $|A_\leftarrow| = |B_\leftarrow|$ with $W_{A\leftarrow} = W_{B\leftarrow}$ as a consequence. $\qquad\square$

We are now in a position to prove Theorem 3.1.
Proof. We show how to solve an instance of the balanced version of the PAR-TITION problem [43, page 223] – which is known to be NP-complete – in polynomial time if we are allowed to consult an *oracle*[1] for solutions to the MAX-MIN PAGERANK problem.

For an instance of the balanced version of PARTITION we have a $w_i \in \mathbb{Z}^+$ for each $i \in N$. The question is whether a subset $N' \subset N$ exists such that $\sum_{i\in N'} w_i = \sum_{i\in N-N'} w_i$ and $|N'| = |N - N'|$.

In polynomial time we transform this instance into an instance of MAX-MIN PAGERANK given by the graph G in Figure 3.1 with $x = \frac{W(n+2)^2}{n(1-\alpha)}$, $T = \{A, B\}$ and $k = n$. We claim that the following two statements are equivalent:

1. $N' \subset N$ exists such that $\sum_{i\in N'} w_i = \sum_{i\in N-N'} w_i$ and $|N'| = |N - N'|$.

2. The solution S to the MAX-MIN PAGERANK instance is a sum-optimal set of links with $W_{A\leftarrow} = W_{B\leftarrow}$ and $|A_\leftarrow| = |B_\leftarrow|$.

$1 \Rightarrow 2$: Let $E' = [N' \times \{A\}] \cup [(N - N') \times \{B\}]$. According to Lemma 3.1 and Lemma 3.2 then $\tilde{\pi}_{AB}(E')$ is at its maximum compared to any other set of n new links. According to Lemma 3.3 we also have that $\tilde{\pi}_A(E') = \tilde{\pi}_B(E')$. This means that $\min(\tilde{\pi}_A(E'), \tilde{\pi}_B(E'))$ is at its maximum. The solution S to

---

[1]An oracle is a hypothetical computing device that can compute a solution in a *single step* of computation.

the MAX-MIN PAGERANK instance must match this value so S must be sum-optimal (Lemma 3.2) with $\tilde{\pi}_A(S) = \tilde{\pi}_B(S)$. According to Lemma 3.3 then $W_{A_\leftarrow} = W_{B_\leftarrow}$ and $|A_\leftarrow| = |B_\leftarrow|$ for S.

$2 \Rightarrow 1$: Take $N' = A_\leftarrow$.

We can now solve the PARTITION instance by checking whether 2) is satisfied in the solution of the MAX-MIN PAGERANK instance. The checking procedure can be done in polynomial time. □

3.2 LINK BUILDING is W[1]-hard and Allows no FPTAS

Before presenting the intractability results for the LINK BUILDING problem defined in Definition 1.1 we provide a brief introduction to the involved complexity classes.

PTAS and FPTAS: Consider a maximization problem "$\arg\max_x f(x)$" with solution x^*. A FPTAS (Fully Polynomial Time Approximation Scheme) can compute an x such that $f(x) \geq (1 - \epsilon)f(x^*)$ in time polynomial in $\frac{1}{\epsilon}$ and the size of the instance. Some NP-hard problems allow a FPTAS (for example the Knapsack problem) and some do not. If there is no FPTAS for a problem there is still a chance for a PTAS (Polynomial Time Approximation Scheme) where we can obtain x in polynomial time for any *fixed* ϵ. As an example an algorithm with running time $n^{\frac{1}{\epsilon}}$ counts as a PTAS but not as an FPTAS for a problem with instance size n.

FPT and W[1]: We will say that a problem with instance size n involving a parameter k is *fixed parameter tractable* if it can be solved in time $f(k)n^c$ where f is some function and c is independent of k. The class FPT contains the *decision* problems with this property. We will write A $\leq$ B if the problem A can be reduced to the problem B preserving fixed parameter tractability in the sense that B $\in$ FPT $\Rightarrow$ A $\in$ FPT. Consider the problems VERTEX COVER and INDEPENDENT SET where we have to decide whether a graph contains a vertex cover[2] of size k or an independent set[3] of size k respectively. FPT is contained in the complexity class W[1] = {P : P $\leq$ INDEPENDENT SET}. Even though VERTEX COVER is NP-complete it has been solved for large n and $k = 400$ [19]. The reason is that VERTEX COVER $\in$ FPT with moderate f and c. A corresponding breakthrough is believed to be impossible for INDEPENDENT SET since there is strong evidence in the literature that FPT$\neq$W[1] so hardness for W[1] is accepted as evidence that a problem is fixed parameter *intractable*. According to a recent paper [20] then the currently "best algorithm" for INDEPENDENT SET runs in time $O(n^{0.792k})$ where the exponent

---

[2]A vertex cover is a subset of the nodes satisfying that every edge has at least one endpoint in the set.

[3]A set of nodes in a graph is independent if no edge connects two of the nodes.

of n increases dramatically with k. For more information on FPT and W[1] we refer to [31].

We will show that LINK BUILDING is intractable by reduction from the independent set problem restricted to undirected regular[4] graphs. This problem is known to be NP-complete even for 3-regular graphs [32, 43]. To be more precise we will show that no FPTAS for LINK BUILDING exists under the assumption NP$\neq$P. Intuitively we build a directed graph where all nodes in the original graph have the same out degree and PageRank value and where neighbors in the original graph will be very "close" to each other – wrt. to the Z-matrix – compared to non-neighbors. Obtaining a link to t from u will only have a significant negative effect on the PageRank values of the neighbors of u so obtaining links from an independent set is preferable. In this way we can solve the independent set problem by doing link building. We need a couple of definitions to clarify matters:

Definition 3.3 *The* REGULAR INDEPENDENT SET *problem:*

- *Instance: An undirected regular graph $H(V_H, E_H)$ and an integer $k \geq 2$.*

- *Question: Does H contain an* independent set *of size k?*

Definition 3.4 *Let S^* be a solution to the LINK BUILDING problem. A* FPTAS *for the LINK BUILDING problem is an algorithm that given input (G, t, k, ϵ) computes a feasible solution S to the LINK BUILDING problem satisfying*

$$\tilde{\pi}_t(S \times \{t\}) > (1 - \epsilon)\tilde{\pi}_t(S^* \times \{t\})$$

in time polynomial in $\frac{1}{\epsilon}$ and the size of (G, t, k).

We will now formally state the first main theorem of this section:

Theorem 3.2 *If NP$\neq$P then there is no FPTAS for LINK BUILDING.*

Please note that the proof of Theorem 3.2 uses Lemma 3.4 presented after the proof in an attempt to make the structure of the proof clear to the reader. *Proof.*

We show how to solve an instance of the REGULAR INDEPENDENT SET problem in polynomial time if we have a FPTAS to the LINK BUILDING problem at our disposal.

Now let the regular graph $H(V_H, E_H)$ and the number $k \in \mathbb{Z}^+$ represent an instance of the REGULAR INDEPENDENT SET problem and let d denote the degree of all nodes in H. From $H(V_H, E_H)$ we now construct the graph $G(V_G, E_G)$ shown in Figure 3.2 in polynomial time in the following way:

1. The nodes in G are all the nodes in H together with four new nodes r, s, t and w: $V_G = V_H \cup \{r, s, t, w\}$.

2. We add links (r, s) and (s, r) with multiplicity y where y satisfies (3.13) in Lemma 3.4 below.

---

[4]A regular graph is a graph where all nodes have the same degree.
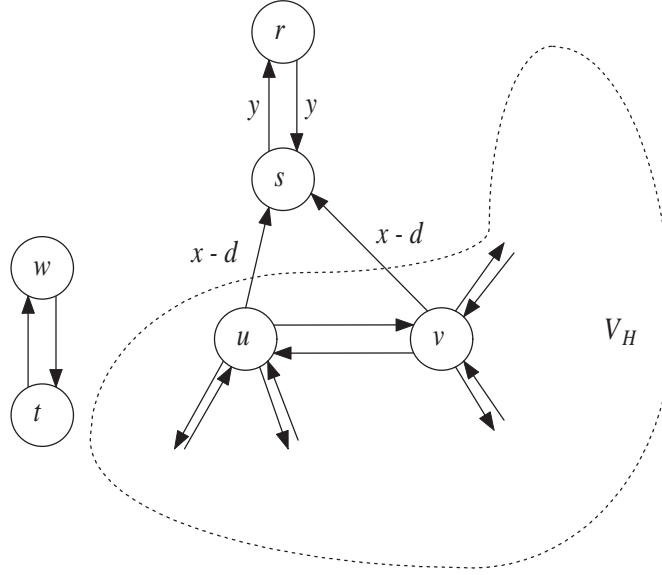
Figure 3.2: The graph $G(V_G, E_G)$.

3. For every node $v \in V_H$ we add a link (v, s) with multiplicity $x - d$ where x satisfies (3.12) in Lemma 3.4 below.

4. For every edge $\{u, v\} \in E_H$ we add two links (u, v) and (v, u) to E_G with multiplicity 1.

5. Finally, we add links (t, w) and (w, t) with multiplicity 1.

Let $n = |V_G|$. Now assume that H contains an independent set of size k. From Lemma 3.4 below we conclude that any solution S^* to the LINK BUILDING problem must be independent and that a constant ρ exists such that the following holds for any feasible solution S which is *not* an independent set in H:

$$\tilde{\pi}_t(S \times \{t\}) < (1 - \rho n^{-1} d^{-6} k^{-9}) \tilde{\pi}_t(S^* \times \{t\}) \ . \tag{3.11}$$

This shows that we can decide whether an independent set exists by activating our LINK BUILDING FPTAS with input $(G, t, k, \epsilon = \rho n^{-1} d^{-6} k^{-9})$ and check whether the solution from the FPTAS is independent. Thus we can solve the REGULAR INDEPENDENT SET problem in polynomial time using the LINK BUILDING FPTAS implying NP=P. $\qquad \square$

Lemma 3.4 *Let $S_1 \subseteq V_H$ be an arbitrary independent set in H and let $S_2 \subseteq V_G \backslash \{t\}$ be an arbitrary set with $|S_1| = |S_2| = k$. If x and y satisfy the following:*

$$x = \frac{2d^2 k^3}{1 - \alpha^2} \ . \tag{3.12}$$

$$y > 2(x + 1) \frac{n}{1 - \alpha} \left(\frac{x - d\alpha}{x - d\alpha k} \right) - 1 \ . \tag{3.13}$$

then the following holds if S_2 is not *an independent set in H where ρ is a positive constant only dependent on α:*

$$\tilde{\pi}_t(S_1 \times \{t\}) - \tilde{\pi}_t(S_2 \times \{t\}) > \rho n^{-1} d^{-6} k^{-9} \ . \tag{3.14}$$

Proof. First we will deal with the case where $S_2 \subseteq V_H$ is a non-independent set in H. In order to align the proof with Theorem 2.1 we will refer to t as node 1 and refer to the nodes in S_1 and S_2 as nodes 2, 3, ..., $k+1$.

According to Theorem 2.1 we have the following:

$$\tilde{\pi}_1 - \pi_1 = \begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} M^{-1} q \ . \tag{3.15}$$

Let $B = \{b_{ij}\}$ be the $k \times k$ matrix defined by the following identities:

$$b_{ij} = \frac{z_{i+1j+1}}{x+1} \text{ if } i \neq j \ . \tag{3.16}$$

$$b_{ii} = \frac{z_{i+1i+1} - 1}{x+1} \ . \tag{3.17}$$

Now we have

$$M = (x+1)(I+B) \ .$$

If $\bar{b}$ is an upper bound on the entries in B then it is not hard to show that $k^{s-1}\bar{b}^s$ is an upper bound on the entries in B^s:

$$0 \leq B^s \leq k^{s-1}\bar{b}^s E = k^{-1}(k\bar{b})^s E \ . \tag{3.18}$$

For S_1 we can use the following upper bound:

$$\bar{b}_1 = \frac{1}{x+1}\left(\frac{d\alpha}{x}\right)^2 \frac{1}{1-\alpha^2} \leq \left(\frac{d^2}{x^3}\right)\left(\frac{\alpha^2}{1-\alpha^2}\right) \ . \tag{3.19}$$

Here we use that $\left(\frac{d\alpha}{x}\right)$ is the probability of following a link and staying in V_H for a random surfer starting in V_H. Because S_1 is independent then $\left(\frac{d\alpha}{x}\right)^2$ is an upper bound on the probability of reaching j from node i without zapping. We also use that $1 + \alpha^2 + \alpha^4 + \alpha^6 + \cdots = \frac{1}{1-\alpha^2}$ is an upper bound on z_{jj}.

We also get an upper bound for S_2:

$$\bar{b}_2 = \frac{1}{x+1}\left(\frac{d\alpha}{x}\right)\frac{1}{1-\alpha^2} \leq \left(\frac{d}{x^2}\right)\left(\frac{\alpha}{1-\alpha^2}\right) \ . \tag{3.20}$$

For $x = \frac{2d^2k^3}{1-\alpha^2}$ we have $k\bar{b} < 1$ and hence we have the following:

$$M^{-1} = \frac{1}{x+1}(I - B + B^2 - B^3 + B^4 - \cdots) = \frac{1}{x+1}\sum_{s=0}^{\infty}(-1)^s B^s \ . \tag{3.21}$$

Now consider a probability distribution w on V_G with the same probability mass for each entry corresponding to a node in V_H. All entries in $w^T Q^i$ corresponding to nodes in V_H will have the same probability mass for any i because

H is regular. The limiting distribution π^T will also have this property. This means that a number β exists such that:

$$\begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} = \beta e^T \ . \tag{3.22}$$

The vector q is given by the following identity:

$$q = \frac{\alpha}{1-\alpha^2} e \ . \tag{3.23}$$

We now insert the results from (3.21), (3.22) and (3.23) in (3.15):

$$\tilde{\pi}_1 - \pi_1 = \frac{\alpha\beta}{(x+1)(1-\alpha^2)} \sum_{s=0}^{\infty} (-1)^s e^T B^s e \ . \tag{3.24}$$

We will now use (3.18) to establish a lower bound of the factor $\sum_{s=0}^{\infty}(-1)^s e^T B^s e$ for S_1:

$$\sum_{s=0}^{\infty} (-1)^s e^T B^s e \ge k(1 - \bar{b}_1 k - (\bar{b}_1 k)^3 - (\bar{b}_1 k)^5 - \cdots) = k\left(1 - \frac{\bar{b}_1 k}{1-(\bar{b}_1 k)^2}\right) \ . \tag{3.25}$$

We will now develop an upper bound for $\sum_{s=0}^{\infty}(-1)^s e^T B^s e$ for S_2. There are two nodes u and v in S_2 such that $(u,v),(v,u) \in E_G$. The probability of reaching v for a random surfer starting at u – preceding the first zapping event – is greater than $\frac{\alpha}{x}$:

$$b_{vu}, b_{uv} \ge \frac{1}{x+1}\frac{\alpha}{x} \ge \frac{1}{x^2}\frac{\alpha}{2} \ . \tag{3.26}$$

Now we can construct the desired upper bound:

$$\sum_{s=0}^{\infty} (-1)^s e^T B^s e \le k(1 - \frac{1}{k}(b_{uv} + b_{vu}) + (\bar{b}_2 k)^2 + (\bar{b}_2 k)^4 + (\bar{b}_2 k)^6 + \cdots) \tag{3.27}$$

$$= k\left(1 - \frac{1}{k}(b_{uv} + b_{vu}) + \frac{(\bar{b}_2 k)^2}{1-(\bar{b}_2 k)^2}\right) \ .$$

By inserting the lower bound from (3.25) and the upper bound from (3.27) in (3.24) we now conclude that

$$\tilde{\pi}_t(S_1 \times \{t\}) - \tilde{\pi}_t(S_2 \times \{t\}) \ge$$

$$\frac{\alpha\beta}{(x+1)(1-\alpha^2)} k\left(\frac{1}{k}(b_{uv} + b_{vu}) - \frac{(\bar{b}_2 k)^2}{1-(\bar{b}_2 k)^2} - \frac{\bar{b}_1 k}{1-(\bar{b}_1 k)^2}\right) \ . \tag{3.28}$$

For $x = \frac{2d^2 k^3}{1-\alpha^2}$ we have that $(\bar{b}_1 k)^2$ and $(\bar{b}_2 k)^2$ are both less than $\frac{1}{2}$ which implies the following where we also use (3.19), (3.20) and (3.26):

$$\frac{1}{k}(b_{uv} + b_{vu}) - \frac{(\bar{b}_2 k)^2}{1-(\bar{b}_2 k)^2} - \frac{\bar{b}_1 k}{1-(\bar{b}_1 k)^2} \ge$$

$$\frac{1}{k}(b_{uv} + b_{vu}) - 2(\bar{b}_2 k)^2 - 2\bar{b}_1 k \ge$$

$$\frac{1}{k}\frac{\alpha}{x^2} - 2\left(\frac{d}{x^2}\right)^2\left(\frac{\alpha}{1-\alpha^2}\right)^2 k^2 - 2\left(\frac{d^2}{x^3}\right)\left(\frac{\alpha^2}{1-\alpha^2}\right)k =$$

$$k^{-1}x^{-2}\left(\alpha - 2\left(\frac{d^2}{x^2}\right)\left(\frac{\alpha}{1-\alpha^2}\right)^2 k^3 - 2\left(\frac{d^2}{x}\right)\left(\frac{\alpha^2}{1-\alpha^2}\right)k^2\right) =$$

$$k^{-1}x^{-2}\left(\alpha - \frac{1}{2}\alpha^2 d^{-2}k^{-3} - \alpha^2 k^{-1}\right) \geq$$

$$k^{-1}x^{-2}\left(\alpha - \frac{1}{16}\alpha^2 - \frac{1}{2}\alpha^2\right) \quad .$$

We now use this inequality together with $\beta > \frac{1-\alpha}{n}$ and $2x > x+1$ to replace the lower bound in (3.28):

$$\tilde{\pi}_t(S_1 \times \{t\}) - \tilde{\pi}_t(S_2 \times \{t\}) \geq$$

$$\frac{\alpha(1-\alpha)}{2(1-\alpha^2)}x^{-1}n^{-1}k \cdot k^{-1}x^{-2}\left(\alpha - \frac{1}{16}\alpha^2 - \frac{1}{2}\alpha^2\right) =$$

$$\frac{\alpha(1-\alpha)}{2(1-\alpha^2)}n^{-1}x^{-3}\left(\alpha - \frac{1}{16}\alpha^2 - \frac{1}{2}\alpha^2\right) \quad .$$

which shows that (3.14) holds.

Up till now we have shown that (3.14) holds if $S_1 \subseteq V_H$ is an independent set from H and $S_2 \subseteq V_H$ is *not* an independent set from H. In the remaining part of the proof we will show that $\tilde{\pi}_t(S_1 \times \{t\}) > \tilde{\pi}_t(S_2 \times \{t\})$ holds if S_1 is *any* subset of V_H and $S_2 \subseteq V_G \setminus \{t\}$ is a subset of V_G such that $|S_1| = |S_2| = k$ and $S_2 \cap \{r, s, w\} \neq \emptyset$ provided that y satisfies (3.13). Let $\tilde{\pi}_t^{(1)}$ denote $\tilde{\pi}_t(S_1 \times \{t\})$ and let $\tilde{\pi}_t^{(2)}$ denote $\tilde{\pi}_t(S_2 \times \{t\})$.

We now compute the PageRank value π_v for v in G for any $v \in V_H$. All nodes in V_H have the same PageRank value in G as shown above:

$$\pi_v = \frac{1-\alpha}{n} + \frac{d\alpha}{x}\pi_v$$

From this identity we get the following:

$$\pi_v = \frac{1-\alpha}{n}\frac{x}{x-d\alpha}$$

Let $\tilde{\pi}_v$ denote the new PageRank value for $v \in V_H$ if one or more nodes establishes links to t. In this case the PageRank value of v can not increase (Theorem 2.1):

$$\frac{1-\alpha}{n} < \tilde{\pi}_v \leq \frac{1-\alpha}{n}\frac{x}{x-d\alpha} = \pi_v$$

Now we have that

$$\tilde{\pi}_t^{(2)} \leq \alpha\left(\alpha\tilde{\pi}_t^{(2)} + \frac{1-\alpha}{n}\right) + (k-1)\frac{\alpha}{x+1}\frac{1-\alpha}{n}\frac{x}{x-d\alpha} + 2\frac{\alpha}{y+1} + \frac{1-\alpha}{n}$$

which is equivalent to

$$(1 - \alpha^2)\tilde{\pi}_t^{(2)} \leq \alpha\frac{1 - \alpha}{n} + (k - 1)\frac{\alpha}{x + 1}\frac{1 - \alpha}{n}\frac{x}{x - d\alpha} + 2\frac{\alpha}{y + 1} + \frac{1 - \alpha}{n} \quad (3.29)$$

We also have that

$$\tilde{\pi}_t^{(1)} \geq \alpha\left(\alpha\tilde{\pi}_t^{(1)} + \frac{1 - \alpha}{n}\right) + k\frac{\alpha}{x + 1}\frac{1 - \alpha}{n} + \frac{1 - \alpha}{n}$$

which is equivalent to

$$(1 - \alpha^2)\tilde{\pi}_t^{(1)} \geq \alpha\frac{1 - \alpha}{n} + k\frac{\alpha}{x + 1}\frac{1 - \alpha}{n} + \frac{1 - \alpha}{n} \quad (3.30)$$

We just have to choose y such that the upper bound in (3.29) is smaller than the lower bound in (3.30) – or such that the difference between the lower and upper bound is positive:

$$\frac{\alpha}{x + 1}\frac{1 - \alpha}{n}\left(k - (k - 1)\frac{x}{x - d\alpha}\right) - 2\frac{\alpha}{y + 1}$$

$$= \frac{\alpha}{x + 1}\frac{1 - \alpha}{n}\left(\frac{x - d\alpha k}{x - d\alpha}\right) - 2\frac{\alpha}{y + 1} > 0$$

This holds if (3.13) holds. □

The REGULAR INDEPENDENT SET problem is W[1]-complete [14] so we immediately get the second main theorem of this section because k is preserved and because the construction of G and the check of independence runs in polynomial time in the reduction in the proof of Theorem 3.2:

Theorem 3.3 *If W[1]$\neq$FPT then LINK BUILDING is not fixed parameter tractable.*

Theorem 3.3 also holds if we are allowed to add or delete links with source t besides adding k new backlinks to t because the link structure regarding links with source t is optimal in G according to [4].

In a real setting backlinks are obviously hard or even impossible to obtain (see Section 1.1.2). We can model this by assigning a cost $c(p) \in \mathbb{Z}^+ \cup \{+\infty\}$ to each page p for obtaining the link (p, t). We can now slightly change the Link Building problem so the objective is to maximize $\tilde{\pi}_t$ for a given *budget* $B \in \mathbb{Z}^+$ – the total cost of the links obtained should not exceed B. This is a generalization of the original problem from Definition 1.1 where we have $B = k$ and $c(p) = 1$ so the intractability results also hold for this formulation – with B as the parameter. The results even hold for this variant in the unweighted PageRank model where multiple links from one page to another is treated as one: we just have to replace r and s in Figure 3.2 with a clique of $x - d$ nodes and let all nodes in V_H link to all nodes in the clique. The budget should be k, all the nodes in the clique should have cost $+\infty$ and all other nodes should have cost 1.

Chapter 4

Upper Bounds for Link Building

In this chapter we look at the Link Building problem from the more positive side compared to the preceding chapter. In Section 4.1 we show how to solve the Link Building problem from Definition 1.1 with *fixed* $k = 1$ with a simple randomized algorithm using time corresponding to a *small* and *constant* number of PageRank computations. Results of experiments with the algorithm on artificial computer generated graphs and a crawl of the Danish part of the web graph are also reported. These results were published by the author of this dissertation in [71].

We present a greedy polynomial time algorithm for the unweighted case of Link Building in Section 4.2 computing a set of k new backlinks to t with a corresponding value of $\tilde{\pi}_t$ within a constant factor from the optimal value. In other words we prove that this variant of LINK BUILDING is a member of the complexity class APX. Based on Theorem 2.1 we also show how to construct a graph with a poor performance for the naive Link Building approach choosing the k u-nodes with the maximum values of π_t in $G(V, E \cup \{(u, t)\})$. These results are obtained recently together with Tasos Viglas, University of Sydney, and are to appear in [73].

In Section 4.3 we show how to attack the Link Building problem by using Mixed Integer Linear Programming (MILP). The work on the MILP approach is also recent and it is also joint work with Tasos Viglas [73].

4.1   An Efficient Algorithm for the Simplest Case

We now turn to the simplest variant of the Link Building problem where the objective is to pick *one* link pointing to a given page t in order to achieve the maximum increase in the PageRank value for t. This problem can be solved by brute force in polynomial time using n PageRank computations by computing π_t in $G(V, E \cup \{(u, t)\})$ for every $u \in V$. Our randomized algorithm "eliminates" the n-factor in the time complexity. The main message is that if we have the machinery capable of calculating the PageRank vector for the network then we can also solve the simple Link Building problem.

If page $j \neq t$ establishes a link to t then we have the following according to Theorem 2.1 (and Theorem 3.1 in [4] – the theorems are equivalent for $k = 1$):

$$\tilde{\pi}_t = \pi_t + \pi_j \frac{\alpha z_{tt} - z_{jt}}{k_j + z_{jj} - \alpha z_{tj}} \ . \tag{4.1}$$

The central idea for the Link Building algorithm is to avoid an expensive matrix inversion and only calculate the entries of Z playing a role in (4.1) for all $j \neq t$. We approximate z_{tt}, z_{tj} and z_{jt} for *all* $j \neq t$ by performing two calculations where each calculation has a running time comparable to one PageRank computation. The diagonal elements z_{jj} are approximated by a randomized scheme tracking a random surfer. When we have obtained approximations of all *relevant* entries of Z then we can calculate (4.1) in constant time for any given page j.

4.1.1   Approximating Rows and Columns of Z

We will use the following expression for Z [53]:

$$Z = (I - \alpha P)^{-1} = \sum_{i=0}^{+\infty} (\alpha P)^i \ . \tag{4.2}$$

In order to get row t from Z we multiply (4.2) with e_t^T from the left where e_t is a vector with a 1 at coordinate t and 0's elsewhere:

$$e_t^T Z = \sum_{i=0}^{+\infty} e_t^T (\alpha P)^i = e_t^T + e_t^T \alpha P + (e_t^T \alpha P)\alpha P + \cdots \ . \tag{4.3}$$

Equation (4.3) shows how to *approximate* row t in Z with a simple iterative scheme using the fact that each term in (4.3) is a row vector obtained by multiplying αP with the previous term from the left. We simply track a group of random surfers starting at page t and count the number of hits they produce on other pages preceding the first zapping event.

The elements appearing in a term are non negative and the sum of the elements in the $i'th$ term is α^{i-1} which can be shown by using the fact that $Pe = e$ where e is the vector with all 1's so the iterative scheme converges quickly for $\alpha = 0.85$. The iterative scheme has roughly the same running time as the power method for calculating PageRank and 50-100 iterations gives adequate precision for approximating the fraction in (4.1) since $z_{jj} \geq 1$ for all j.

By multiplying (4.2) with e_t from the right we obtain an iterative scheme for calculating the first column in Z with similar arguments for the convergence.

4.1.2   Approximating the Diagonal of Z

Now we only have to find a way to approximate z_{jj} for $j \neq t$. In order to do this we will keep track of a *single* random surfer. Each time the surfer decides *not* to follow a link the surfer changes identity and continues surfing from a new page – we chose the new page to start from by adding 1 (cyclically) to the previous start page. For each page p we record the identity of the surfer who made the most recent visit, the total number of visits to p and the number of different surfers who have visited p. The total number of visits divided by the number of different surfers will most likely be close to z_{pp} if the number of visits is large.

If Z_{pp} denotes the stochastic variable denoting the number of visits on page p for a random surfer starting at page p prior to the first zapping event then we have the following [53]:

$$Var(Z_{pp}) = z_{pp}^2 - z_{pp} = z_{pp}(z_{pp} - 1) \ . \tag{4.4}$$

where $Var(\cdot)$ denotes the variance. Since we will obtain the highest value of z_{pp} if all the nodes pointed to by p had only one link back to p then we have that

$$z_{pp} \leq 1 + \alpha^2 + \alpha^4 + \cdots = \frac{1}{1 - \alpha^2} \ . \tag{4.5}$$

Combining (4.4) and (4.5) we have that $Var(Z_{pp}) = O(1)$ so according to *The Central Limit Theorem* we roughly need a *constant* number of visits per node of the random surfer to achieve a certain level of certainty of our approximation of z_{pp}.

Our main interest is to calculate z_{pp} for pages with high values of π_p – luckily $i\pi_p$ is the expected number of visits to page p if the random surfer visits i pages for large i [53] so our approximation of z_{pp} tends to be more precise for pages with high values of π_p. We also note that it is easy to parallelize the algorithm described above simply by tracking several random surfers in parallel.

4.1.3 Experiments

Experiments with the algorithm were carried out on artificial computer generated graphs and on a crawl of the Danish part of the web graph. Running the algorithm on a subgraph of the web graph might seem to be a bad idea but if the subgraph is a community it actually makes sense as suggested by the discussion in Section 1.1.2. In this case we are trying to find optimal link modifications only involving our direct competitors. Locating the community in question by cutting away irrelevant nodes seems to be a reasonable preprocessing step for the algorithm.

Experiments on Artificial Graphs

The algorithm was tested on 10 computer generated graphs each with 500 nodes numbered from 1 to 500 and 5000 links with multiplicity 1 inserted totally at random. For each graph $G(V, E)$ and for each $v \in V$ such that $(v, 1) \notin E$ we computed $\tilde{\pi}_1(\{(v, 1)\})$. The new PageRank value $\tilde{\pi}_1$ of node 1 was computed in two ways: 1) by the algorithm described in this section and 2) by the power method. We used 50 terms when calculating the rows and columns of the Z-matrix and 50 moves per edge for the random surfer when calculating the diagonal of Z. For the PageRank power method computation we used 50 iterations. For all graphs and all v the *relative difference* of the two values of $\tilde{\pi}_1$ was less than 0.1%.

Experiments on the Web Graph

Experiments were also carried out on a crawl from Spring 2005 of the Danish part of the web graph with approximately 9.2 million pages and 160 millions links. For each page v in the crawl we used the algorithm to compute the new PageRank value for www.daimi.au.dk – the homepage of the Department of Computer Science at Aarhus University, Denmark – obtained after adding a link from v to www.daimi.au.dk. The list of potential new PageRank values was sorted in decreasing order.

The PageRank vector and the row and column of Z corresponding to www.daimi.au.dk was calculated using 50 iterations/terms and the diagonal of Z was computed using 300 moves of the random surfer per edge. The computation took a few hours on standard PC's using no effort on optimization.

The links were stored on a file that was read for each iteration/term in the computation of the PageRank vector and the rows and columns of Z.

As can be seen from Equation (4.1) then the diagonal element of Z plays an important role for a potential source with a low out degree. As an example we will look at the pages www.kmdkv.dk/kdk.htm and news.sunsite.dk which we will denote as page a and b respectively in the following. The pages a and b are ranked 22 and 23 respectively in the crawl with π_a only approximately 3.5% bigger than π_b. Page a has out degree 2 and page b has out degree 1 so based on the information on π_a, π_b and the out degrees it would seem reasonable for www.daimi.au.dk to go for a link from page b because of the difference on the out degrees. The results from the experiment show that it is a better idea to go for a link from page a: If we obtain a link to www.daimi.au.dk from page a we will achieve a PageRank value approximately 32% bigger than if we obtain a link from page b. The reason is that z_{bb} is relatively big producing a relatively big denominator in the fraction in (4.1).

4.2 LINK BUILDING $\in$ APX

4.2.1 Ideal Sets of New Backlinks

We will now briefly sketch how Theorem 2.1 can be used to characterize an ideal set of sources for new links to t under the assumption that the minimum out degree d in the graph is sufficiently big. More work has to be done to analyze the general case. We will use the notation from Theorem 2.1 where the matrix $M = \{m_{ij}\}$ is defined and we will also refer to t as node 1. Let $D = \{d_{ij}\}$ be a matrix with $d_{ii} = m_{ii}$ and $d_{ij} = 0$ if $i \neq j$ and now let B be a matrix such that $b_{ii} = 0$ and $b_{ij} = \frac{m_{ij}}{m_{jj}}$ if $i \neq j$. The matrices are constructed such that $M = (I + B)D$. It is not hard to show that $b_{ij} = O(d^{-2})$, so for d sufficiently big we have that $(I + B)^{-1} \approx I - B$ and thus we have the following:

$$\tilde{\pi}_1 - \pi_1 = \begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} M^{-1} q \approx \begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} D^{-1}(I - B)q \tag{4.6}$$

Negative entries can only appear in $I - B$ in (4.6) so the entries are relatively high in $\begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} D^{-1}$, $I - B$ and q for an ideal set S of sources:

1. Any node u in S satisfies at least one of the following two conditions:

    (a) u is relatively popular compared to its out degree or

    (b) u has a low out degree and is within a short distance *from* t (z_{tu} is big)

2. The nodes belong to different communities (z_{uv} is small for $u, v \in S$)

3. The distances from the nodes *to* t are long (z_{ut} is small for $u \in S$)

The entries in $\begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} D^{-1}$ are high if 1a is satisfied. High entries in $I - B$ are assured by 1b and 2 and high entries in q are assured by 3. It is tempting to focus on 1a but it is important to notice that a node satisfying 1b

```
Naive(G, t, k)
    S := ∅
    forall u ∈ V \ {t}  do
        y := π̃_t({(u,t)})
        S := S ∪ {(u,y)}
    Sort S on the y-values in descending order
    Report the first coordinates of the first k elements in S as the solution
```

Figure 4.1: Pseudo code for a naive approach.

will have a corresponding column in $I - B$ amplifying the contribution to $\tilde{\pi}_1$ of all the other nodes. If t only links to a sink s then we might achieve a significant increase in $\tilde{\pi}_t$ by adding the reverse link (s, t) as we saw in Section 2.4 – so greedily picking only nodes satisfying 1a is not always wise.

4.2.2   Analysis of a Naive Approach

We will now analyze the algorithm naively assuming *additivity* for the process of adding backlinks from the nodes in S to t. The underlying false assumption is that the left hand sides of (4.7) and (4.8) below are identical – on the right hand sides of (4.7) and (4.8) we are using the terms from Theorem 2.1:

$$\sum_{u \in S} (\tilde{\pi}_t(\{(u,t)\}) - \pi_t) = \begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} D^{-1} q \tag{4.7}$$

$$\tilde{\pi}_t(S \times \{t\}) - \pi_t = \begin{bmatrix} \pi_2 & \pi_3 & \dots & \pi_{k+1} \end{bmatrix} D^{-1} (I + B)^{-1} q \tag{4.8}$$

The naive algorithm shown in Figure 4.1 picks the k u-nodes with maximum values of $\tilde{\pi}_t(\{(u,t)\})$. The interesting question is how much the topology expressed by B rocks the boat. The first thing we can observe is that the assumption of additivity is OK if the minimum out-degree of the nodes is big in which case we have that $B \approx 0$. We will also have $B \approx 0$ if $z_{tu} \approx 0$ for $u \in S$ and $z_{uv} \approx 0$ for $u, v \in S$ with $u \neq v$. We will restrict our analysis to sets of nodes satisfying $Bq = \lambda q$ for some $\lambda \in \mathbb{R}$. If this is the case then we have that $(I + B)^{-1} q \approx \frac{1}{1+\lambda} q$ and thus we have the following:

$$\frac{1}{1 + \lambda} \sum_{u \in S} (\tilde{\pi}_t(\{(u,t)\}) - \pi_t) = \tilde{\pi}_t(S \times \{t\}) - \pi_t \tag{4.9}$$

The strategy for the analysis is to construct a graph with two sets of nodes with almost similar values of $\sum_{u \in S} \tilde{\pi}_t(\{(u,t)\})$ but with extreme values of λ. In the following we will let π_t^N denote the value of π_t obtained by the naive approach and let π_t^* denote the optimal value. We will show how to construct a directed graph with $\pi_t^* \approx 13.8\pi_t^N$ for $\alpha = 0.85$ showing that the naive approach is indeed naive. For the analysis of the construction we will write $a \approx b$ if we for any $\epsilon > 0$ can construct the graph such that $|a - b| < \epsilon$. A part of our graph is shown in Figure 4.2. The graph is parameterized by k and it contains k grey
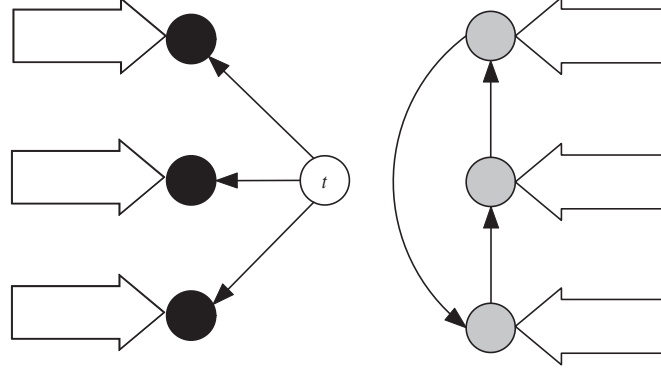
Figure 4.2: A directed graph where t will gain a lot more by obtaining links from the black nodes compared to obtaining links from the grey nodes (assuming that all the grey nodes and black nodes are solutions for the simple case $k = 1$)

nodes with out degree 1 linking to each other in a cycle and k black sinks. The node t has out degree k and links to all the sinks. The big arrows symbolize that the k nodes in the cycle and the k sinks have other nodes linking to them such that $\tilde{\pi}_t(\{(u,t)\})$ is slightly smaller for the sinks compared to the cycle nodes. We can make $\tilde{\pi}_t(\{(u,t)\})$ for the sinks come arbitrarily close to $\tilde{\pi}_t(\{(u,t)\})$ for the cycle nodes by adjusting the number of nodes linking to the sinks and cycle nodes respectively. We will also add a very big strongly connected component to our graph that is isolated from the part of the graph shown in Figure 4.2 with nodes with a small value of $\tilde{\pi}_t(\{(u,t)\})$.

The naive algorithm chooses the k grey cycle nodes and the graph is constructed such that the B-matrix corresponding to the grey nodes has relatively big entries. A major reason for the popularity of the grey nodes is the cycle – and this cycle is suffering a lot of damage if all the grey nodes decide to link to t. It is also worth noting that z_{tt} will only increase marginally if the grey nodes link to t. For a grey node u we have the following:

- $z_{tu} \approx 0$ due to the big isolated strongly connected component.

- $z_{ut} = 0$

- $z_{uu} = \frac{1}{1-\alpha^k}$

- If S denotes the grey nodes then $\sum_{j \in S \setminus \{u\}} z_{uj} = \frac{1}{1-\alpha} - z_{uu} = \frac{1}{1-\alpha} - \frac{1}{1-\alpha^k}$

- $q = \alpha z_{tt} e$ where e is a column vector with all 1'es

We now consult the definition of B from Section 4.2.1 and Theorem 2.1 and get the following:

$$Bq \approx \left(\frac{1}{1-\alpha} - \frac{1}{1-\alpha^k}\right)\frac{1}{1 + \frac{1}{1-\alpha^k}}q = \left(\frac{1-\alpha^k}{1-\alpha} - 1\right)\frac{1}{2-\alpha^k}q$$

Setting $\lambda_1 = \left(\frac{1-\alpha^k}{1-\alpha} - 1\right)\frac{1}{2-\alpha^k}$ we have the following:

$$(I + B)^{-1}q \approx \frac{1}{1+\lambda_1}q$$

The optimal solution is the k black sinks with negative entries with a relatively big absolute value in the B-matrix. If the black nodes decide to link to t they will all benefit from the change of the link structure. For a black node u we have the following – in all cases due to the big isolated strongly connected component:

- $z_{tu} \approx \frac{\alpha}{k}$

- $z_{ut} \approx 0$

- $z_{uu} \approx 1$

- $z_{uv} \approx 0$ for a black node $v \neq u$

- $q \approx \alpha z_{tt} e$ where e is a column vector with all 1's

In this case we have:

$$Bq \approx \frac{1}{1 - \frac{\alpha^2}{k}} \cdot (-\alpha^2) \frac{k-1}{k} q = -\alpha^2 \frac{k-1}{k-\alpha^2} q$$

Setting $\lambda_2 = -\alpha^2 \frac{k-1}{k-\alpha^2}$ we have the following:

$$(I + B)^{-1} q \approx \frac{1}{1 + \lambda_2} q$$

If we assume that π_t is much smaller compared to π_t^N (we can make the ratio $\frac{\pi_t}{\pi_t^N}$ arbitrarily small) then we use (4.9) and get the following:

$$\frac{\pi_t^*}{\pi_t^N} \approx \frac{\pi_t^* - \pi_t}{\pi_t^N - \pi_t} \approx \frac{1 + \lambda_1}{1 + \lambda_2}$$

The ratio is 3.83 for $k = 2$, 8.45 for $k = 5$, 11.42 for $k = 10$ and the limit as k tends to infinity $\frac{1}{1-\alpha^2} \frac{2-\alpha}{2-2\alpha}$ is 13.81 (for $\alpha = 0.85$).

4.2.3   Proof of APX Membership

Now consider the algorithm consisting of k steps where we at each step adds a backlink to t producing the maximum increase in $\frac{\pi_t}{z_{tt}}$ – the pseudo code of the algorithm is shown in Figure 4.3. This algorithm is a polynomial time algorithm producing a solution to the unweighted Link Building problem with a corresponding value within a constant factor from the optimal value as stated by the following theorem so the unweighted variant of LINK BUILDING is a member of the complexity class APX.

Theorem 4.1 *If we let π_t^G and z_{tt}^G denote the values obtained by the greedy algorithm in Figure 4.3 for the unweighted case of LINK BUILDING with optimal value π_t^* then we have the following*

$$\pi_t^G \geq \pi_t^* \frac{z_{tt}^G}{z_{tt}^*} (1 - \frac{1}{e}) \geq \pi_t^* (1 - \alpha^2)(1 - \frac{1}{e})$$

where $e = 2.71828\ldots$ and z_{tt}^* is the value of z_{tt} corresponding to π_t^*.

Greedy(G, t, k)
    $S := \emptyset$
    repeat k **times**
        Let u be a node with maximum value of $\frac{\pi_t}{z_{tt}}$ in $G(V, E \cup \{(u, t)\})$
        $S := S \cup \{u\}$
        $E := E \cup \{(u, t)\}$
    Report S as the solution

Figure 4.3: Pseudo code for the greedy approach.

Proof. Proposition 2.1 in [4] by Avrachenkov and Litvak states the following

$$\pi_t = \frac{1 - \alpha}{n} z_{tt}(1 + \sum_{i \neq t} r_{it}) \ , \tag{4.10}$$

where r_{it} is the probability that a random surfer starting at i reaches t before zapping. This means that the algorithm in Figure 4.3 greedily adds backlinks to t in an attempt to maximize the probability of reaching node t before zapping for a surfer dropped at a node chosen uniformly at random. We show in Lemma 4.1 below that r_{it} in the graph obtained by adding links from $X \subseteq V$ to t is a *submodular* function of X – informally this means that adding the link (u, t) early in the process produces a higher increase of r_{it} compared to adding the link later. We also show in Lemma 4.2 below that r_{it} is not decreasing after adding (u, t) which is intuitively clear. We now conclude from (4.10) that $\frac{\pi_t}{z_{tt}}$ is a submodular and nondecreasing function since $\frac{\pi_t}{z_{tt}}$ is a sum of submodular and nondecreasing terms.

When we greedily maximize a nonnegative nondecreasing submodular function we will always obtain a solution within a fraction $1 - \frac{1}{e}$ from the optimal according to [65] by Nemhauser *et al.*. We now have that:

$$\frac{\pi_t^G}{z_{tt}^G} \geq \frac{\pi_t^*}{z_{tt}^*}(1 - \frac{1}{e}) \ .$$

Finally, we use that z_{tt}^G and z_{tt}^* are numbers between 1 and $\frac{1}{1-\alpha^2}$. $\qquad\square$

For $\alpha = 0.85$ this gives an upper bound of $\frac{\pi_t^*}{\pi_t^G}$ of approximately 5.7 which is much better compared to the performance of the naive approach on the graph from Section 4.2.2. *It must be stressed that this upper bound is considerably smaller if z_{tt} is close to the optimal value prior to the modification – if z_{tt} can not be improved then the upper bound is $\frac{e}{e-1} = 1.58$. It* may *be the case that we obtain a bigger value of π_t by greedily maximizing π_t instead of $\frac{\pi_t}{z_{tt}}$ but $\tilde{\pi}_t(X \times \{t\})$ is* not *a submodular function of X so we can not use the approach above to analyze this situation. To see that $\tilde{\pi}_t(X \times \{t\})$ is not submodular we just have to observe that adding the link $(2, 1)$ from Figure 2.2 late in the process will produce a higher increase in π_1 compared to adding the link early in the process.*

Proof of Submodularity and Monotonicity of r_{it}

Let $f_i(X)$ denote the value of r_{it} in $G(V, E \cup (X \times \{t\}))$ – the graph obtained after adding links from all nodes in X to t.

Lemma 4.1 *f_i is submodular for every $i \in V$.*

Proof. Let $f_i^r(X)$ denote the probability of reaching t from i without zapping in r steps or less in $G(V, E \cup (X \times \{t\}))$. We will show by induction in r that f_i^r is submodular. We shall show the following for arbitrary $A \subset B$ and $x \notin B$:

$$f_i^r(B \cup \{x\}) - f_i^r(B) \leq f_i^r(A \cup \{x\}) - f_i^r(A) \qquad (4.11)$$

- Induction basis $r = 1$. It is not hard to show that the two sides of (4.11) are equal for $r = 1$.

- Induction step. If you want to reach t in $r + 1$ steps or less you have to follow one of the links to your neighbors and reach t in r steps or less from the neighbor:

$$f_i^{r+1}(X) = \frac{\alpha}{outdeg(i)} \sum_{j:i \to j} f_j^r(X) \qquad (4.12)$$

  where $j : i \to j$ denotes the nodes that i links to – this set includes t if $i \in X$. The out degree of i is also dependent on X. If i is a sink in $G(V, E \cup (X \times \{t\}))$ then we can use (4.12) with $outdeg(i) = n$ and $j : i \to j = V$ – as explained in Section 2.1 then the sinks can be thought of as linking to all nodes in the graph. Please also note that $f_t^r(X) = 1$.

We will now show that the following holds for every $i \in V$ assuming that (4.11) holds for every $i \in V$:

$$f_i^{r+1}(B \cup \{x\}) - f_i^{r+1}(B) \leq f_i^{r+1}(A \cup \{x\}) - f_i^{r+1}(A) \qquad (4.13)$$

  - $i \in A$: The set $j : i \to j$ and $outdeg(i)$ are the same for all four terms in (4.13). We use (4.12) and the induction hypothesis to see that (4.13) holds.

  - $i \in B \setminus A$:
    * i is a sink in $G(V, E)$: The left hand side of (4.13) is 0 while the right hand side is positive or 0 according to Lemma 4.2 below.
    * i is not a sink in $G(V, E)$: In this case $j : i \to j$ includes t on the left hand side of (4.13) but not on the right hand side – the only difference between the two sets – and $outdeg(i)$ is one bigger on the left hand side. We now use (4.12), the induction hypothesis and $\forall X : f_t^r(X) = 1$.

  - $i = x$: We rearrange (4.13) such that the two terms including x are the only terms on the left hand side. We now use the same approach as for the case $i \in B \setminus A$.

  - $i \in V \setminus (B \cup \{x\})$: As the case $i \in A$.

Finally, we use $\lim_{r \to \infty} f_i^r(X) = f_i(X)$ to prove that (4.11) holds for f_i. $\square$

Lemma 4.2 f_i *is nondecreasing for every* $i \in V$.

Proof. We shall prove the following by induction in r for $x \notin B$:

$$f_i^r(B \cup \{x\}) \geq f_i^r(B) \tag{4.14}$$

- Induction basis $r = 1$.

  - $i = x$: The left hand side is $\frac{\alpha}{outdeg(x)}$ where $outdeg(x)$ is the new out degree of x and the right hand side is at most $\frac{\alpha}{n}$ (if x is a sink in $G(V, E)$).
  - $i \neq x$: The two sides are the same.

- Induction step. Now assume that (4.14) holds for r and all $i \in V$. We will show that the following holds:

$$f_i^{r+1}(B \cup \{x\}) \geq f_i^{r+1}(B) \tag{4.15}$$

  - $i = x$:
    * i is a sink in $G(V, E)$: The left hand side of (4.15) is α and the right hand side is smaller than α.
    * i is not a sink in $G(V, E)$: We use (4.12) in (4.15) and obtain simple averages on both sides with bigger numbers on the left hand side due to the induction hypothesis.
  - $i \neq x$: Again we can obtain averages where the numbers are bigger on the left hand side due to the induction hypothesis.

Once again we use $\lim_{r \to \infty} f_i^r(X) = f_i(X)$ to conclude that (4.14) holds for f_i. $\square$

4.3 MILP for Link Building

In this section we will show how to formulate the Link Building problem as a Mixed Integer Linear Program (MILP). Actually, we will construct a MILP solving the following more general problem:

Definition 4.1 *The* MARKOV CHAIN MODIFICATION *problem:*

- *Instance: A quadruple* (P, P', C, k) *where* $P = \{p_{ij}\}$ *and* $P' = \{p'_{ij}\}$ *are* $n \times n$ *transition probability matrices for Markov chains,* $C \subset \{1 \ldots n\}$ *and* $k \in \mathbb{Z}^+$. *We assume that we obtain a matrix with a unique stationary probability distribution if we replace any set of k rows from P with indices in C with the corresponding rows in P'.*

Maximize π_1 subject to
1. $\forall j \in V : \pi_j = \sum_{i \in V} \pi_i p_{ij} + \sum_{i \in C} \pi_i x_i (p'_{ij} - p_{ij})$
2. $\sum_{i \in V} \pi_i = 1$
3. $\sum_{i \in C} x_i = k$

Figure 4.4: A Quadratic program for Link Building.

Maximize π_1 subject to
1. $\forall j \in V : \pi_j = \sum_{i \in V} \pi_i p_{ij} + \sum_{i \in C} z_i (p'_{ij} - p_{ij})$
2. $\sum_{i \in V} \pi_i = 1$
3. $\sum_{i \in C} x_i = k$
4. $\forall i \in C : z_i \leq x_i$
5. $\forall i \in C : z_i \leq \pi_i$
6. $\forall i \in C : z_i \geq \pi_i + x_i - 1$

Figure 4.5: A MILP for Link Building.

- *Solution: A set $S \subseteq C$ with $|S| = k$ such that $\tilde{\pi}_1$ is maximized where $\tilde{\pi}_1$ is the first element in the stationary probability distribution for the matrix obtained by replacing the rows in P specified by S with the corresponding rows in P'.*

For the Markov Chain Modification Problem we have an alternative set of transition probabilities for each state and we are allowed to change the transition probabilities for k states in the *candidate set* C. The objective is to maximize a given element in the stationary probability distribution. The LINK BUILDING problem from Definition 1.1 is a special case of this problem – also in the unweighted case.

4.3.1   MILP Specification

It is straightforward to formulate the MARKOV CHAIN MODIFICATION problem as the quadratic program shown in Figure 4.4 using the following variables:

- A binary variable $x_i \in \{0, 1\}$ for every node $i \in C$: $x_i = 1 \Leftrightarrow i \in S$.

- A variable π_i for every node $i \in V$: $\pi_i \geq 0$. The i'th element in the stationary probability distribution.

We now transform the quadratic program into the linear program shown in Figure 4.5 by introducing a new variable $z_i \geq 0$ for $i \in C$ replacing the quadratic term $\pi_i x_i$.

4.3.2   MILP Experiments

We have conducted preliminary experiments solving the LINK BUILDING problem using our linear program. We have solved the problem for varying
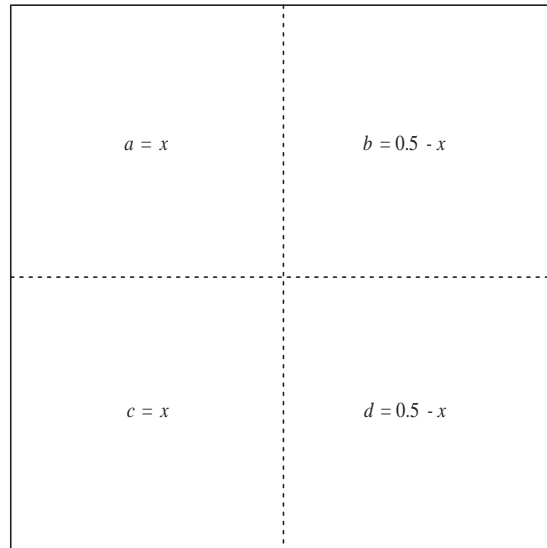
Figure 4.6: Our variant of the R-MAT algorithm recursively puts an entry in the sub-matrices of the adjacency matrix with the probabilities shown above. The distribution of PageRank values gets more "skew" as $x \in [0.25; 0.5)$ increases. If $x = 0.25$ then the entry is placed uniformly at random.

n, m and k on synthetic graphs generated by the R-MAT algorithm introduced by Chakrabarti *et al.* [18]. The number of nodes in a graph constructed by the R-MAT algorithm is a power of 2 and the construction is controlled by the parameters $a, b, c, d \geq 0$ with $a + b + c + d = 1$. The construction process starts with a graph with no edges such that the adjacency matrix[1] for the graph contains all 0'es. Now we randomly choose one of the four sub-matrices of the adjacency matrix shown in Figure 4.6 with probabilities a, b, c and d respectively. The chosen matrix is divided into four new matrices and the process is repeated recursively until we reach a simple cell in which case we place a 1 in the cell. This process is repeated once per link.

Directly citing Chakrabarti *et al.* they "... illustrate experimentally that several, diverse real graphs can be well approximated by an R-MAT model with the appropriate choice of parameters" [18]. We choose parameters $a = x$, $b = 0.5 - x$, $c = x$ and $d = 0.5 - x$ for $x \in [0.25; 0.5)$. By varying x we are now able to adjust the structure of the graph and examine how the structure affects the run time of the linear program. If x increases then the nodes in the left half of the matrices will obtain a higher probability of other nodes linking to them and the "skewness" of the PageRank distribution will increase.

All experiments are done on an Intel® Core™i7 CPU 920 2.67GHz (quad core) with 6Gb RAM running Linux, using a commercial version of AMPL/CPLEX. All our graphs are unweighted and we use $m = 4n$ links. For each node we set up a link[2] to another node chosen uniformly at random

---

[1]Entry i, j in the adjacency matrix is 1 if $(i, j) \in E$ and 0 otherwise.
[2]The linear program in Figure 4.5 is capable of handling sinks but this was not the case for an earlier and significantly different version of the linear program.

and the remaining $3n$ links are placed using the procedure described above. In all experiments we solve the LINK BUILDING problem for $t = \frac{n}{2}$ where we assume that the nodes are numbered from 1 to n and as usual we use $\alpha = 0.85$. For each data point we generate 5 random graphs[3] and average over their running time. In our first experiment we have $x = 0.45$ and $k = 4$ for all our graphs and we vary $n \in \{128, 256, 512, 1024, 2048, 4096, 8192\}$. The running time of the linear program is shown in Figure 4.7a as a function of n. In our second experiment we keep $x = 0.45$ and $n = 1024$ fixed and runs the program for $k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Figure 4.7b depicts the run time as a function of k. In our third and final experiment we keep $n = 256$ and $k = 4$ fixed and vary $x \in \{0.25, 0.30, 0.35, 0.40, 0.45\}$. The running times in the third experiment varied a lot for the 5 graphs for each x. The graph in Figure 4.7c shows the run time as a function of the R-MAT parameter $a = x$. The linear program seems capable of handling graphs with several thousands nodes for moderate k if the in-degree and thus the PageRank distribution is "skew". It does not seem practically possible to handle graphs that have a random nature. In Section 3.2 we saw how to reduce the REGULAR INDEPENDENT SET problem to the LINK BUILDING problem where all relevant nodes in the LINK BUILDING instances involved had identical PageRank values so maybe the LINK BUILDING problem gets "easier" if we assume a certain level of "skewness" on the distribution of the PageRank values? It should be noted that the PageRank distribution appears to be "skew" for the web graph [8].

4.3.3   Other MILP Variants

We now show how we can change the linear program from Figure 4.5 in order to achieve other objectives than obtaining the maximum value of π_1 which has been the main focus up till now in this dissertation. As an example we will consider the natural problem of matching or beating a specific set of nodes $L \subseteq V$ in the *ranking* induced by the PageRank vector for a minimum price – we assume that every new backlink has a fixed price as we assumed in the final comments in Section 3.2. It is straightforward to change the linear program from Figure 4.5 in order to solve this problem: The objective must now be to minimize the price and constraint 3. must be replaced by $\forall i \in L : \pi_1 \geq \pi_i$.

As an example we will revisit the Hexagon examples from Section 2.4.1 in Figure 4.8 and use our linear program to compute the cheapest set of backlinks for node 5 that would make node 5 rank at least as high as the other nodes in the cycle $L = \{2, 3, 4, 5, 6, 7\}$. We will assume that the price of a link (u, v) is proportional to $\frac{\pi_u}{outdeg(u)+1}$ where π_u as usual denotes the PageRank value of u prior to the modification. It seems reasonable that u estimates the value of the link to be proportional to $\frac{\pi_u}{outdeg(u)+1}$ if u only knows π_u and $outdeg(u)$. Adding the link $(6,5)$ turns out to be the cheapest modification that would make node 5 rank higher than the other nodes in the cycle as shown in Figure 4.8b. This is a "value for money" update since z_{55} will improve considerably and node 7

---

[3]The measure of the running time for $n = 8192$ in our first experiment is only based on one graph.

will be substantially hurt. Figure 4.8c shows the cheapest modification bringing node 5 to the top of the ranking.

Another obvious problem that would be interesting to consider is the problem of achieving the highest improvement in the ranking for a given budget (once again we assume that each link has a fixed price). This problem can also easily be modeled by adjusting the linear program. We just have to add a "budget constraint" and change the objective into "Maximize $\sum_{i \in V} r_i$" where $r_i \in \{0, 1\}$ is a new binary variable that only can be 1 if $\pi_1 \geq \pi_i$. As an example we can add the constraint $\forall i \in V : r_i \leq 1 + \pi_1 - \pi_i$.

4.3.4 Reducing the Size of the MILP Instances

At this point the reader of the dissertation might rightfully be worried about the running time of the linear programs introduced in this section. In Section 1.1.2 we presented arguments for the point of view that obtaining backlinks from related nodes is preferable. As we shall see in the next chapter then it is in some cases possible efficiently to identify members of communities in the web graph – a community is a relatively isolated part of the web graph consisting of related nodes. Langville and Meyer [58] and Chien *et al.* [23] present a method for reducing the size of the Markov Chain dramatically by modeling all states/nodes that are not a member of the community as a *single* state/node. We can now use the reduced Markov Chain to compute an *approximation* of the PageRank values in the community following an update of the link structure of the community. This suggests that it is sensible and practically possible – at least in some cases – to reduce the size of the MILP instances dramatically by focusing on nodes related to node 1. We can maybe reduce the running time even further by reducing the candidate set C for the MARKOV CHAIN MODIFICATION problem so C only contains nodes "satisfying" 1a and 1b from the analysis of ideal sets of backlinks in Section 4.2.1. Another possibility is to let C be the set of nodes that are willing to sell links pointing to node 1 (see Section 1.1.2). More work has to be done in order to analyze this approach.
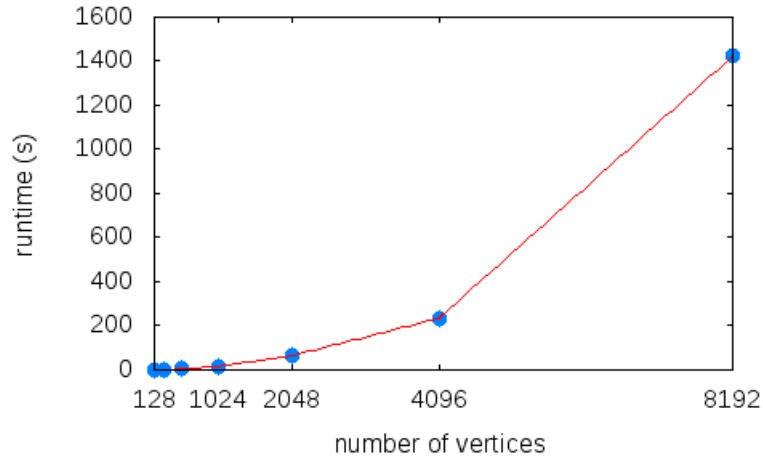
(a) In our first experiment $x = 0.45$ and $k = 4$ are fixed and we vary n.
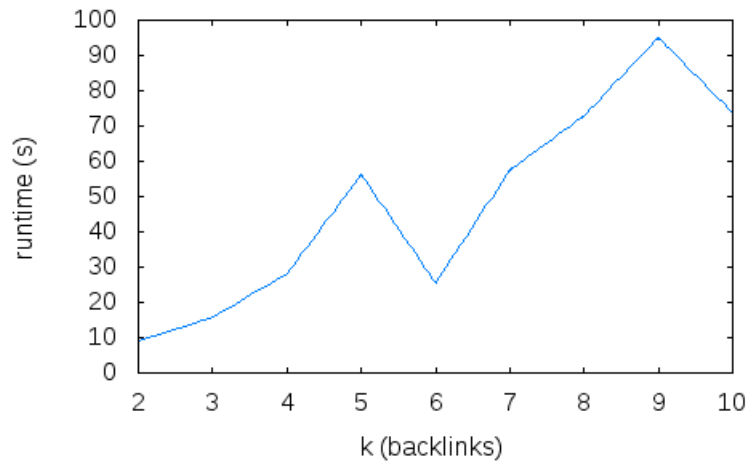
(b) In our second experiment we keep $x = 0.45$ and $n = 1024$ fixed and runs the program for different k.
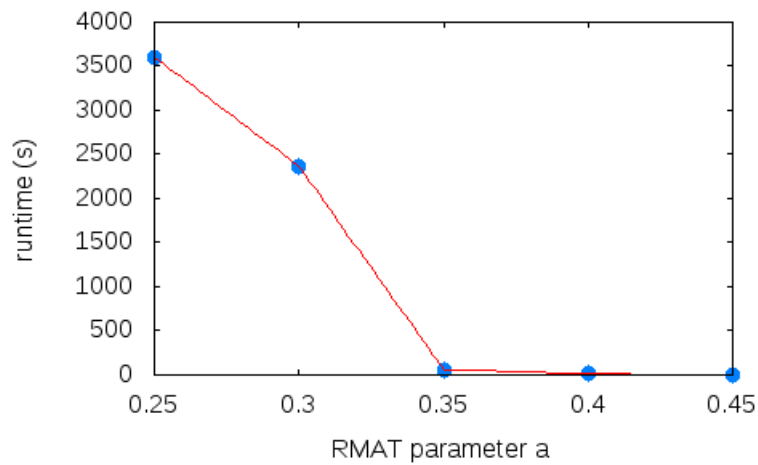
(c) In our third experiment we keep $n = 256$ and $k = 4$ fixed and vary $a = x$.

Figure 4.7: The graphs depict the running time in seconds for our linear program solving the Link Building problem for $t = \frac{n}{2}$. The number of links is $4n$ in all graphs.
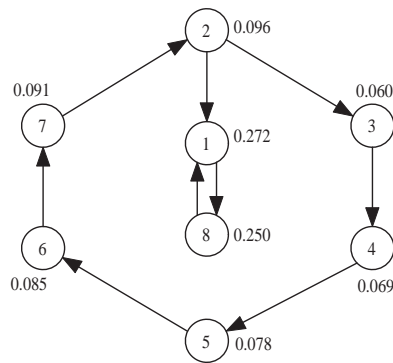
(a) The original graph.
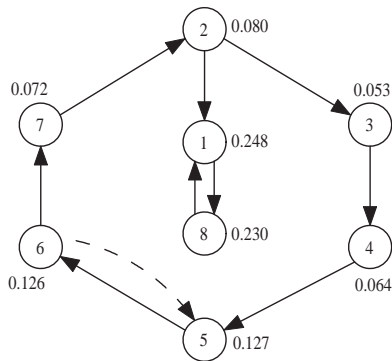
(b) Node 5 matches all nodes in $\{2, 3, 4, 6, 7\}$ in the ranking if the link (6,5) is added to the graph. This is the cheapest way for node 5 to achieve this position in the ranking.
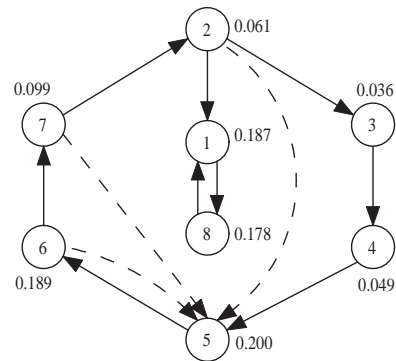
(c) Node 5 tops the ranking if the links (6,5), (7,5) and (2,5) are added to the graph .

Figure 4.8: The PageRank values for the *modified* graphs are shown besides the nodes.

Chapter 5

Detection of Community Members

We now present the details of the contribution of this dissertation with respect to detection of members of communities in networks. These results were published by the author of this dissertation in [70].

As in the preceding chapters we will let $G(V, E)$ denote a directed graph where multiple occurrences of $(u, v) \in E$ is allowed. We will call $(u, v) \in E$ a *link* on u and say that u links to v etc. A link could for example represent a link from site u to site v in the web graph or a reference in a paper written by u to a paper written by v. We define the *relative attention* that u shows v as $w_{uv} = \frac{m(u,v)}{outdeg(u)}$ where $m(u, v)$ is the multiplicity of link (u, v) in E. If $outdeg(u) = 0$ then $w_{uv} = 0$. For $C \subseteq V$ we let $w_{uC} = \sum_{c \in C} w_{uc}$, i.e. the attention that u shows the set of nodes C. In the following we will let $\bar{C}$ denote the complement $V - C$ of C.

We present a community definition justified by a formal analysis of a very simple model of the evolution of a directed graph. We show that the problem of deciding whether a community $C \neq V$ exists such that $R \subseteq C$ for a given set of representatives R is NP-complete. Nevertheless, we show that a fast and simple parameter free greedy approach performs well when detecting communities in the Danish part of the web graph. The time complexity of the approach is only dependent on the size of the found community and its immediate surroundings. Our method is "local" as the method in [6] but it does not use breadth first searches. We also show how to use a computationally inexpensive local variant of the PageRank algorithm to rank the members of the communities and compare the ranking with the PageRank values for the total graph.

These are two possible applications of the algorithms presented in this chapter:

- Consider the following scenario: A user interested in Computer Science visits some sites on this subject. A piece of software running in the background finds that the Computer Science sites are similar by analyzing the content of the sites. It uses the Computer Science sites as the set R and reports a community C containing R with the sites ranked by our ranking algorithm. A real world example in Section 5.3.2 documents that this list could be very useful to the user!

- The community found can be used to reduce the size of the MILP instance for the Link Building problem as explained in Section 4.3.4.

In Section 5.1 the community definition and the greedy approach for identifying community members are presented. The ranking algorithm is introduced in Section 5.2 and the experiments are reported in Section 5.3.

5.1   Locating Communities

5.1.1   Community Definition

The intuition behind our community definition is that every community member shows more attention to the community than any non member:

Definition 5.1 *A community is a set $C \subseteq V$ such that*

$$\forall u \in C, \forall v \in \bar{C} : w_{uC} \geq w_{vC} \ .$$

Consider the following process: Assume the existence of a set $C \subset V$ and numbers p_1 and p_2 with $0 \leq p_1 < p_2 \leq 1$ such that the following holds: Every time a node $u \in C$ links to another node it will link to a member in C with probability p_2. Every time a node $v \in \bar{C}$ establishes a link it will link to a member in C with probability p_1. Each member of V establishes exactly q links independently of all other links established.

The number p_2 can be smaller than $\frac{1}{2}$ which means that the members of C does not necessarily predominantly link to other members of C as supposed in [38].

Definition 5.1 is justified by the following theorem:

Theorem 5.1 *Consider the process defined above and let $n = |V|$. If $\gamma = \left(1 - \frac{p_1}{p_2}\right) / \ln \frac{p_2}{p_1}$ then:*

$$P(\forall u \in C, \forall v \in \bar{C} : w_{uC} \geq w_{vC}) \geq 1 - n \left(\frac{e^{\gamma-1}}{\gamma^\gamma}\right)^{p_2 q} \ . \qquad (5.1)$$

Proof. Let X_{xC} denote the number of links established by x linking to members in C. Let $\mu_2 = p_2 \cdot q$ denote the expected value for X_{uC} if $u \in C$. The expected value for X_{vC} for $v \in \bar{C}$ is $\mu_1 = p_1 \cdot q$.

We will establish an upper bound for the probability of the event in (5.1) *not happening*:

$$P(\exists u \in C, \exists v \in \bar{C} : X_{uC} < X_{vC}) \leq$$

$$P(\exists u \in C : X_{uC} < \tau \vee \exists v \in \bar{C} : X_{vC} > \tau) \leq$$

$$|C| \cdot P(X_{uC} < \tau) + |\bar{C}| \cdot P(X_{vC} > \tau) \ . \qquad (5.2)$$

where u and v are generic elements in C and $\bar{C}$ respectively. This upper bound holds for any value of τ. The strategy of the proof is to find a τ such that the factors $P(X_{uC} < \tau)$ and $P(X_{vC} > \tau)$ have a low common upper bound.

We will use two Chernoff bounds and produce upper bounds on the factors in (5.2) assuming $\tau = \gamma\mu_2 = \frac{p_2}{p_1}\gamma\mu_1$ for $\gamma \in (\frac{p_1}{p_2}, 1)$:

$$P(X_{uC} < \gamma\mu_2) \leq e^{-\mu_2} \left(\frac{e^\gamma}{\gamma^\gamma}\right)^{\mu_2} \ . \qquad (5.3)$$

$$P\left(X_{vC} > \frac{p_2}{p_1}\gamma\mu_1\right) \leq e^{-\mu_1} \left(\frac{e}{\frac{p_2}{p_1}\gamma}\right)^{\frac{p_2}{p_1}\gamma\mu_1} = e^{-\mu_1} \left(\frac{p_1}{p_2}\right)^{\gamma\mu_2} \left(\frac{e^\gamma}{\gamma^\gamma}\right)^{\mu_2} \ . \qquad (5.4)$$

Now we will find a necessary and sufficient condition for these upper bounds to be identical:

$$e^{-\mu_2} = e^{-\mu_1} \left(\frac{p_1}{p_2}\right)^{\gamma\mu_2} \Leftrightarrow$$

$$-\mu_2 = -\mu_1 + \gamma\mu_2 \ln \frac{p_1}{p_2} \Leftrightarrow$$

$$\gamma = \left(1 - \frac{p_1}{p_2}\right) / \ln \frac{p_2}{p_1} \quad .$$

The upper bounds in (5.3) and (5.4) are identical for this value of γ which is easily shown to satisfy $\gamma \in (\frac{p_1}{p_2}, 1)$. We will put the common value $(\frac{e^{\gamma-1}}{\gamma^\gamma})^{\mu_2}$ in (5.2):

$$P(\exists u \in C, \exists v \in \bar{C} : X_{uC} < X_{vC}) \leq n \left(\frac{e^{\gamma-1}}{\gamma^\gamma}\right)^{p_2 q} \quad .$$

$\square$

Theorem 5.1 shows that real communities with $p_2 > p_1$ probably will obey Definition 5.1 in a large network where the number of links from each node is logarithmically lower bounded as pointed out by the following corollary:

Corollary 5.1 *For fixed p_1 and p_2 with $p_1 < p_2$ there exists a constant $k > 0$ such that*

$$P(\forall u \in C, \forall v \in \bar{C} : w_{uC} \geq w_{vC}) \to 1 \quad for \quad n \to \infty \quad .$$

for $q = k \cdot \log n$.

Before addressing computability issues a couple of remarks on our community definition are in place. First of all there might be several communities containing a given set of representatives so picking the representatives might require several attempts. The experiments in Section 5.3.1 deal with the problem of choosing representatives. Secondly the union $C = C_1 \cup C_2$ of two communities C_1 and C_2 is not necessarily a community. For example there might be a node $v \in \bar{C}$ with $w_{vC} = 1$ and a node $u \in C$ with $w_{uC} < 1$ in which case C would not be a community since $w_{uC} < w_{vC}$. Communities in the "real world" seem to share these properties with our formal communities.

5.1.2 Intractability

We will now formally define the problem of deciding whether a non trivial community exists for a given set of representatives R:

Definition 5.2 *The COMMUNITY problem:*

- *Instance: A directed graph $G(V, E)$ and a set of nodes $R \subset V$.*

- *Question: Does a community $C \neq V$ according to Definition 5.1 exist such that $R \subseteq C$?*

If we had an effective algorithm locating a non trivial community if at least one such community existed then we also could solve COMMUNITY effectively but even solving COMMUNITY effectively seems hard according to the following theorem:

Theorem 5.2 *COMMUNITY is NP-complete.*
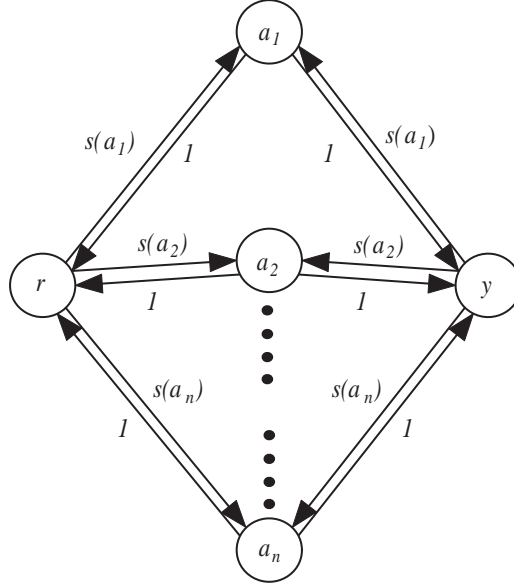
Figure 5.1: A non trivial community C with $r \in C$ exists if and only it is possible to divide the set A in two parts with the same size. Each link is labeled with its multiplicity.

Proof. We can check in polynomial time whether C is a community containing R by calculating w_{xC} for all $x \in V$ thus COMMUNITY is in NP.

We will transform an instance of the NP-complete problem PARTITION [43, page 223] into an equivalent instance of COMMUNITY in polynomial time. This means that we can solve the NP-complete problem PARTITION in polynomial time if we can solve COMMUNITY in polynomial time thus COMMUNITY is NP-complete since it is a member of NP. The rest of the proof contains the details of the transformation.

An instance of PARTITION is a finite set $A = \{a_1, a_2, \ldots, a_n\}$ and a size $s(a_i) \in \mathbb{Z}^+$ for each $a_i \in A$. The question is whether a subset $A' \subset A$ exists such that $\sum_{a \in A'} s(a) = \frac{S}{2}$ where S is the sum of the sizes of all elements in A? We will transform this instance into the instance of COMMUNITY given by a directed graph $G(V, E)$ with $n + 2$ nodes and $R = \{r\}$ where r is one of the nodes in G. The graph G is constructed in the following way:

We will start with two nodes r and y. For each $a_i \in A$ we will make a node with two links (a_i, r) and (a_i, y) with multiplicity 1 and two links (r, a_i) and (y, a_i) with multiplicity $s(a_i)$. The resulting network is shown on Figure 5.1.

Now we will prove that G contains a non trivial community C containing R if and only if A' exists.

- If A' exists then $C = \{r\} \cup A'$ is a non trivial community containing r since $w_{xC} = \frac{1}{2}$ for all $x \in V$.

- Now assume that C is a non trivial community containing r. If C contains y then C also contains all the a's since $w_{aC} = 1$ if $\{r, y\} \subseteq C$. Since C is a non trivial community we have $y \notin C$. Now set $A' = C \cap A$.

- If $\sum_{a \in A'} s(a) < \frac{S}{2}$ then $w_{rC} < \frac{1}{2}$ but there is at least one $a \notin C$ with $w_{aC} = \frac{1}{2}$ contradicting that C is a community.

- If $\sum_{a \in A'} s(a) > \frac{S}{2}$ then $w_{yC} > \frac{1}{2}$ but there is at least one $a \in C$ with $w_{aC} = \frac{1}{2}$ - yet another contradiction.

We can conclude that $\sum_{a \in A'} s(a) = \frac{S}{2}$.

$\square$

The network in Figure 5.1 might be illustrative when comparing the definitions of a community in this chapter and in [39]. If $A' \subset A$ exists such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$ then $C = \{r\} \cup A'$ will not be a community by the definition in [39] for any value of α (see Section 1.4.2).

5.1.3   A Greedy Approach

Despite the computational intractability experiments show that it is possible to find communities in the Danish part of the web graph with a simple greedy approach. We will present the results of the experiments in Section 5.3.

The approach starts with $C = R$. It then moves one element from $\bar{C}$ to C at a time choosing the element $v \in \bar{C}$ with the highest value of w_{vC}. After moving v to C it updates w_{xC} for all x linking to v and checks whether the current C satisfies Definition 5.1. The approach can be effectively implemented using two priority queues containing the elements in C and the elements in $\bar{C}$ linking to C respectively using w_{xC} as the key for x. The C-queue is a min-queue and the $\bar{C}$-queue is a max-queue. It is possible to find the next element to move and to decide if C is a community by inspecting the first elements in the queues as can be seen from the pseudo code of the approach shown in Figure 5.2.

The time complexity of the approach is $O((n_C + m_C) \log n_C)$ where n_c is the number of elements in the union of the found community C and the set of nodes linking to C and m_C is the number of links between elements in C plus the number of links to C from $\bar{C}$ - multiple occurrences of $(u, v) \in E$ only counts as one link. The argument for the complexity is that less than n_C elements have to move between the two queues and that m_C update-priority operations are performed on the two queues containing no more than n_C elements. We are assuming that finding one node x linking to v can be done in constant time.

Some of the representatives might have no links, so we do not consider the attention shown by the representatives to C when we check whether C satisfies our definition of a community for the experiments in this chapter. To be more specific we check whether

$$\forall u \in C - R, \forall v \in \bar{C} : w_{uC} \geq w_{vC} \ .$$

5.2   Ranking the Members

The PageRank algorithm can be viewed as a vote among *all* pages yielding a global measure of popularity. We will turn this into a vote among the *relevant*

Greedy(G, R)
　　C-queue $:= \emptyset$
　　$\bar{C}$-queue $:= \emptyset$
　　forall $r \in R$ **do**
　　　　forall $x \in V - R$ linking to r **do**
　　　　　　if $x \in \bar{C}$-queue **then**
　　　　　　　　increase the priority of x with w_{xr}
　　　　　　else
　　　　　　　　insert x in the $\bar{C}$-queue with priority w_{xr}
　　while $|C$-queue$| <$ minimum size or min(C-queue) $<$ max($\bar{C}$-queue) **do**
　　　　move the element v with maximum priority from the $\bar{C}$-queue to the C-queue
　　　　forall $x \in V - R$ linking to v **do**
　　　　　　if $x \in C$-queue or $x \in \bar{C}$-queue **then**
　　　　　　　　increase the priority of x with w_{xv}
　　　　　　else
　　　　　　　　insert x in the $\bar{C}$-queue with priority w_{xv}
　　Report $R \cup C$-queue as a community

Figure 5.2: Pseudo code for the greedy approach. Details for handling an empty C-queue or an empty $\bar{C}$-queue in the while-loop have been omitted for clarity.

pages that are the pages in C. The experiments carried out produce what we believe to be very valuable rankings which support the validity of the mathematical models behind the rankings. We will adjust the random surfer model explained in Section 2.1 in the following way – the modification is simpler than but similar to the state lumping approach in [58] but the objective is to obtain a ranking strengthening the position of "locally popular" nodes:

A visitor to a community member $i \in C$ is assumed to have the following behavior:

- With probability given by some number α he decides to follow a link from i. As usual we use $\alpha = 0.85$. In this case there are two alternatives:

  - He decides to visit another member j of C. The probability that j gets a visit in this way is $\alpha \cdot w_{ij}$.
  - He follows a link to a non member v. Assuming a low upper bound on w_{vC} it is not likely that the visitor will use a link to go back to C. Thus we treat this case as a jump to another member of C chosen uniformly at random.

- With probability $1 - \alpha$ he decides to jump to another place without following a link which is treated as a jump to a member in C chosen uniformly at random.

A visitor to $i \in C$ will visit $j \in C$ with probability

$$p_{ij} = \frac{1 - \alpha}{|C|} + \frac{\alpha(1 - w_{iC})}{|C|} + \alpha \cdot w_{ij} = \frac{1 - \alpha \cdot w_{iC}}{|C|} + \alpha \cdot w_{ij} \ .$$

Like PageRank the ranking of the members is based on the unique stationary probability distribution of the Markov chain given by the transition matrix $P = \{p_{ij}\}_{i,j \in C}$. An iterative calculation of $w^T \cdot P^s$ will converge to the stationary distribution in a few iterations where w is an arbitrary initial probability distribution. For details on convergence rates etc. we refer to the work of Langville and Meyer [56].

5.3 Experimental Work

For an online version of the results of the experiments please visit the home page of the author: www.cs.au.dk/~mo/. Besides the results reported in this chapter you can also find results from experiments with the *s-t* minimum cut approach from [39].

5.3.1 Identification of Community Members in Artificial Graphs

Inspired by Newman *et al.* [67] we test the greedy approach on some random computer generated graphs with known community structure. The graphs contain 128 nodes divided into four groups with 32 nodes each with nodes 1 - 32 in the first group, 33 - 64 in the next group etc. We will denote the first of the four groups as *group 1*. For each pair of nodes u and v either two links - (u,v) and (v,u) - or none are added to the graph. The pairs of links are placed independently at random such that the *expected* number of links from a node to nodes in the same group is 9 and the expected number of links to nodes outside the group is 7.

For 10 graphs the greedy approach reported the first community found containing at least 32 members with node number 1 as the single representative. The average size of the community found was 64.3 and the average number of nodes from group 1 in the community found was 28.9. If we use nodes 1 to 5 as representatives instead the corresponding numbers are 39.3 and 31.3 and if we use nodes 1 to 10 as representatives the numbers are 32.4 and 31.2. These admittedly few experiments suggest that the greedy approach can actually identify members of communities if the number of representatives is sufficient.

5.3.2 Identification and Ranking of Danish Computer Science Sites

Now we will demonstrate that the greedy approach is able to identify communities in the web graph using only a few representatives. A crawl of the Danish part of the web graph from April 2005 was used as the basis for the web experiments. In the first experiment conducted on the crawl V consists of the 180468 *sites* in the crawl where a link from site u to v is represented by $(u,v) \in E$.

The objective of the experiment was to identify and rank Danish Computer Science sites. The following four sites were used as representatives:

- **www.itu.dk**, IT University of Copenhagen

Table 5.1: The Top 20 of two communities of Danish Computer Science sites. Representatives are written with bold font. The numbers after a site is the "global" ranking in the dk domain.

	556 members	1460 members
1	www.daimi.au.dk 267	www.au.dk 109
2	www.diku.dk 655	www.sdu.dk 108
3	**www.itu.dk** 918	www.daimi.au.dk 267
4	**www.cs.auc.dk** 1022	www.hum.au.dk 221
5	www.brics.dk 1132	www.diku.dk 655
6	**www.imm.dtu.dk** 1124	www.ifa.au.dk 681
7	www.dina.kvl.dk 1153	**www.itu.dk** 918
8	www.agrsci.dk 1219	www.ruc.dk 945
9	www.foejo.dk 1504	www.phys.au.dk 1051
10	www.darcof.dk 2113	www.brics.dk 1132
11	www.it-c.dk 2313	**www.cs.auc.dk** 1022
12	www.dina.dk 2169	www.dina.kvl.dk 1153
13	www.cs.aau.dk 2010	**www.imm.dtu.dk** 1124
14	rapwap.razor.dk 4585	www.agrsci.dk 1219
15	imv.au.dk 2121	www.kvinfo.dk 1122
16	razor.dk 2990	www.foejo.dk 1504
17	**www.imada.sdu.dk** 2998	www.bsd-dk.dk 1895
18	www.plbio.kvl.dk 3543	www.humaniora.sdu.dk 1826
19	www.math.ku.dk 2634	www.imv.au.dk 2121
20	mahjong.dk 3813	www.statsbiblioteket.dk 867

- **www.cs.auc.dk**, Department of Computer Science, University of Aalborg

- **www.imm.dtu.dk**, Department of Informatics and Mathematical Modeling, Technical University of Denmark

- **www.imada.sdu.dk**, Department of Mathematics and Computer Science, University of Southern Denmark

The sites of the Departments of Computer Science for the two biggest universities in Denmark, **www.diku.dk** and **www.daimi.au.dk**, were *not included* in the set of representatives. These sites represent the universities in Copenhagen and Aarhus respectively.

The greedy approach found several communities. The Top 20 ranking of two communities with 556 and 1460 sites respectively are shown in Table 5.1 which also shows the ranking produced by a PageRank calculation on the dk domain. Members of both communities use more than 15-16 % of their links to other members and non members use less than 15-16 % on members.

The Top 20 lists contain mainly academic sites and the smaller community seems to be dominated by sites related to Computer Science. The ranking seems to reflect the "sizes" of the corresponding real world entities. It is worth

noting that **www.daimi.au.dk** and **www.diku.dk** are ranked 1 and 2 in the smaller community. The site ranked 5 in the smaller community represents BRICS, Basic Research in Computer Science, which is an international PhD school within the areas of computer and information sciences, hosted by the Universities of Aarhus and Aalborg.

The larger community seems to be a more general academic community with the sites for University of Aarhus and University of Southern Denmark ranked 1 and 2 respectively. The larger community obviously contains the smaller community by the nature of the greedy approach.

The local ranking seems to reflect the global ranking with a few exceptions. The site rapwap.razor.dk is popular among the relevant sites but seems not to be that popular overall. The person behind rapwap.razor.dk has pages in Top 5 on Google searches[1] for Danish pages on "cygwin" and "php" which justifies rapwap.razor.dk's place on the Top 20 list of Danish Computer Science sites.

5.3.3   Identification and Ranking of Danish Chess Pages

We also carried out an experiment at the *page level* in order to rank Danish Chess pages using *one representative only*: www.dsu.dk, the homepage for the Danish Chess Federation. For this experiment V consisted of all pages up to three inter site links away from the representative where the links were considered unoriented. V contains approximately 330.000 pages. The weight w_{uv} is the fraction of inter site links on page u linking to page v.

The greedy approach located a community with 471 members. All members use at least 1.4% of their inter site links on members and non members use less than 1.4% on members. This means that only heavily linked non members link to the pages in the community and if they do they only link to the community with a few links. The Top 20 for this experiment – using the ranking from Section 5.2 – is shown in Table 5.2.

The page ranked 2 in the Top 20 is a page for a chess tournament held in Denmark in 2003 with several grandmasters competing. The pages ranked 13 and 20 are pages (at that time) for the Danish and Scandinavian Chess championships respectively. Several of the subdivisions of the Danish Chess Federation (4, 7, 9, 19) are represented on the Top 20 and the page ranked 6 provides access to a database of more than 40.000 Chess games[2]. Most of the rest of the pages on the Top 20 are Chess Club pages. All in all the Top 20 seems useful from a Danish chess players point of view.

For comparison we searched Google[3] for Danish pages containing the word "skak" – the Danish word for chess. Several of the sites with pages in the Top 20 from Table 5.2 are also present in the Google search result but the latter seems targeted at a broader chess audience. The Google Top 20 contains for example several pages dealing with online chess and chess programs. The Top 20 from Table 5.2 seems to be targeted at a dedicated Danish chess player being a member of a chess club.

---

[1]The searches were carried out on January 23 2007.
[2]Appear to have moved to http://dsu9604.dsu.dk/partier/danbase.htm.
[3]The searches were carried out on April 12 2007.

Table 5.2: The top 20 of a community of 471 Danish chess pages found with the homepage of the Danish Chess Federation as a representative (written with bold font). The Danish word for chess is "skak".

1.	**www.dsu.dk**
2.	www.sis-mh-masters.dk
3.	dsus.dk
4.	www.8-hk.dk
5.	www.dsus.dk
6.	www.dsu.dk/partier/danbase.htm
7.	www.vikingskak.dk/4hk
8.	www.sk1968.dk
9.	www.4hk.dk
10.	www.skovlundeskakklub.dk
11.	www.vikingskak.dk
12.	www.alssundskak.dk
13.	www.skak-dm.dk
14.	www.frederikssundskakklub.dk
15.	www.birkeskak.dk
16.	home13.inet.tele.dk/dianalun
17.	www.rpiil.dk/nvf
18.	www.enpassant.dk/chess/index.html
19.	www.4hk.dk/index.htm
20.	www.skak-nm.dk

Chapter 6

Additively Separable Hedonic Games

This chapter contains the details of the results related to Hedonic Games. The results were published by the author of this dissertation in [69,72] where [72] is a journal version of [69] containing considerations relating the results to community structures in networks.

For a formal introduction to Additively Separable Hedonic Games and the related stability concepts we refer to Section 1.5 which also includes a discussion of related work. In Section 6.1 we provide an example of an Additively Separable Hedonic Game in an attempt to ease the understanding of the formal definitions. In Section 6.2 we show that the problem of deciding whether a Nash stable partition exists in an Additively Separable Hedonic Game is NP-complete. In Section 6.3 we relate the field of detection of community structures to Nash stable partitions in Additively Separable Hedonic Games and argue that community structures in networks can be viewed as Nash stable partitions. This motivates looking at the computational complexity of computing equilibriums in games with symmetric and positive preferences which is the subject of Section 6.4. In this section we show that the problem of deciding whether a *non trivial* Nash stable partition exists in an Additively Separable Hedonic Game with *non-negative* and *symmetric* preferences is NP-complete. This result also applies to individually stable partitions since individually stable partitions are Nash stable and vice versa in such games.

6.1   The buffalo-parasite-game

We will now present an example of an Additively Separable Hedonic Game. We will use biological terminology metaphorically to ease the understanding for the game. The game does *not* represent a serious attempt to model a biological system.

Assume that there are two buffaloes b_1 and b_2 in an area with n waterholes $w_1, w_2, \ldots, w_n$. Each waterhole w_i has a capacity $c(w_i)$ specifying how much water a buffalo can drink from that hole per year. There are also two parasites p_1 and p_2 in the area. The only possible host for p_1 is b_1 and b_1 must drink a lot of water if p_1 is sitting on its back. The same goes for p_2 and b_2. Now assume that b_1 and b_2 are enemies and that a buffalo must drink water corresponding to half the total capacity C of the waterholes if it is the host of a parasite. This system can be viewed as an Additively Separable Hedonic Game depicted as a weighted directed graph in Figure 6.1 where the weight of edge (i, j) is $v_i(j)$ - if there is no edge (i, j) then $v_i(j) = 0$. We have added two edges (b_1, b_2) and (b_2, b_1) with capacity $-C - 1$ to model that b_1 and b_2 are enemies. Please note that the waterholes are also players in the game. The waterholes do not care which coalitions they belong to.

A partition Π of the players is not Nash stable if b_1 is not the host of p_1 – in this case p_1 would be strictly better off by joining $S_\Pi(b_1)$. This fact can be expressed more formally: $S_\Pi(b_1) \cup \{p_1\} \succ_{p_1} S_\Pi(p_1)$ if $S_\Pi(p_1) \neq S_\Pi(b_1)$. In this game a Nash stable partition of the players exists if and only if we can split the waterholes in two groups with the same capacity. We will formally show and use this fact in the next section.
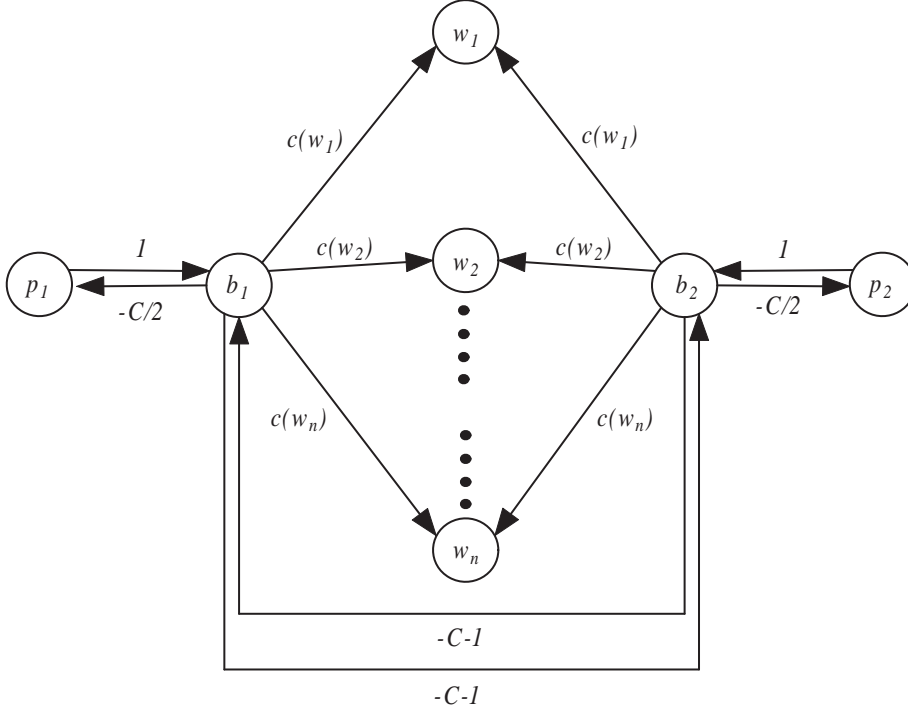
Figure 6.1: An example of an Additively Separable Hedonic Game: The buffalo-parasite-game.

6.2   Restricting to Additively Separable Games

In this section we restrict our attention to Additively Separable Hedonic Games compared to Ballester [7]. Compared to Bogomolnaia and Jackson [51], we also allow asymmetric preferences. Informally we show that things are complicated even when looking at Additively Separable Hedonic Games. With an intuitively clear proof based on the buffalo-parasite-game from Section 6.1 we show that the problem of deciding whether a Nash stable partition exists in a Hedonic Game remains NP-complete when restricting to additively separable preferences. We will now formally define the problem:

Definition 6.1 *The ASH-NASH problem:*

- *Instance: A set $N = \{1, 2, \ldots, n\}$ and a function $v_i : N \to \mathbb{R}$ such that $v_i(i) = 0$ for each $i \in N$.*

- *Question: Does a partition Π of N exist such that*

$$\forall i \in N, \forall S_k \in \Pi \cup \{\emptyset\} : \sum_{j \in S_{\Pi}(i)} v_i(j) \geq \sum_{j \in S_k \cup \{i\}} v_i(j) \ ? \qquad (6.1)$$

We are going to prove that this problem is intractable.

Theorem 6.1 *ASH-NASH is NP-complete.*

Proof. It is easy to check in polynomial time that Π is a partition satisfying (6.1) thus ASH-NASH is in NP.

We will transform an instance of the NP-complete problem PARTITION [43] into an instance of ASH-NASH in polynomial time such that the answers to the questions posed in the two instances are identical - if such a transformation exists we will write PARTITION $\propto$ ASH-NASH following the notation in [43]. This means that we can solve the NP-complete problem PARTITION in polynomial time if we can solve ASH-NASH in polynomial time thus ASH-NASH is NP-complete since it is a member of NP. The rest of the proof explains the details of the transformation.

An instance[1] of PARTITION is a finite set $W = \{w_1, w_2, \ldots, w_n\}$ and a capacity $c(w) \in \mathbb{Z}^+$ for each $w \in W$. The question is whether a subset $W' \subset W$ exists such that $\sum_{w \in W'} c(w) = \frac{C}{2}$ where $C = \sum_{w \in W} c(w)$.

Now suppose we are given an instance of PARTITION. The PARTITION instance is transformed into the buffalo-parasite-game from Section 6.1 in linear time. All we have to do to translate this as an ASH-NASH instance is to perform a simple numbering of the players in the game.

Now we only have to show that a Nash stable partition of the game in Figure 6.1 exists if and only if W' exists. This can be seen from the following argument:

- The partition $\Pi = \{\{b_1, p_1\} \cup W', \{b_2, p_2\} \cup W - W'\}$ is Nash stable if W' exists.

- Now assume that a Nash stable partition Π exists and define $W_1 = S_\Pi(b_1) \cap W$ and $W_2 = S_\Pi(b_2) \cap W$. The set $S_\Pi(b_1)$ must contain p_1. Due to the stability we can conclude that $\sum_{w \in W_1} c(w) \geq \frac{C}{2}$ - otherwise b_1 would be better off by its own. By a symmetric argument we have $\sum_{w \in W_2} c(w) \geq \frac{C}{2}$. The two nodes b_1 and b_2 are not in the same coalition so the two sets W_1 and W_2 are disjoint, so we can conclude that $\sum_{w \in W_1} c(w) = \sum_{w \in W_2} c(w) = \frac{C}{2}$. We can take $W' = W_1$ or $W' = W_2$.

$\square$

6.3 Community Structures as Nash Stable Partitions

In this section we relate community structures in networks and Nash stable partitions in Additively Separable Hedonic Games. It seems natural to *define* a *community structure* of N as a partition Π of N such that for any $C \in \Pi$ we have that all members of C feel more related to the members of C compared to any other set in the partition. This is just a less formal way of stating (1.1) – the property defining a Nash stable partition in a Hedonic Game.

Suppose we are given a set N and a number $v_{ij} \in \mathbb{R}^+ \cup \{0\}$ for each pair of nodes $\{i, j\}$ in N modeling the strength of the connection between i and j. As an example we could be given an undirected and unweighted graph $G(N, E)$ and

---

[1]The objects constituting an instance in [43] are renamed to match the game from Section 6.1

let $v_{ij} = 1$ if $\{i, j\} \in E$ and 0 otherwise. If we adopt the definition above of a community structure then we essentially have an Additively Separable Hedonic Game with *non-negative* and *symmetric* preferences with community structures appearing as Nash stable partitions. That community structures appear in this way seems to be a reasonable assumption based on visual inspection of the communities identified by Newman and Girvan in [67].

If for example the members of N form a clique where all the connections have identical strength then the trivial partition $\Pi = \{N\}$ is the only Nash stable partition. In this case there would not be any non trivial community structure which sounds intuitively reasonable. On the other hand, let us assume that two disjoint communities S and T of players exist as defined in [38] (the definition is presented in Section 1.4.2). If we collapse these communities to two players s and t then we can effectively calculate the s-t minimum cut in the underlying graph for the game. This cut defines a non trivial Nash stable partition. As noted in Section 1.5.3 then a partition of communities following the definition in [38] would certainly be a community structure – but the converse is not always true. The definition of a community structure suggested above can thus be seen as a sort of generalization of the definition of a community in [38].

We will denote a non trivial Nash stable partition as an *inefficient equilibrium* – if the numbers v_{ij} are seen as payoffs then it is optimal for all members of the network to cooperate. In the next section we will prove that inefficient equilibriums generally are hard to compute. To be more specific we will prove that the problem of deciding whether they exist is NP-complete. This result formally indicates that computing community structures is a hard job.

6.4 Non-negative and Symmetric Preferences

As in the proof of Theorem 6.1 we need a known NP-complete problem in the proof of the theorem of this section. The "base" problem of the proof in this section is the *EQUAL CARDINALITY PARTITION* problem:

Definition 6.2 *The EQUAL CARDINALITY PARTITION problem:*

- *Instance: A finite set $W = \{w_1, w_2, \ldots, w_n\}$ and a capacity $c(w) \in \mathbb{Z}^+$ for each $w \in W$*

- *Question: Does a non trivial partition $\{W_1, \ldots, W_k\}$ of W exist such that $|W_i| = |W_j|$ and $\sum_{w \in W_i} c(w) = \sum_{w \in W_j} c(w)$ for all $1 \le i, j \le k$?*

EQUAL CARDINALITY PARTITION is closely related to the balanced version of PARTITION where we are looking for a set $W' \subset W$ such that $\sum_{w \in W'} c(w) = \frac{C}{2}$ and $|W'| = \frac{|W|}{2}$. The balanced version of PARTITION is known to be NP-complete [43]. An instance of the balanced version of PARTITION is transformed into an equivalent instance of EQUAL CARDINALITY PARTITION by adding two more elements to the set W - both with capacity $C + 1$. This shows that EQUAL CARDINALITY PARTITION is NP-complete since it is easily seen to belong to NP.
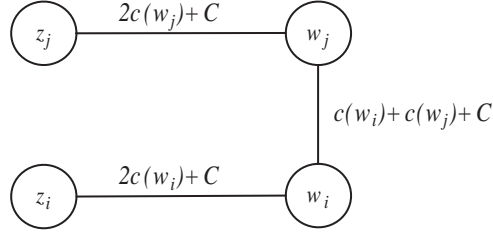
Figure 6.2: A part of a game with positive and symmetric preferences.

We will now formally define the problem of deciding whether a non trivial Nash stable partition exists in an Additively Separable Hedonic Game with non-negative and symmetric preferences:

Definition 6.3 *The INEFFICIENT EQUILIBRIUM problem:*

- *Instance: A set $N = \{1, 2, \ldots, n\}$ and a function $v_i : N \to \mathbb{R}^+ \cup \{0\}$ such that $v_i(i) = 0$ for each $i \in N$ and $v_i(j) = v_j(i)$ for each $i, j \in N$.*

- *Question: Does a non trivial partition Π of N exist such that*

$$\forall i \in N, \forall S_k \in \Pi \cup \{\emptyset\} : \sum_{j \in S_\Pi(i)} v_i(j) \geq \sum_{j \in S_k \cup \{i\}} v_i(j) \ ?$$

Theorem 6.2 *INEFFICIENT EQUILIBRIUM is NP-complete.*

Proof.

We will show that EQUAL CARDINALITY PARTITION $\propto$ INEFFICIENT EQUILIBRIUM. By the same line of reasoning as in the proof of Theorem 6.1 we conclude that INEFFICIENT EQUILIBRIUM is NP-complete since INEFFICIENT EQUILIBRIUM is easily seen to belong to NP.

We will now show how to transform an instance of EQUAL CARDINALITY PARTITION into an equivalent instance of INEFFICIENT EQUILIBRIUM. All the members of W are players in the instance of INEFFICIENT EQUILIBRIUM and the payoff for w_i and w_j for cooperating is $c(w_i) + c(w_j) + C$. For each player w_i we also add a player z_i. Player z_i only gets a strictly positive payoff by cooperating with w_i - in this case the payoff is $2c(w_i) + C$. Figure 6.2 depicts a part of the INEFFICIENT EQUILIBRIUM instance as an undirected weighted graph. The members of W are fully connected but z_i is only connected to w_i in the graph.

We will now prove that the two instances are equivalent:

- Suppose that we have a non trivial Nash stable partition Π of the players in Figure 6.2. For $S_k \in \Pi$ we define $W_k = S_k \cap W$. The player z_i cooperates with w_i - otherwise Π would not be stable. The total payoff of $w_i \in W_k$ is $|W_k|(C + c(w_i)) + \sum_{w \in W_k} c(w)$.

   - $|W_i| = |W_j|$: If $|W_i| < |W_j|$ then all the players in W_i would be strictly better off by joining W_j. This contradicts that Π is stable.

    – $\sum_{w \in W_i} c(w) = \sum_{w \in W_j} c(w)$: Now assume $\sum_{w \in W_i} c(w) < \sum_{w \in W_j} c(w)$.
    Once again the players in W_i would be strictly better off by joining
    W_j since $|W_i| = |W_j|$. Yet another contradiction.

- Suppose that we have a non trivial partition of W into sets with equal
  cardinality and capacity. For a set W_i in this partition let S_i be the union
  of W_i and the corresponding z-members. The set of S_i's is easily seen to
  be a non trivial Nash stable partition of the game in Figure 6.2.

    $\square$

Chapter 7

Simple Games

The details of the results on *Simple Games* are presented in this chapter. The work on Simple Games is joint work with Josep Freixas, Xavier Molinero and Maria Serna from the Polytechnic University of Catalonia, Barcelona, Spain and the results can also be found in [41].

Simple game theory is a very dynamic and expanding field. Taylor and Zwicker [84] pointed out that "*few structures arise in more contexts and lend themselves to more diverse interpretations than do simple games*". Indeed, simple games cover voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo. In these systems, each voter responds with a vote of "yea" or "nay". A simple game or a yes–no voting system is a set of rules that specifies exactly which collections of "yea" votes yield passage of the issue at hand – each of these collections of "yea" voters forms a winning coalition.

Democratic societies and international organizations use a wide variety of complex rules to reach decisions. Examples, where it is not always easy to understand the consequences of the way voting is done, include the Electoral College to elect the President of the United States, the United Nations Security Council, the governance structure of the World Bank, the International Monetary Fund, the European Union Council of Ministers, the national governments of many countries, the councils in several counties, and the system to elect the major in cities or villages of many countries. Another source of examples comes from economic enterprises whose owners are shareholders of the society and divide profits or losses proportionally to the numbers of stocks they posses, but make decisions by voting according to a pre-defined rule (i.e., an absolute majority rule or a qualified majority rule).

There are several alternative ways to introduce a simple game; the most natural is by giving the list of winning coalitions, then the complementary set is the set of losing coalitions and the simple game is fully described. A considerable reduction in introducing a simple game can be obtained by considering only the list of minimal winning coalitions, i.e. winning coalitions which are minimal by the inclusion operation. Coalitions containing minimal winning coalitions are also winning. Analogously, one may present a simple game by using either the set of losing coalitions or the set of maximal losing coalitions.

We are interested in performing a complexity analysis of problems on simple games, in the case that the number of players is large, as pointed out in [33], *from a computational point of view, the key issues relating to coalitional games are, first, how such games should be represented (since the obvious representation is exponentially large in the number of players); and second, the extent to which cooperative solution concepts can be efficiently computed.* We undertake here the task of looking into these issues.

Previous results have focused on problems where the input is a subclass of the class of simple games, the so called *weighted games*. A way to describe a weighted game is to assign a (positive) real number weight to each voter, and declare a coalition to be winning precisely when its total weight meets or exceeds some predetermined quota. Not every simple game is weighted but every simple game can be decomposed as an intersection of some weighted games. Work with the complexity of problems on weighted games dates back

to [79], where Prasad and Kelly provide NP-completeness results on determining properties of weighted voting games. For instance, they show that computing standard political power indices, such as absolute Banzhaf, Banzhaf–Coleman and Shapley-Shubik, are all NP-hard problems. More recent work is related with the notion of *dimension* considered by Taylor and Zwicker [83, 84]. The dimension of a simple game is the minimum number of weighted games whose intersection coincides with the game. The computational effort to weigh up the dimension of a simple game, given as the intersection of d weighted games, was determined by Deǐneko and Woeginger [30]: computing the dimension of a simple game is a NP-hard problem. More results on solution concepts for weighted games can be found in [24, 25, 29, 33, 34, 62, 63]. There also exist works related to economics [5, 35, 46, 86].

Our first objective is to fix some natural game representations. After doing so, as usual, we analyze the complexity of transforming one representation into another and the complexity of the problem of recognizing simple games. Our second aim is to classify the complexity of testing whether a simple game is of a special type. Apart from weighted games there are some other subclasses of simple games which are very significant in the literature of voting systems. Strongness, properness, decisiveness and homogeneity are, among others, desirable properties to be fulfilled for a simple game. Our results are summarized in Tables 7.1 and 7.2.

Input → Output ↓	(N, W)	(N, L)	(N, W^m)	(N, L^M)
(N, W)	–	EXP	EXP	EXP
(N, L)	EXP	–	EXP	EXP
(N, W^m)	P	P	–	EXP
(N, L^M)	P	P	EXP	–

Table 7.1: Complexity of changing the representation form of a simple game.

Input →	(N, W)	(N, W^m)	(N, L)	(N, L^M)	$(q; w)$
IsSimple	P	P	P	P	–
IsStrong	P	co-NPC	P	P	co-NPC
IsProper	P	P	P	co-NPC	co-NPC
IsWeighted	P	P	P	P	–
IsHomogeneous	P	?	P	?	?
IsDecisive	P	?	P	?	co-NPC
IsMajority	P	?	P	?	co-NPC

Table 7.2: Our results on the complexity of problems on simple games.

Table 7.1 shows the complexity of passing from a given form to another one. All *explicit* forms are represented by a pair (N, C) in which $N = \{1, \ldots, n\}$ for some positive integer n, and C is the set of winning, minimal winning, losing

or maximal losing coalitions. Note that it is possible to pass from winning and losing coalitions to minimal winning and maximal losing coalitions in polynomial time, but the other swaps require exponential time. On the other hand, given a game in a specific form, Table 7.2 shows the complexity of determining whether it is simple, strong, proper, weighted, homogeneous, decisive or majority. Here $(q; w)$ denotes an *integer representation* of a weighted game where q is the quota and w are the weights. Observe that there are some problems that still remain open.

Finally, we refer the reader to Papadimitriou [77] for the definitions of the complexity classes P, NP, co-NP, and their subclasses of complete problems NPC and co-NPC.

7.1 Recognizing simple games

We start stating some basic definitions on simple games (we refer the interested reader to [84] for a thorough presentation).

Simple games can be viewed as models of voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo.

Definition 7.1 *A simple game* Γ *is a pair* (N, W) *in which* $N = \{1, \ldots, n\}$ *for some positive integer* n, *and* W *is a collection of subsets of* N *that satisfies* $N \in W$, $\emptyset \notin W$, *and the* monotonicity *property:* $S \in W$ *and* $S \subseteq R \subseteq N$ *implies* $R \in W$.

Any set of voters is called a *coalition*, the set N is called the *grand coalition*, and the empty set $\emptyset$ is called the *null coalition*. Members of N are called *players* or *voters*, and the subsets of N that are in W are called *winning coalitions*. The intuition here is that a set S is a winning coalition *iff* the bill or amendment passes when the players in S are precisely the ones who vote for it. A subset of N that is not in W is called a *losing coalition*. The collection of losing coalitions is denoted by L. The set of *minimal winning coalitions* (*maximal losing coalitions*) is denoted by W^m (L^M), where a minimal winning coalition (a maximal losing coalition) is a winning (losing) coalition all of whose proper subsets (supersets) are losing (winning). Because of monotonicity, any simple game is completely determined by its set of minimal winning coalitions. A voter i is null if $i \notin S$ for all $S \in W^m$.

From a computational point of view a simple game can be given under different representations. In this chapter we essentially consider the following options:

- Explicit or Extensive winning form: the game is given as (N, W) by providing a listing of the collection of subsets W.

- Explicit or Extensive minimal winning form: the game is given as (N, W^m) by providing a listing of the family W^m. Observe that this form requires less space than the explicit winning form whenever $W \neq \{N\}$.

When we consider descriptions of a game in terms of winning coalitions in this chapter, we also consider the corresponding representations for losing coalitions, replacing minimal by maximal. Thus, in addition we also consider the explicit or extensive losing, and explicit or extensive maximal losing forms.

We analyze first the computational complexity of obtaining a representation of a game in a given form when a representation in another form is given.

Theorem 7.1 *Given a simple game:*

    i. passing from the explicit winning (losing) form to the explicit minimal winning and maximal losing (maximal losing and minimal winning) form can be done in polynomial time.

    ii. passing from the explicit minimal winning (maximal losing) form to the explicit winning (losing) form requires exponential time.

    iii. passing from the explicit minimal winning (maximal losing) form to the explicit maximal losing (minimal winning) form requires exponential time.

    iv. passing from the explicit minimal winning (maximal losing) form to the explicit losing (winning) form requires exponential time.

    v. passing from the explicit winning (losing) form to the explicit losing (winning) form requires exponential time.

This theorem gives us all the results presented in Table 7.1. The polynomial time results are obtained from simple properties of monotonic sets. For the exponential time transformations we provide examples in which the size of the representation increases exponentially. The transformations are similar to the ones used to show that computing a CNF[1] from a given DNF[2] requires exponential time. The difference relies in that now instead of transforming the same formula we have to get a different maximal normal form for a formula and its negation.

Before proving Theorem 7.1 in detail, we introduce some notation and definitions together with some preliminary technical results.

Given a family of subsets C of a set N, $\overline{C}$ denotes the closure of C under $\subseteq$, and $\underline{C}$ the closure of C under $\supseteq$.

Definition 7.2 *A subset C of a set N is closed under $\subseteq$ ($\supseteq$) if $C = \overline{C}$ ($\underline{C}$).*

The following lemma is proved in [77].

Lemma 7.1 *Given a family of subsets C of a set N, we can check whether it is closed under $\subseteq$ or $\supseteq$ in polynomial time.*

---

[1]A Boolean formula is in *Conjunctive Normal Form* (CNF) if it is a conjunction of disjunction of literals.

[2]A Boolean formula is in *Disjunctive Normal Form* (DNF) if it is a standardization (or normalization) of a logical formula which is a disjunction of conjunction of literals.

Lemma 7.2 *Given a family of subsets C of a set N, the families $\overline{C}^m$ and $\underline{C}^M$ can be obtained in polynomial time.*

Proof. Observe that, for any set S in C we have to check whether there is a subset (superset) of S that forms part of C, and keep those S that do not have this property. Therefore, the complete computation can be done in polynomial time on the input length of C.   □

Now we define the minimal and maximal subset families.

Definition 7.3 *Given a family of subsets C of a set N, we say that it is minimal if $C = \overline{C}^m$.*

Definition 7.4 *Given a family of subsets C of a set N, we say that it is maximal if $C = \underline{C}^M$.*

As a consequence of Lemma 7.2 we have the following corollary.

Corollary 7.1 *Given a family of subsets C of a set N, we can check whether it is maximal or minimal in polynomial time.*

The proof of Theorem 7.1 is split into five lemmata. We start with our first result for simple games given in explicit winning or losing form.

Lemma 7.3 *Given a simple game Γ in explicit winning (or losing) form, the representation of Γ in explicit minimal winning or maximal losing (maximal losing or minimal winning) form can be obtained in polynomial time.*

Proof. Given a simple game $\Gamma = (N, W)$, consider the set

$$R = \bigcup_{i=1}^{|N|} W_{-i}$$

where $W_{-i} = \{S \setminus \{i\} : i \in S \in W\}$. Observe that all the sets in $R \setminus W$ are losing coalitions, $R \setminus W \subseteq L$. We claim that $(R \setminus W)^M = L^M$. We are going to prove that in two steps:

- $(R \setminus W)^M \subseteq L^M$: Now suppose that $T \in (R \setminus W)^M$ and that $T \notin L^M$. Consequently we have that $T \in L$ and that $T \cup \{i\} \in W$ for some $i \in N$. We also have that $T \subset U$ for some $U \in L$. Due to the monotonicity we conclude that $U \cup \{i\} \in W$. This means that $U \in R \setminus W$ which contradicts that T is maximal in $R \setminus W$.

- $L^M \subseteq (R \setminus W)^M$: We will show this inclusion in two steps:

  i. $L^M \subseteq R \setminus W$: If $T \in L^M$ then $T \cup \{i\} \in W$ for any $i \notin T$. Thus T can be obtained from a winning coalition $(T \cup \{i\})$ from removing an element (i). This means that $T \in R \setminus W$ since T is a losing coalition.

  ii. Maximal elements in a set will also be maximal in any subset they appear in. From $L^M \subseteq R \setminus W \subseteq L$ we conclude that $L^M \subseteq (R \setminus W)^M$.

For the cost of the algorithm, observe that, given (N, W), the set R has cardinality at most $|N| \cdot |W|$, and thus R can be obtained in polynomial time. Using Lemma 7.2, from W and $R \setminus W$, we can compute W^m and L^M in polynomial time.

Analogously, when the game is given by the family of losing coalitions a symmetric argument provides the proof for explicit maximal losing or minimal winning form. □

Now we focus on simple games given in explicit minimal winning or explicit maximal losing form.

Lemma 7.4 *Given a simple game Γ in explicit minimal winning (maximal losing) form, computing the representation of Γ in explicit winning (losing) form requires exponential time.*

Proof. The following two examples show that the size of the computed family can be exponential in the size of the given one. Therefore, any algorithm that solves the problem requires exponential time.

Consider $N = \{1, \ldots, n\}$, then:

i. The simple game defined by $W^m = \bigcup_{i=1}^{n} \{\{i\}\}$ has $W = \{T \subseteq N : T \neq \emptyset\}$. Therefore, $|W^m| = n$ and $|W| = 2^n - 1$.

ii. The simple game defined by $L^M = \{T \subseteq N : |T| = n - 1\}$ has $L = \{T \subset N\}$. Therefore, $|L^M| = n$ and $|L| = 2^n - 1$.

□

Lemma 7.5 *Given a simple game Γ in explicit minimal winning (maximal losing) form, computing the representation of Γ in explicit maximal losing (minimal winning) form requires exponential time.*

Proof. In a similar way as we did in the previous Lemma, we show two examples in which the size of the computed family can be exponential in the size of the given one.

Consider $N = \{1, \ldots, 2n\}$ and coalitions $S_i = \{2i - 1, 2i\}$, for all $i = 1, \ldots, n$. Then,

i. The simple game defined by $W^m = \bigcup_{i=1}^{n} \{S_i\}$ has

$$L^M = \{T \subseteq N : |T \cap S_i| = 1, \text{for all } i = 1, \ldots, n\}.$$

Therefore, $|W^m| = n$ and $|L^M| = 2^n$.

ii. The simple game defined by

$$W^m = \{T \subseteq N : |T \cap S_i| = 1, \text{for all } i = 1, \ldots, n\}$$

has $L^M = \bigcup_{i=1}^{n} \{N \setminus S_i\}$. Therefore, $|W^m| = 2^n$ and $|L^M| = n$.

□

As a consequence of Lemmata 7.3 and 7.5 we have Corollary 7.2.

Corollary 7.2 *Given a simple game* Γ *in explicit minimal winning (maximal losing) form, computing the representation of* Γ *in explicit losing (winning) form requires exponential time.*

The remaining cases of Theorem 7.1 are again computationally hard.

Lemma 7.6 *Given a simple game* Γ *in explicit winning (losing) form, computing the representation of* Γ *in explicit losing (winning) form requires exponential time.*

Proof. We present two examples where the size of the computed family is exponential in the size of the given one. Let (N, W) be the game, where $W = \{N\}$, then $|W| = 1$ and $|L| = 2^{|N|} - 1$. Similarly, let (N, W) be the game, where $L = \{\emptyset\}$, then $|W| = 2^{|N|} - 1$ and $|L| = 1$.   □

Lemmata (7.3)-(7.6) make up Theorem 7.1.

The next step is to analyze the computational complexity of the following recognition problems:

> Name: IsSimpleE
> Input: (N, C)
> Question: Is (N, C) a correct explicit representation of a simple game?

We have in total four different problems depending on the input description: winning, minimal winning, losing and maximal losing. However, the recognition problem becomes polynomial time solvable in all these cases.

Theorem 7.2 *The* IsSimpleE *problem belongs to P for any explicit form* F: *winning, minimal winning, losing, or maximal losing.*

Proof. The proof follows from the fact that given a family of subsets C of a set N, the families of minimal or maximal sets of its closure can be obtained in polynomial time. It is a direct consequence of Lemmata 7.1 and 7.2 and Corollary 7.1, stating that whether the family is monotonic[3] or minimal/maximal can be tested in polynomial time.   □

Observe that, as the recognition problem can be solved in polynomial time, we can use any of the proposed representations in the complexity analysis to follow.

---

[3]We say that a family of sets is *monotonic iff* it satisfies the monotonicity property.

7.2 Problems on simple games

In this section we consider a set of decision problems related to properties that define some special types of simple games (again we refer the reader to [84]). In general we will state a property P for simple games and consider the associated decision problem which has the form:

> Name: IsP
> Input: A simple game Γ
> Question: Does Γ satisfy property P?

Further considerations on the complexity of such problems will be stated in terms of the input representation.

7.2.1 Recognizing strong and proper games

Now we study the complexity of determining if a given simple game (in explicit form) is strong, weak, proper or improper.

Definition 7.5 *A simple game* (N, W) *is* strong *if* $S \notin W$ *implies* $N \setminus S \in W$. *A simple game that is not strong is called* weak.

Intuitively speaking, if a game is weak it has to few winning coalitions, because adding sufficiently many winning coalitions would make the game strong. Note that the addition of winning coalitions can never destroy strongness.

Definition 7.6 *A simple game* (N, W) *is* proper *if* $S \in W$ *implies* $N \setminus S \notin W$. *A simple game that is not proper is called* improper.

An improper game has to many winning coalitions, in the sense that deleting sufficiently many winning coalitions would make the game proper. Note that the deletion of winning coalitions can never destroy properness.

When a game is both proper and strong, a coalition wins *iff* its complement loses. Therefore, in this case we have $|W| = |L| = 2^{n-1}$.

A related concept with the properness and strongness is the dualityness.

Definition 7.7 *Given a simple game* (N, W), *its* dual game *is* (N, W^*), *where* $S \in W^*$ *if and only if* $N \setminus S \notin W$.

That is, winning coalitions in the dual game are just the "blocking" coalitions in the original game. Thus, (N, W) is proper *iff* (N, W^*) is strong, and (N, W) is strong *iff* (N, W^*) is proper.

Theorem 7.3 *The* IsStrong *problem, when the input game is given in explicit losing or maximal losing form, and the* IsProper *problem, when the game is given in explicit winning or minimal winning form, can be solved in polynomial time.*

Proof.

To prove this result we provide an adequate formalization of the strong and proper properties in terms of simple properties of the set of minimal winning or maximal losing coalitions respectively. Those properties can be checked in polynomial time when the games are given in the specified forms.

First observe that, given a family of subsets F, we can check, for any set in F, whether its complement is not in F in polynomial time. Therefore, taking into account the definitions, we have that the IsStrong problem, when the input is given in explicit losing form, and the IsProper problem, when the input is given in explicit winning form, are polynomial time solvable.

Thus, taking into account that

- A simple game is weak *iff*

$$\exists S \subseteq N : S \in L \ \wedge \ N \setminus S \in L$$

  which is equivalent to

$$\exists S \subseteq N : \exists L_1, L_2 \in L^M : S \subseteq L_1 \ \wedge \ N \setminus S \subseteq L_2$$

  The last assertion is equivalent to the fact that there are two maximal losing coalitions L_1 and L_2 such that $L_1 \cup L_2 = N$.

- A simple game is *improper iff*

$$\exists S \subseteq N : S \in W \ \wedge \ N \setminus S \in W$$

  which is equivalent to

$$\exists S \subseteq N : \exists W_1, W_2 \in W^m : W_1 \subseteq S \ \wedge \ W_2 \subseteq N \setminus S.$$

  This last assertion is equivalent to the fact that there are two minimal winning coalitions W_1 and W_2 such that $W_1 \cap W_2 = \emptyset$.

Observe that, given a family of subsets F, checking whether any one of the two conditions hold can be done in polynomial time. Thus the theorem holds also when the set of maximal losing (or minimal winning) coalitions is given. $\square$

As a consequence of Theorems 7.1 and 7.3 we have:

Corollary 7.3 *The* IsStrong *problem, when the input game is given in explicit winning form, and the* IsProper *problem, when the game is given in explicit losing form, can be solved in polynomial time.*

Our next result states the complexity of the IsStrong problem when the game is given in the remaining form.

Theorem 7.4 *The* IsStrong *problem is co-NP-complete when the input game is given in explicit minimal winning form.*

Proof. The membership proof follows from an adequate formalization. To prove hardness we consider the *set splitting* problem which asks whether it is possible to partition N into two subsets P and $N \setminus P$ such that no subset in a given collection C is entirely contained in either P or $N \setminus P$. It is known that the problem is NP-complete [43]. We provide a polynomial time reduction from *set splitting* to the IsWEAK problem. In other words we have to decide whether $P \subseteq N$ exists such that

$$\forall S \in C : S \nsubseteq P \wedge S \nsubseteq N \setminus P \tag{7.1}$$

We transform a set splitting instance (N, C) into the simple game in explicit minimal winning form (N, C^m). This transformation can be computed in polynomial time according to Lemma 7.2. We will now show that (N, C) has a set splitting *iff* (N, C^m) is a weak game:

- Now assume that $P \subseteq N$ satisfying (7.1) exists. This means that P and $N \setminus P$ are losing coalitions in the game (N, C^m).

- Let P and $N \setminus P$ be losing coalitions in the game (N, C^m). As a consequence we have that $S \nsubseteq P$ and $S \nsubseteq N \setminus P$ for any $S \in C^m$. This implies that $S \nsubseteq P$ and $S \nsubseteq N \setminus P$ holds for any $S \in C$ since any set in C contains a set in C^m.

$\square$

Finally we prove a similar complexity result for the remaining version of the IsPROPER problem.

Theorem 7.5 *The* IsPROPER *problem is co-NP-complete when the input game is given in extensive maximal losing form.*

Proof. The hardness of the IsPROPER problem is obtained by using duality and providing a polynomial time reduction from the IsSTRONG problem.

From Definition 7.6, a game is *improper* if and only if there exists a coalition $S \subseteq N$ such that neither S nor $N \setminus S$ is contained in a member of L^M. For a given coalition S we can easily perform this check in polynomial time. Therefore the problem IsIMPROPER belongs to NP, and the problem IsPROPER belongs to co-NP.

To complete the proof we provide a reduction from the IsSTRONG problem for games given in extensive minimal winning form. First observe that, if a family C of subsets of N is minimal then the family $\{N \setminus L : L \in C\}$ is maximal. Given a game $\Gamma = (N, W^m)$, in minimal winning form, let us consider its dual game $\Gamma' = (N, \{N \setminus L : L \in W^m\})$ given in maximal losing form. Of course Γ' can be obtained from Γ in polynomial time. Thus Γ is weak *iff*

$$\exists S \subseteq N : S \in L(\Gamma) \ \wedge \ N \setminus S \in L(\Gamma)$$

which is equivalent to

$$\exists S \subseteq N : N \setminus S \in W(\Gamma') \ \wedge \ S \in W(\Gamma')$$

iff Γ' is improper.

Thus, the IsPROPER problem belongs to co-NP and it is co-NP-hard – in other words it is co-NP-complete.   □

7.2.2   Recognizing weighted games

In this subsection we study the complexity of determining if a given simple game (in explicit form) is weighted, homogeneous, majority or decisive.

Definition 7.8 *A simple game* (N, W) *is* weighted *if there exist a "quota"* $q \in \mathbb{R}$ *and a "weight function"* $w : N \to \mathbb{R}$ *such that each coalition* S *is winning exactly when the sum of weights of* S *meets or exceeds* q.

Weighted games are probably the most important kind of simple games. Any specific example of a weight function w and quota q is said to *realize G* as a weighted game. A particular realization of a weighted game is denoted $(q; w_1, \ldots, w_n)$, or briefly $(q; w)$. By $w(S)$ we denote $\sum_{i \in S} w_i$.

Observe also that, from the *monotonicity property*, it is obvious that a simple game (N, W) is *weighted iff* there exist a "quota" $q \in \mathbb{R}$ and a "weight function" $w : N \to \mathbb{R}$ such that

$$w(S) \geq q \qquad \forall\, S \in W^m$$
$$w(S) < q \qquad \forall\, S \in L^M.$$

Theorem 7.6 *The* IsWEIGHTED *problem can be solved in polynomial time when the input game is given in explicit winning, losing, minimal winning and maximal losing forms.*

Proof. A simple polynomial time reduction from the IsWEIGHTED problem to the *Linear Programming* problem, which is known to be solvable in polynomial time [52, 55], gives the result for the cases of explicit winning and explicit losing forms.

Taking into account Lemma 7.2, in both cases we can obtain W^m and L^M in polynomial time. Once this is done we can write, again in polynomial time, the following *Linear Programming* instance Π:

$$
\begin{aligned}
&\min q \\
&\text{subject to} \quad && w(S) \geq q && \text{if } S \in W^m \\
& && w(S) < q && \text{if } S \in L^M \\
& && 0 \leq w_i && \text{for all } 1 \leq i \leq n \\
& && \textstyle\sum_i w_i = 1 \\
& && 0 \leq q
\end{aligned}
$$

As (N, W) is weighted *iff* Π has a solution, the proposed construction is a polynomial time reduction.

For the minimal winning form we provide a reduction to the *threshold function* problem for monotonic DNF formula which is known to be polynomial time solvable [49, 78]. For the maximal losing form we make use of duality

and provide a reduction to the problem when the input is described in minimal winning form.

Given (N, W^m), we are going to prove that we can decide in polynomial time whether a simple game is weighted.

For $C \subseteq N$ we let $x_C \in \{0,1\}^n$ denote the vector with the i'th coordinate equal to 1 if and only if $i \in C$. In polynomial time we transform W^m into the Boolean function Φ_{W^m} given by the DNF formula:

$$\Phi_{W^m}(x) = \bigvee_{S \in W^m} (\wedge_{i \in S} x_i)$$

By construction we have the following:

$$\Phi_{W^m}(x_C) = 1 \Leftrightarrow C \text{ is winning in the game given by } (N, W^m) \qquad (7.2)$$

Note that Φ_{W^m} is a threshold function if and only if the game given by (N, W^m) is weighted:

- **only if** ($\Rightarrow$): Assume that Φ_{W^m} is a threshold function. Let $w \in \mathbb{R}^n$ be the weights and $q \in \mathbb{R}$ the threshold value. Thus we have that

$$\Phi_{W^m}(x_C) = 1 \Leftrightarrow \langle w, x_C \rangle \geq q$$

  where $\langle \cdot, \cdot \rangle$ denotes the usual inner product. By using (7.2) we conclude that the game given by (N, W^m) is weighted.

- **if** ($\Leftarrow$): Now assume that the game given by (N, W^m) is weighted and that $(q; w)$ is a realization of the game. In this case we have the following:

$$C \text{ is winning in the game given by } (N, W^m) \Leftrightarrow \langle w, x_C \rangle \geq q$$

  Again we use (7.2) and conclude that Φ_{W^m} is a threshold function.

The Boolean function Φ_{W^m} is monotonic (i.e. *positive*) so according to the papers [49, 78] (pages 211 and 59, respectively) we can in polynomial time decide whether Φ_{W^m} is a threshold function. Consequently we can also decide in polynomial time whether the game given by (N, W^m) is weighted.

On the other hand, we can prove a similar result given (N, L^M) just taking into account that a game Γ is weighted *iff* its dual game Γ' is weighted. Then, we can use the technique from the proof of Theorem 7.5. $\square$

It is important to remark that it is known that "*a simple game is weighted* iff *it is trade robust* iff *it is invariant-trade robust*" [33, 40, 84]. Thus, according to Theorem 7.6, checking whether a simple game is trade robust or invariant-trade robust can be done in polynomial time.

Corollary 7.4 *The* ISTRADEROBUST *and the* ISINVARIANTTRADEROBUST *problem can be solved in polynomial time when the input game is given in explicit winning, minimal winning, losing or maximal losing form.*

7.2.3   Recognizing homogeneous, decisive and majority games

In this section we define the homogeneous, decisive and majority games and, afterwards, we analyze the complexity of the ISHOMOGENEOUS, ISDECISIVE and ISMAJORITY problems.

Definition 7.9 *A weighted game* (N, W) *is* homogeneous *if there exist a realization* $(q; w)$ *such that* $q = w(S)$ *for all* $S \in W^m$.

That is, a weighted game is homogeneous *iff* the sum of the weights of any minimal winning coalition is equal to the quota.

Theorem 7.7 *The* ISHOMOGENEOUS *problem can be solved in polynomial time when the input game is given in explicit winning or losing form.*

Proof. The polynomial time reduction from the ISHOMOGENEOUS problem to the *Linear Programming* problem, is done in the same way as in the proof of Theorem 7.6, but considering the instance Π' obtained by replacing $w(S) \geq q$, in the first set of inequalities of Π, by $w(S) = q$. It is immediate to see that (N, W) is homogeneous *iff* Π' has a solution. This modification provides the proof of Theorem 7.7.   □

Now we introduce the remaining subclasses of simple games.

Definition 7.10 *A simple game is* decisive *(or* self–dual, *or* constant sum*) if it is proper and strong. A simple game is* indecisive *if it is not decisive.*

Note that the decisiveness is related with the dualityness. As we said before, (N, W) is proper *iff* (N, W^*) is strong, and (N, W) is strong *iff* (N, W^*) is proper. As a consequence, we have that a simple game (N, W) is decisive *iff* $W = W^*$. On the other hand, W is closed under $\subseteq$ or $\supseteq$ *iff* W^* is closed under $\subseteq$ or $\supseteq$, respectively.

In the seminal work on game theory by von Neumann and Morgenstern [66] only decisive simple games were considered. Nowadays, many governmental institutions make their decisions through voting rules that are in fact decisive games. If abstention is not allowed (see [42] for voting games with abstention) ties are not possible in decisive games.

Another interesting subclass of simple games are the so–called majority games:

Definition 7.11 *A simple game is a* majority game *if it is weighted and decisive.*

Observe that, although a simple game can fail to be proper and fail to be strong, this cannot happen with weighted games (the proof appears in [84]).

Proposition 7.1 *Any weighted game is either proper or strong.*

From Proposition 7.1, it follows that there are three kind of weighted games: proper but not strong, strong but not proper, and both strong and proper.

From Theorem 7.6 and taking into account that decisive games are characterized by having 2^{n-1} winning coalitions, we have the following result.

Theorem 7.8 *The* IsMajority *and the* IsDecisive *problems can be solved in polynomial time when the input game is given in explicit winning or losing form.*

Proof. Given a monotonic simple game (N, W), we can check whether it is strong and proper by checking $|W| = 2^{n-1}$ and $S \in W \Rightarrow N \setminus S \notin W$ in polynomial time. We check (N, L) in a similar way. Furthermore, under both forms, we can check in addition whether the game is weighted in polynomial time using Theorem 7.6. □

7.3 Problems on weighted games

In this section we consider weighted games which are described by an integer realization $(q; w)$. Observe that it is well–known that any weighted game admits an integer realization (see for instance [15]), that is, a weight function with nonnegative integer values, and a positive integer as quota. Integer realizations naturally arise; just consider the seats distributed among political parties in any voting system. In consequence we assume an integer realization as representation of a weighted game. We analyze the complexity of problems of the type:

> Name: IsP
> Input: An integer realization $(q; w)$ of a weighted game Γ.
> Question: Does Γ satisfy P?

We are interested in such problems associated to the properties of being strong, proper, homogeneous, and majority[4]. Observe that for weighted games majority and decisive are just the same property, so we consider only the majority games.

From now on some of the proofs are based on reductions from the NP-complete problem Partition [43], which is defined as:

> Name: Partition
> Input: n integer values, $x_1, \ldots, x_n$
> Question: Is there $S \subseteq \{1, \ldots, n\}$ for which $\sum_{i \in S} x_i = \sum_{i \notin S} x_i$.

Observe that, for any instance of the Partition problem in which the sum of the n input numbers is odd, the answer must be no.

---

[4]Note that the definition of majority weighted games given in [30] is equivalent to our definition of weighted games.

Theorem 7.9 *The* IsStrong, IsProper *and* IsMajority *(here, equivalent to* IsDecisive*) problems, when the input is described by an integer realization of a weighted game* $(q; w)$*, are co-NP-complete.*

Proof. From the definitions of strong, proper and majority games, it is straightforward to show that the three problems belong to co-NP.

Observe that the weighted game with integer representation $(2; 1, 1, 1)$ is proper and strong, and thus decisive.

We transform an instance $x = (x_1, \ldots, x_n)$ of the Partition problem into a realization of a weighted game according to the following schema

$$f(x) = \begin{cases} (q(x); x) & \text{when } x_1 + \cdots + x_n \text{ is even,} \\ (2; 1, 1, 1) & \text{otherwise.} \end{cases}$$

The function f can be computed in polynomial time provided q does, and we will use a different q for each problem.

Nevertheless, independently of q, when $x_1 + \cdots + x_n$ is *odd*, x is a NO input for partition, but $f(x)$ is a YES instance of IsStrong, IsProper, and IsMajority, and thus a NO instance of the complementary problems.

Therefore, we have to take care only of the case in which $x_1 + \cdots + x_n$ is *even*. Assume that this is the case and let $s = (x_1 + \cdots + x_n)/2$ and $N = \{1, \ldots, n\}$. We will provide the proof that f reduces Partition to the complementary problem.

a) IsStrong *problem.*

For the case of strong games, taking $q(x) = s + 1$, we have:

- If there is a $S \subset N$ such that $\sum_{i \in S} x_i = s$, then $\sum_{i \notin S} x_i = s$, thus both S and $N \setminus S$ are losing coalitions and $f(x)$ is weak.

- Now assume that S and $N \setminus S$ are both losing coalitions in $f(x)$ If $\sum_{i \in S} x_i < s$ then $\sum_{i \notin S} x_i \geq s + 1$, which contradicts that $N \setminus S$ is losing. Thus we have that $\sum_{i \in S} x_i = \sum_{i \notin S} x_i = s$, and there exists a partition of x.

Therefore, f is a polytime reduction from Partition to IsWeak

b) IsProper *problem.*

For the case of proper games we take $q(x) = s$. Then, if there is a $S \subset N$ such that $\sum_{i \in S} x_i = s$, then $\sum_{i \notin S} x_i = s$, thus both S and $N \setminus S$ are winning coalitions and $f(x)$ is improper. When $f(x)$ is improper

$$\exists S \subseteq N : \sum_{i \in S} x_i \geq s \wedge \sum_{i \notin S} x_i \geq s,$$

and thus $\sum_{i \in S} x_i = s$. Thus, we have a polytime reduction from Partition to IsImproper.

c) IsMajority *problem.*

For the case of majority games we take again $q(x) = s$. Observe that $f(x)$ cannot be weak, as in such a case there must be some $S \subseteq N$ for which,

$$\sum_{i \in S} x_i < s \wedge \sum_{i \notin S} x_i < s,$$

contradicting the fact that $s = (x_1 + \cdots + x_n)/2$. Therefore, the game is not majority *iff* it is improper, and the claim follows. □

Before finishing this section we introduce the following related problem:

Name: IsHomogeneousRealization
Input: An integer realization $(q; w)$ of a weighted game Γ.
Question: Is $(q; w)$ a homogeneous realization?

Given the weights w, Rosenmüller [80] solves the problem of computing all q such that $(q; w)$ is a homogeneous realization. Although in [80] the analysis on the complexity is omitted, it is easy to check that the dynamic programming algorithm given in Section 3 of [80] runs in polynomial time. Thus, given an integer realization $(q; w)$ it can be checked whether it is a homogeneous realization in polynomial time.

Theorem 7.10 *The* IsHomogeneousRealization *problem can be solved in polynomial time.*

Note that, given an integer realization $(q; w)$ of a weighted game, we cannot yet check whether this game is homogeneous, only whether a given realization is a homogeneous one. We want to remark that the previous result does not imply that the IsHomogeneous problem belongs to NP. Consider the problem

Name: IsAnotherRealization
Input: Two integer realizations $(q; w)$ and $(q'; w')$.
Question: Is $(q'; w')$ another realization of the game (q, w)?

In [34] it is shown that the IsAnotherRealization problem is co-NP-complete: it is easy to see that $(x_1, \ldots, x_n)$ is a no instance of Partition if and only if $(s + 1; x)$ is another realization of $(s; x)$.

7.4 Succinct representations

We finish the analysis of simple games introducing a natural succinct representation of families of sets by means of Boolean formula. A Boolean formula Φ on n variables provides a compact description of a family of subsets C of a set N with n elements in the following way: we associate to each truth assignment $x = (x_1, \ldots, x_n)$ the set $A_x = \{i \mid x_i = 1\}$. Therefore Φ describes the family of subsets $\{A_x \mid \Phi(x) = 1\}$ in a compact way. In consequence we consider the following succinct representations

- Succinct winning form: the game is given by (N, Φ) where Φ is a Boolean formula on $|N|$ variables providing a compact description of the sets in W.

- Succinct minimal winning form: the game is given by (N, Φ) but now Φ describes the family W^m. Observe again that this form might require less space than the previous one whenever $W \neq \{N\}$.

In addition we consider the succinct losing and maximal losing forms. Our first objective again is to analyze the complexity of the recognition problem.

> Name: IsSimpleS
> Input: (N, Φ)
> Question: Is (N, Φ) a correct succinct representation of a simple game?

As it happened with IsSimpleE problem, we have in total four different problems depending on the input description: winning, minimal winning, losing and maximal losing.

Unfortunately we can show that the recognition problem is hard in all the proposed succinct forms thus forbidding a posterior use of such representations.

Theorem 7.11 *The* IsSimpleS *problem is co-NP-complete for any succinct form* F*: winning or losing, and co-NP-hard for any succinct form* F*: minimal winning or maximal losing.*

Proof. Observe that, from the Definition 7.1 of the *monotonicity property*, a set $W(L)$ is not monotonic *iff* there are two sets S_1 and S_2 such that $S_1 \subseteq S_2$ but $S_1 \in W$ and $S_2 \notin W$ ($S_1 \notin L$ and $S_2 \in L$). When the game is given in succinct winning or losing form, these tests can be done by guessing two truth assignments x_1 and x_2 and checking that $x_1 < x_2$, $\Phi_W(x_1) = 1$ and $\Phi_W(x_2) = 0$ ($\Phi_L(x_1) = 0$ and $\Phi_L(x_2) = 1$). Both properties can be checked in polynomial time once S_1 and S_2 are given. Thus the problems belong to co-NP.

A Boolean formula is *monotonic* if for any pair of truth assignments x, y, such that $x \leq y$ in canonical order (i.e., $x_i \leq y_i$ for all i), we have that $\Phi(x) \leq \Phi(y)$ (assuming that false $<$ true). The latter problem (i.e., to know whether a Boolean formula is monotonic or not) is co-NP-complete (even for DNF formulas) [64]. Consider the following reduction: Given a boolean formula Φ on n variables we construct Φ' on $n + 2$ variables as follows

$$\Phi'(\alpha\beta x) = \begin{cases} 1 & \alpha = \beta = 1 \\ 0 & \alpha = \beta = 0 \\ \Phi(x) & \alpha \neq \beta \end{cases}$$

Now we have that Φ is monotonic iff Φ' is monotonic. Furthermore we have that Φ' is monotonic iff (N, Φ') is a simple game in the explicit winning form since $\Phi'(1^n) = 1$ and $\Phi'(0^n) = 0$. This shows that IsSimpleS for the explicit

winning form is co-NP-complete. Observe that (N, Φ_L) is an explicit loosing representation of a simple game iff $(N, \neg\Phi_L)$ is an explicit winning representation of a simple game. Then the IsSimpleS for the explicit loosing form is co-NP-complete.

Recall now that the SAT problem asks whether a given Boolean formula has a satisfying assignment. SAT is a well known NP-complete problem. Consider the following reduction: Given a boolean formula ϕ on n variables we construct Φ for minimal winning forms on $n + 2$ variables as follows

$$\Phi(\alpha\beta x) = \begin{cases} 1, & \text{if } \alpha = \beta = 1 \text{ and } x = 1^n \\ 0, & \text{if } \alpha = \beta = 1 \text{ and } x \neq 1^n \\ \phi(x), & \text{if } \alpha \neq \beta \\ 0, & \text{if } \alpha = \beta = 0 \end{cases}$$

We have that ϕ does not have satisfying assignment *iff* Φ describes a non empty minimal winning set. Similarly for maximal losing forms, now we should consider

$$\Phi(\alpha\beta x) = \begin{cases} 0, & \text{if } \alpha = \beta = 1 \\ \phi(x), & \text{if } \alpha \neq \beta \\ 0, & \text{if } \alpha = \beta = 0 \text{ and } x \neq 0^n \\ 1, & \text{if } \alpha = \beta = 0 \text{ and } x = 0^n \end{cases}$$

Thus the minimal winning and the maximal losing problems are co-NP-hard. □

Observe that in the case that Φ represents $W^m(L^M)$ we have to check on one side that the represented set is minimal (maximal) and second that the formula has a satisfying assignment different from 0^n. This places the problem in the class DP [77]. The exact classification of those problems remains open.

7.5 Open Problems on Simple Games

As this is the first time in which problems on simple games are analyzed there are still many interesting open question as there are many other interesting properties on simple games. With respect to the unclassified problems on Table 7.2 we conjecture the following:

Conjecture 7.1 *The* IsDecisive *problem is co-NP-complete when the input game is given in explicit minimal winning or maximal losing form.*

Conjecture 7.2 *The* IsMajority *problem is co-NP-complete when the input game is given in explicit minimal winning or maximal losing form.*

We would also like to remark that our study can be enlarged by considering new explicit forms to present a simple game. For example, blocking coalitions and minimal blocking coalitions provide an alternative way to fully describe a simple game. Precisely, a blocking coalition wins whenever its complementary

loses. From the point of view of succinct representations, there are other pro-
posals for representing a simple game, which make use of Boolean functions
or weighted representations. For example the multilinear extension of a sim-
ple game [75], *succinct representations* [64], or the intersection of a collection
of weighted games [30]. It will be of interest to perform a similar complexity
analysis on such representations.

Interestingly enough, we have shown in Theorem 7.6 that we can decide
in polynomial time whether a simple game is weighted. This result opens the
possibility of analyzing the complexity of problems on weighted games described
in a explicit form. In particular, as weighted games are games with dimension
1, our results imply that we can decide in polynomial time whether a simple
game has dimension 1. Recall that the results in [30] show that computing the
dimension of a simple game is NP-hard. The latter result is obtained when
the game is described as the intersection of some weighted games. It will be of
interest to determine whether the dimension of a simple game given in explicit
form can be computed in polynomial time. The same questions can also be
formulated for other parameters and solution concepts on simple games.

Bibliography

[1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486. IEEE Computer Society, 2006.

[2] R. Andersen, F. R. K. Chung, and K. J. Lang. Local partitioning for directed graphs using PageRank. In *Proc. Algorithms and Models for the Web-Graph, 5th International Workshop, WAW 2007*, volume 4863 of *Lecture Notes in Computer Science*, pages 166–178. Springer, 2007.

[3] R. Andersen and K. J. Lang. Communities from seed sets. In *Proc. 15th International Conference on World Wide Web, WWW 2006*, pages 223–232. ACM, 2006.

[4] K. Avrachenkov and N. Litvak. The effect of new links on Google Pagerank. *Stochastic Models*, 22(2):319–331, 2006.

[5] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proc. 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 1023–1030. IFAAMAS, 2008.

[6] J. Bagrow and E. Bollt. A local method for detecting communities. *Physical Review E*, 72:046108, 2005.

[7] C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, Oct 2004.

[8] L. Becchetti and C. Castillo. The distribution of PageRank follows a power-law only for particular values of the damping factor. In *Proc. 15th International Conference on World Wide Web, WWW 2006*, pages 941–942. ACM, 2006.

[9] A. Bifet, C. Castillo, P. A. Chirita, and I. Weber. An analysis of factors used in search engine ranking. In *Proc. First International Workshop on Adversarial Information Retrieval on the Web*, pages 48–57, 2005. http://airweb.cse.lehigh.edu/2005/proceedings.pdf.

[10] S. Blankson. *SEO. How to Optimize your Web Site for Internet Search Engines*. Blankson Enterprises Ltd, London, UK, 2008.

[11] J. Bollen, M. A. Rodriguez, and H. Van de Sompel. Journal status. *Scientometrics*, 69:669, 2006.

[12] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[13] N. Burani and W. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.

[14] L. Cai. Parameterized complexity of cardinality constrained optimization problems. *Comput. J.*, 51(1):102–121, 2008.

[15] F. Carreras and J. Freixas. Complete simple games. *Mathematical Social Sciences*, 32:139–155, 1996.

[16] K. Cechlárová and J. Hajduková. Computational complexity of stable partitions with B-preferences. *Int. J. Game Theory*, 31(3):353–364, 2002.

[17] K. Cechlárová and J. Hajduková. Stable partitions with $\mathcal{W}$-preferences. *Discrete Appl. Math.*, 138(3):333–347, 2004.

[18] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *Proc. 4th SIAM International Conference on Data Mining*, pages 442–446. SIAM, 2004.

[19] J. Cheetham, F. Dehne, A. Rau-chaplin, U. Stege, and P. J. Taillon. Solving large FPT problems on coarse grained parallel machines. *Journal of Computer and System Sciences*, 67:691–706, 2002.

[20] J. Chen and J. Meng. On parameterized intractability: Hardness and completeness. *Comput. J.*, 51(1):39–59, 2008.

[21] N. Chen and A. Rudra. Walrasian equilibrium: Hardness, approximations and tractable instances. In *Proc. Internet and Network Economics, First International Workshop, WINE 2005*, volume 3828 of *Lecture Notes in Computer Science*, pages 141–150. Springer, 2005.

[22] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with Google. *Informetrics*, 1:8, 2007.

[23] S. Chien, C. Dwork, R. Kumar, D. R. Simon, and D. Sivakumar. Link evolution: Analysis and algorithms. *Internet Mathematics*, 1(3):277–304, 2003.

[24] V. Conitzer and T. Sandholm. Complexity of determining nonemptiness of the core. In *Proc. 4th ACM Conference on Electronic Commerce (EC-2003)*, pages 230–231. ACM, 2003.

[25] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6-7):607–619, 2006.

[26] M. Cutts. PageRank sculpting. http://www.mattcutts.com/blog/pagerank-sculpting/ (retrieved June 2009), June 2009.

[27] K. Daskalakis and C. H. Papadimitriou. The complexity of games on highly regular graphs. In *Proc. 13th Annual European Symposium on Algorithms, ESA 2005*, volume 3669 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 2005.

[28] C. de Kerchove, L. Ninove, and P. van Dooren. Maximizing PageRank via outlinks. *Linear Algebra and its Applications*, 429(5–6):1254–1276, 2008.

[29] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[30] V. Deǐneko and G. Woeginger. On the dimension of simple monotonic games. *European Journal of Operational Research*, 170:315–318, 2006.

[31] R. Downey and M. Fellows. *Parameterized Complexity*. Springer, 1999.

[32] I. Elias. Settling the intractability of multiple alignment. Technical Report TRITA-NA-0316, Nada, KTH, 2003.

[33] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. Computational complexity of weighted threshold games. In *Proc. 22nd AAAI Conference on Artificial Intelligence*, pages 718–723. AAAI Press, 2007.

[34] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. On the dimensionality of voting games. In *Proc. 23rd AAAI Conference on Artificial Intelligence*, pages 69–74. AAAI Press, 2008.

[35] E. Elkind and D. Pasechnik. Computing the nucleolus of weighted voting games. In *SODA '09: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 327–335. SIAM, 2009.

[36] E. Enge. High-value link building is hard work. http://searchenginewatch.com/3631957 (retrieved June 2009), December 2008.

[37] M. P. Evans. Analysing Google rankings through search engine optimization data. *Internet Research*, 17:21–37, 2007.

[38] G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160. ACM Press, 2000.

[39] G. Flake, R. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1(4):385–408, 2004.

[40] J. Freixas and X. Molinero. Simple games and weighted games: A theoretical and computational viewpoint. *Discrete Applied Mathematics*, 157(7):1496–1508, April 2009.

[41] J. Freixas, X. Molinero, M. Olsen, and M. Serna. On the complexity of problems on simple games (*submitted*).

[42] J. Freixas and W. Zwicker. Weighted voting, abstention, and multiple levels of approval. *Social Choice and Welfare*, 21:399–431, 2003.

[43] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[44] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. 30th International Conference on Very Large Data Bases*, pages 576–587. Morgan Kaufmann, 2004.

[45] J. Hajdukova. On coalition formation games. Technical Report A5-2004, Institute of Mathematics, P.J. Safarik University, 2004.

[46] G. W. Harrison and T. McDaniel. Voting games and computational complexity. *Oxford Economic Papers*, 60(3):546–565, January 2008.

[47] D. A. Harville. *Matrix Algebra From a Statistician's Perspective*. Springer, 1997.

[48] T. H. Haveliwala. Topic-sensitive PageRank. In *Proc. 11th International Conference on World Wide Web, WWW 02*, pages 517–526. ACM Press, 2002.

[49] T. Hegedüs and N. Megiddo. On the geometric separability of boolean functions. *Discrete Applied Mathematics*, 66:205–218, 1996.

[50] M. Jackson. Link building, circa 2008. http://searchenginewatch.com/3631928 (retrieved June 2009), December 2008.

[51] M. O. Jackson and A. Bogomolnaia. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, Feb 2002.

[52] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[53] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, 1960.

[54] P. Kent. *Search Engine Optimization For Dummies*. Wiley, Indianapolis, USA, 2006.

[55] L. Khachiyan. A polynomial algorithm for linear programming. *Dokl. Akad. Nauk. SSSR*, 244:1093–1096, 1979. English Translation Soviet Math. Doklad. 20, pp. 191-194, 1979.

[56] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2005.

[57] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[58] A. N. Langville and C. D. Meyer. Updating Markov chains with an eye on Google's PageRank. *SIAM J. Matrix Anal. Appl.*, 27(4):968–987, 2006.

[59] J. L. Ledford. *Search Engine Optimization Bible.* Wiley, Indianapolis, USA, 2008.

[60] J. D. Leon Danon, Albert Díaz-Guilera and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[61] R. A. Malaga. Worst practices in search engine optimization. *Commun. ACM*, 51(12):147–150, 2008.

[62] Y. Matsui. A survey of algorithms for calculating power indices of weighted majority games. *J. Oper. Res. Soc. Japan*, 43:71–86, 2000.

[63] Y. Matsui and T. Matsui. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1-2):305–310, 2001.

[64] D. Mehta and V. Raghavan. Decision tree approximations of boolean functions. *Theoretical Computer Science*, 270(2):609–623, 2002.

[65] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14:265–294, 1978.

[66] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, New Jersey, USA, 1944.

[67] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[68] M. Olsen. Maximizing PageRank with new backlinks (*submitted*).

[69] M. Olsen. Nash stability in additively separable hedonic games is NP-hard. In *Proc. Computation and Logic in the Real World, Third Conference on Computability in Europe, CiE 2007*, volume 4497 of *Lecture Notes in Computer Science*, pages 598–605. Springer, 2007.

[70] M. Olsen. Communities in large networks: Identification and ranking. In *Proc. Algorithms and Models for the Web-Graph, 4th International Workshop, WAW 2006*, volume 4936 of *Lecture Notes in Computer Science*, pages 84–96. Springer, 2008.

[71] M. Olsen. The computational complexity of link building. In *Proc. Computing and Combinatorics, 14th Annual International Conference, COCOON 2008*, volume 5092 of *Lecture Notes in Computer Science*, pages 119–129. Springer, 2008.

[72] M. Olsen. Nash stability in additively separable hedonic games and community structures. *Theory of Computing Systems*, 45(4):917–925, 2009.

[73] M. Olsen and T. Viglas. MILP for link building (*in preparation*).

[74] P. R. Olsen. A future in directing online traffic. The New York Times, http://www.nytimes.com/2009/01/11/jobs/11starts.html, January 2009.

[75] G. Owen. *Game Theory*. Academic Press, San Diego, USA, third edition, 1995.

[76] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[77] C. Papadimitriou. *Computatorial Complexity*. Addison-Wesley, 1994.

[78] U. N. Peled and B. Simeone. Polynomial-time algorithms for regular set-covering and threshold synthesis. *Discrete Applied Mathematics*, 12:57–69, 1985.

[79] K. Prasad and J. S. Kelly. NP-completeness of some problems concerning voting games. *International Journal of Game Theory*, 19(1):1–9, March 1990.

[80] J. Rosenmüller. An algorithm for the construction of homogeneous games. In *Ökonomie und Mathematik*, pages 63–74. Springer, 1987.

[81] J. Smith. *Get into Bed with Google (Used the Danish translation)*. The Infinite Ideas Company Limited, Oxford, UK, 2008.

[82] S. C. Sung and D. Dimitrov. On core membership testing for hedonic coalition formation games. *Oper. Res. Lett.*, 35(2):155–158, 2007.

[83] A. Taylor and W. Zwicker. Simple games and magic squares. *Journal of combinatorial theory, Series A*, 71:67–68, 1995.

[84] A. Taylor and W. Zwicker. *Simple games: desirability relations, trading, and pseudoweightings*. Princeton University Press, New Jersey, USA, 1999.

[85] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proc. 14th International Conference on World Wide Web, WWW 2005 - Special interest tracks and posters*, pages 820–829. ACM, 2005.

[86] M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki. Coalitional games in open anonymous environments. *Transactions of Information Processing Society of Japan*, 47(5):1451–1462, 2006.