COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 47 (2023) e13330

ISSN: 1551-6709 online
DOI: 10.1111/cogs.13330

The Computational Challenges of Means Selection Problems: Network Structure of Goal Systems Predicts Human Performance

Daniel Reichman,[a,#] Falk Lieder,[b,#] David D. Bourgin,[c] Nimrod Talmon,[d] Thomas L. Griffiths[e]

[a]*Department of Computer Science, Worcester Polytechnic Institute*
[b]*Department of Psychology, University of California, Los Angeles*
[c]*Adobe Research*
[d]*Department of Industrial Engineering and Management, Ben-Gurion University*
[e]*Department of Computer Science and Department of Psychology, Princeton University*

Received 17 March 2022; received in revised form 3 July 2023; accepted 25 July 2023

Abstract

We study human performance in two classical NP-hard optimization problems: Set Cover and Maximum Coverage. We suggest that Set Cover and Max Coverage are related to means selection problems that arise in human problem-solving and in pursuing multiple goals: The relationship between goals and means is expressed as a bipartite graph where edges between means and goals indicate which means can be used to achieve which goals. While these problems are believed to be computationally intractable in general, they become more tractable when the structure of the network resembles a tree. Thus, our main prediction is that people should perform better with goal systems that are more tree-like. We report three behavioral experiments which confirm this prediction. Our results suggest that combinatorial parameters that are instrumental to algorithm design can also be useful for understanding when and why people struggle to choose between multiple means to achieve multiple goals.

Keywords: Goal systems; Computational complexity; Bounded rationality; Graph theory

[#]Equal contribution.
A preliminary version of our findings about the effect of tree-width on means-selection appeared in the Proceedings of the 39th Annual Meeting of the Cognitive Science Society under the title: The Structure of Goal Systems Predicts Human Performance.

Correspondence should be sent to Daniel Reichman, Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, 01609, USA. E-mail: daniel.reichman@gmail.com

1. Introduction

We study human performance in solving algorithmic problems arising from networks of goals and means interconnected to one another. Using goals, means, and their connections to understand cognition and problem-solving dates to the work of Newell and Simon (1972). Thereafter, established motivational theories of goal pursuit suggested that people's goals and the means available to pursue them are organized into a network known as their *goal system* (Kruglanski et al., 2002). A goal system is a *graph* where each vertex is a goal or a means and an edge connecting a means to a goal indicates that the means corresponding to one endpoint of the edge can be used to achieve the goal on the other endpoint.

Following goal systems theory, we identify computational problems that arise when people try to efficiently manage their goal systems. We consider two specific means-selection problems related to goal systems: Selecting means to achieve the largest possible number of goals with a limited number of means (Max-Coverage, or MC in short) or to achieve all goals with as few means as possible (Set Cover, or SC in short). These problems are known to be computationally intractable, or NP-hard in the language of computer science (Garey & Johnson, 1979; Kleinberg & Tardos, 2005)[1] and this intractability might be part of the reason why people often find it challenging to manage multiple goals.

Based on the principle that psychological processes are often characterized by underlying *efficient* algorithms (Griffiths, Lieder, & Goodman, 2015; Sanborn, Griffiths, & Navarro, 2010), our main idea is that graph-theoretic properties that are used by computer scientists to determine the difficulty of instances of MC and SC should also be instrumental in quantifying how difficult it will be for people to solve MC and SC instances as well. Therefore, it should be possible to predict how well people can solve problems such as MC and SC given a network of goals and means from the graph-theoretic properties of the network.

Correspondingly, here we study how well structural properties of people's goal systems can predict the success or failure of selecting means effectively by focusing on the similarity of a graph to a tree. The rationale is that many NP-hard optimization problems over graphs often become easier when the graph structure is *tree*-like (Kleinberg & Tardos, 2005, Kloks, 1994; Robertson & Seymour, 1986) where the term *tree* is defined as a connected graph *without* cycles. Our main measure of similarity to a tree is the *treewidth* of a network[2] (Kloks, 1994; Robertson & Seymour, 1986), where the *smaller* the treewidth is, the more similar the network is to a tree. There is vast evidence that many optimization problems on networks that are "far" from trees ("large" treewidth) are likely to be hard to solve exactly or approximately (Clementi & Trevisan, 1999; Marx, 2007; Trevisan, 2001). Based on this connection between graph properties and algorithmic hardness, we hypothesis that *goal systems that are arranged in a network that is "tree-like" will be easier for people to deal with and enable them to perform better*.

We rigorously test this prediction in three experiments where people are required to solve MC and SC instances on goal systems that are presented to them. Our empirical findings lend strong support to our prediction that the more tree-like networks are, the easier they are for people leading to superior performance in MC and SC. Importantly, these findings *cannot* be attributed to the size of the networks as we take special care to manipulate the tree-likeness of networks without changing the size of the network (all networks in our experiments have the *same number* of nodes and edges). Furthermore, the solution spaces of the different networks are similar in magnitude and we show they do not account for the effect of tree-likeness on performance in MC and SC. These findings offer a fresh computational perspective on why people find it difficult to achieve their goals and consist of one of the first empirical examinations whether combinatorial parameters associated with NP-hard problems can be used in a meaningful way to predict how difficult (or easy) it will be for people to cope with these problems. Our theorizing is *computational* in nature and is qualitatively different from a myriad of *motivational* explanations for difficulties people experience when pursuing multiple goals (Ainslie & Haslam, 1992; Ariely & Wertenbroch, 2002; Bandura, 1997; Baumeister & Heatherton, 1996; Carver & Scheier, 1998; Gollwitzer, 1999; Green, Fristoe, & Myerson, 1994; Kruglanski et al., 2012; Liberman & Dar, 2009; Mischel, 1974; Myerson & Green, 1995; Stajkovic & Luthans, 1998; Steel, 2007).

Our experimental results align well with theoretical knowledge from computer science and combinatorial optimization and support the relevance of findings from algorithm design and computational complexity to cognitive science (van Rooij, Blokpoel, Kwisthout, & Wareham, 2019).

This paper is organized as follows: We begin by formalizing goal systems and the computational problems entailed by networks of goals and means. In the subsequent sections, we derive and empirically confirm the predictions that people's performance at deciding which means to employ to achieve a set of goals critically depends on the tree-likeness of the goal system: We close by summarizing our findings and suggesting directions for future research.

2. Computational problems arising from pursuing multiple goals

People often simultaneously pursue multiple goals with hierarchically nested subgoals (Little & Gee, 2007; Newell, Shaw, & Simon, 1958). In line with this observation, previous psychological research has suggested that people mentally represent their goals, subgoals, and the means available to attain them as nodes in interconnected networks known as "goal systems" (Fishbach, Shah, & Kruglanski, 2004; Huang & Zhang, 2013; Kruglanski et al., 2002; Louro, Pieters, & Zeelenberg, 2007; Shah & Kruglanski, 2000). In this section, we formalize goal systems and the computational problems they entail using graph theory.

2.1. Goal systems

Using a graph-theoretic formalism,[3] we can describe a *goal system* as a bipartite graph with a vertex set (M, G), where $M = \{m_1, \ldots, m_k\}$is a set of means and $G = \{g_1, \ldots, g_l\}$ is a set of goals. We will generally assume that at least *one* means connected to a given goal must be acquired to achieve the goal. Finally, each goal (means) can be assigned a nonnegative number that measures how desirable the goal is (how costly the means is). Here, we shall assume all goals (means) have the same value (cost) which equals 1. For an ilusration please see Figure 1.

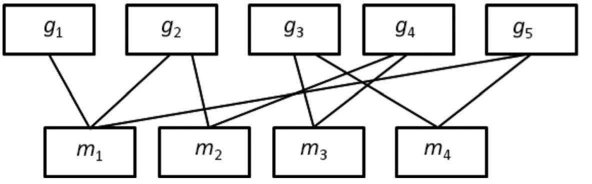


Fig. 1. Example of goal system: Means appear at the bottom marked by $m_1, \ldots, m_4$. Goals appear on top marked by $g_1, \ldots, g_5$. A goal can be attained if at least one means connected to it is chosen. The set $\{m_1, m_3\}$ is a minimal set of means covering all goals.

It is also possible to consider more complicated weighted models, where edges have weights, and a goal is achieved if the weighted sum of chosen means attached to it exceeds a given threshold. One can also consider models where different goals or means have different values or costs. To simplify the analysis and the experiments, we will not deal with such weighted models here.

Goal systems give rise to means-selection problems where one seeks to choose a set of means that are instrumental to the ends one is trying to achieve (Kruglanski et al., 2002). Such choices are usually constrained by limited resources, such as time and money. For example, it is often desirable to find the set of means that attains one's goals with the least amount of time, money, and effort possible (Shenhav et al., 2017). In other situations, people are constrained by a limited amount of time or energy and aim to achieve as many of their goals as possible. The challenge then is to select a set of means that is feasible (does not exceed the constrained resource) and achieves as many goals as possible. For example, suppose that you visit London for an academic conference and would like to explore the various attractions of the city. Getting a hotel in a good and central location is one means toward achieving this goal. Another means could be to buy a public transport ticket for the entire visit. A third means could be to rent a bicycle, and a fourth could be to take one of the many cruises on the Thames. You can also take an organized tour. Acquiring all these means would be too costly. Instead, you need to find a clever combination of a small number of suitable means.

Below, we formalize these two computational challenges in terms of well-defined computational problems which are known to be NP-hard.

2.1.1. *Computational problems*

We focus on the following computational problems that arise in goal pursuit.

Maximum Coverage. The first computational problem we consider is trying to achieve as many goals as possible with a fixed budget that limits the number of activities that one can perform, corresponding to the number of means one can utilize. We formalize this challenge in terms of the Maximum Coverage (MC) problem. In the MC problem, we are given a bipartite graph (C, D, F), where C is one side of the bipartition, D is the second part, and F is a set of edges where each edge connects a vertex from C to a vertex in D. We are also given a nonnegative integer $k \leq |C|$. We seek to find a set $C' \subseteq C$, of cardinality at most k, maximizing the total value of vertices in D covered by vertices in C' (a node $d \in D$ is covered by a node

$c \in C'$ if the edge (c,d) belongs to F). Recall that we assume that covering a goal by a *single* means is sufficient for that goal to be achieved.

Set Cover. The second problem that we study is achieving a given set of goals as efficiently as possible by selecting a set of means of minimal size that will accomplish *all* goals. We formalize this challenge in terms of the SC problem. In the SC problem, we are given a bipartite graph $G = (A, B, E)$ where A, B are the sides of the bipartition and E is the set of edges connecting vertices in A (means) to vertices in B (goals). Our goal is to cover *all* vertices in B by a set $A' \subseteq A$ of minimal cardinality. As before, a node b in B is covered by a subset A' of A if there is a node in A' adjacent to b.

The two optimization problems defined above (i.e., MC and SC) are in alignment with previous theoretical work on goal pursuit. The assumption that one seeks to maximize the total value of attained goals is in line with expectancy-value theory originating from utility theory in economics as well as goal system theory (Kruglanski et al., 2002; Kruglanski & Kopetz, 2009, 2010; Vroom, 1964). That managing multiple goals has to occur under limited resources is widely acknowledged as well (Kahneman, 1973; Kruglanski et al., 2002). Finally, representing goals and means as graphs and assuming interconnectedness between goals and means appear either implicitly or explicitly in several settings (Carver & Scheier, 1998; Kruglanski et al., 2002; Louro et al., 2007). It is worth noting that, while it is suggested that people seek to *maximize* the value of goals they attain (Kruglanski & Kopetz, 2009, 2010), hardly any work has addressed the properties of goal systems that allow people to maximize despite worst-case intractability results.

2.2. *NP-hardness and prediction of human performance*

Here, we consider a few additional technical issues related to using the NP-hardness of optimization problems to predict how people will perform on these problems. The reader wishing to avoid these technicalities can skip this section with no loss of continuity.

Roughly speaking, a computational problem is said to be *in* NP if there is an efficient (polynomial-time) algorithm that certifies the feasibility of a proposed solution. For example, there is an efficient algorithm that can attest to the fact that a collection of nodes is indeed a set cover (i.e., indeed covers all goals) for an SC instance, and thus, the computational problem of determining whether an SC instance has a solution of size t (for an integer t) is in NP. Informally, a problem is NP-hard if an efficient algorithm for it entails that every problem in NP has an efficient algorithm. It is widely agreed among computer scientists that NP-hard problems are very likely to be intractable and not admit a polynomial-time algorithm (e.g., Garey & Johnson, 1979; Kleinberg & Tardos, 2005; Papadimitriou, 1994). Since it is also agreed that the nonexistence of polynomial-time algorithms entails computational intractability, computer scientists usually believe that NP-hard problems are computationally intractable, in the sense that no efficient algorithms for them exist (notice that the claim is not merely that such efficient algorithms are currently not known, but indeed that such algorithms do not exist thus cannot be found). Both MC and SC are known to be NP-hard (Garey & Johnson, 1979).

The idea that NP-hardness constrains people's ability to effectively deal with goal systems appears intuitively plausible and is in line with theoretical reasoning in psychology and cognitive science. Indeed, many computational tasks that arise in perception, decision-making, problem-solving, reasoning, and other domains (e.g., Bossaerts & Murawski, 2017; Gigerenzer, 2008; MacGregor & Ormerod, 1996; van Rooij, 2008; van Rooij, Stege, & Kadlec, 2005; van Rooij & Wareham, 2012; van Rooij, Wright, & Wareham, 2012; Thagard & Verbeurgt, 1998) are NP-hard, and the general consensus is that people are unable to find optimal solutions to NP-hard problems. Nevertheless, using NP-hardness results to explain the difficulties people may experience in dealing with goal systems is not as straightforward as it may appear. There are several limitations of notions from complexity theory (such as NP-hardness) that limit their use toward predicting the performance of both minds and machines. First, NP-hardness results from computational complexity theory apply to algorithms that solve the problem on every possible input. Therefore, it is not clear whether these results constrain human problem-solving and goal-pursuit on "real-life" instances that may exclude pathological worst-case instances. Furthermore, polynomial-time algorithms may be too complicated as well as time- and space-consuming for people to implement. Hence, it could well be that "floor effects" would emerge in the sense that treewidth would have little influence on human performance due to poor performance on instances with low treewidth. These caveats of complexity theory have led to increasing reliance on empirical examinations in computer science when determining factors that influence how hard an instance of an NP-hard problem is likely to be to solve (Leyton-Brown, Nudelman, & Shoham, 2009; Hutter, Xu, Hoos, & Leyton-Brown, 2014). Therefore, we believe that such an empirical approach is needed when evaluating how people cope with NP-hard problems and whether theoretical results from complexity theory can be used to predict human behavior. Moreover, there is a large body of theoretical work connecting computational complexity to cognitive science. Empirical validation is necessary in the theory building of human cognition that relies on these theoretical works. Hence, our empirical results supporting predictions from computational complexity theory with respect to human behavior are significant and do not obviously follow from computational complexity theory.

An empirical examination of the connection between NP-hardness and human performance is not straightforward. One difficulty is that NP-hardness is a property of an algorithmic problem, rather than individual instances of a problem. While some instances of an NP-hard problem may be difficult for people, other instances may be easy for them. For example, the observation that MC is NP-hard is of little use if we want to determine whether two MC instances (whose underlying goal systems have the same number of goals and means) are likely to differ significantly in the number of computational steps that are needed to solve them. To overcome this difficulty, we draw on the rich literature within theoretical computer science, artificial intelligence, and combinatorial optimization that identifies structural properties of individual[4] instances of NP-hard problems that can render them easy to solve (Kleinberg & Tardos, 2005, Leyton-Brown et al., 2009; Hutter et al., 2014). We use this literature to derive concrete predictions about people's performance at selecting the right means to achieve goals effectively and test them in dedicated experiments. We observe that there appears to be scarce empirical evidence validating the utility of computational complexity theory and algorithm design in explaining human performance and behavior (but see Enemark, McCubbins, Paturi, & Weller, 2011). While there have been empirical efforts to examine which strategies people use in solving NP-hard problems (e.g., Carruthers, Masson, & Stege, 2012; Dworkin & Kearns, 2015; MacGregor, Ormerod, & Chronicle, 2000), these efforts do not aim at validating or refuting concrete predictions regarding what features of such problems make them hard for people.

Two possible additional measures of the complexity of an instance of an optimization problem are either the *size* of the solution space (the number of possible solutions) and the *number of paths* in the solution space that lead from the empty set to an optimal solution. Here, we endow the solution space with an undirected graph where two solutions which are subsets A,B of means vertices of the MC or SC instance are connected in the solution space if B contains A and $|B| = |A| + 1$. The idea is that a path in the solution space represents a sequence of actions whose execution yields a solution to the MC or SC instance.

These complexity measures originate from the human problem-solving literature (Campbell, 1988; Newell & Simons, 1972), where it is suggested that the larger they are, the harder the instance is. Such measures are typically exponential in the number of nodes of the goal system (even for instances that admit polynomial-time algorithms and even for trees). For this reason, these measures are considered by theoretical computer scientists to be less useful in differentiating between hard and easy problems. Here, we focus on other predictors of problem difficulty in our theorizing although for completeness, we also include the number of paths to an optimal solution as a predictor.

In summary: Both MC and SC are NP-hard. The hardness of these problems gives a first indication of why people might find it so difficult to juggle multiple goals at the same time. To devise testable hypotheses as to which instances will be harder for people, we use combinatorial parameters that assign to each individual instance of MC or SC a real number which is then used to predict human performance.

2.3. Performance metrics

People's performance at solving optimization problems can be measured in multiple ways. Our first performance measure simply quantifies the ability to find the optimal solution. This is a binary measure that equals 1 if the person finds an optimal solution and 0 otherwise. For SC, this means finding a cover of minimal size and for MC, this means finding a set of k means maximizing the covered goals.

Our second measure for maximization problems is the *approximation ratio* achieved on the instance of the problem, namely, the solution value divided by the optimal (maximal) value (for minimization problems, the approximation ratio is the optimal value divided by the solution value). More formally, for an MC instance, let OPT_{MC} to be the maximum number of means that are covered by a set of means of cardinality k. Let SOL_{MC} be the value of goals covered by a set of means chosen by a person that solved an MC instance. In this case, the approximation ratio is $SOL_{\mathrm{MC}}/ OPT_{\mathrm{MC}}$. For SC, let OPT_{SC} be the size of a Set Cover of minimal cardinality and let SOL_{SC} be the size of a solution returned by a person solving SC. Then, the approximation ratio is $OPT_{\mathrm{SC}}/ SOL_{\mathrm{SC}}$. Observe that the approximation ratio is always nonnegative and at most 1 and that the closer the approximation ratio to 1, the better the performance is considered.

For both MC and SC, we shall also refer to the approximation ratio as the *solution quality*. The subscripts will be omitted when the optimization problem is clear from the context. The approximation ratio is widely used to analyze the quality of approximation algorithms for NP-hard problems (Vazirani, 2003).

In the remainder of this paper, we apply computational complexity theory to predict human performance in means-selection problems (specifically, MC and SC) and evaluate these predictions empirically. Our approach draws inspiration from empirical hardness models in computer science (e.g., Hutter et al., 2014; Leyton-Brown et al., 2009) which leverage tools from machine learning and statistics to predict the performance of algorithms. Concretely, the following two sections evaluate whether graph-theoretic measures of tree-likeness as well as multi- and equi-finality can predict the performance of the cognitive strategies people use to decide which means to employ to achieve a given set of goals.

3. The tree-likeness of a goal system predicts performance

In this section, we derive, test, and confirm the complexity-theoretic prediction that people should perform better at means-selection when the goal system is tree-like. We start by introducing suitable graph-theoretic measures of tree-likeness.

3.1. Measures of tree-likeness

Recall that a *tree* is a *connected* network that has *no cycles*. Intuitively, a tree has the minimal number of edges that ensure connectivity: the lack of cycles implies that removing even a single edge will result in a disconnected graph. Here, we describe several measures of tree-likeness, namely, treewidth, combinatorial expansion, spectral expansion, and others. Each of these is a well-known measure quantifying different aspects of measuring closeness to a tree of a given graph. Our main predictor is treewidth and the other predictors can be glossed upon in a first reading of the experimental results. We study additional predictors for exploratory reasons: As our data are one of the first to study the performance of people on a variety of graphs, we reason it is of value to relate multiple predictors to human performance.

3.1.1. Treewidth

When restricted to trees, many NP-hard problems can be solved by efficient polynomial-time algorithms, such as divide-and-conquer methods and greedy algorithms (Cormen, Leiserson, Rivest, & Stein, 2001). Treewidth (Fig. 2) is a combinatorial parameter, which is always a nonnegative integer, that can be associated with any given graph. Intuitively, a smaller treewidth value implies that the nodes and edges of the network can be arranged in a way that resembles a tree (e.g., Kloks, 1994; Robertson & Seymour, 1986). For an exact definition of treewidth, see Appendix A.

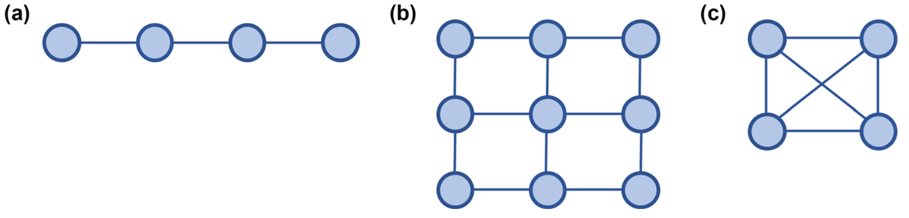


Fig. 2. *Treewidth*. Some examples of graphs and their treewidths. (A) Trees (upper left) have treewidth 1. (B) Square grids of side length d have treewidth d, hence this graph has treewidth 3. (C) Complete graphs of order n where every two pairs of nodes are connected have treewidth $n-1$. Hence, the treewidth of this graph is 3.

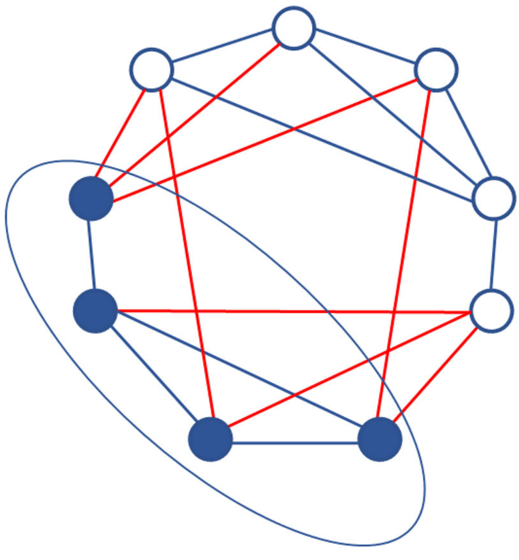


Fig. 3. The edge expansion of a graph $G=(V,E)$ is the minimal edge expansion over all of its vertex subsets whose size is at most $|V|/2$. Bounded degree trees have small edge expansion. In this figure, we consider the circled subset S of four vertices in a 4-regular graph. Edges that connect vertices in S to vertices not belonging to S are highlighted in red. The edge expansion of S is 8/4=2, since there are eight edges "going out" of S, and we normalize by the size of S.

It is known that a connected graph has treewidth 1 if and only if it is a tree (e.g., Kleinberg & Tardos, 2005) and that, for every $d \geq 3$, there are d-regular graphs (a d-regular graph is a graph where each vertex has exactly d edges attached to it) that have treewidth of size cn where n is the number of vertices and c is some fixed positive constant (Perarnau & Serra, 2014).

3.1.2. Combinatorial expansion

Intuitively, a given set S is *expanding* if "most" of the edges incident to S touch vertices outside S. See Fig. 3 for an illustration. The edge (vertex) expansion quantifies the magnitude of edge (vertex) expansion of the graph by taking the "worst" expanding subset of vertices that is not too large. For a precise definition, see Appendix A.

Trees of bounded degree have the smallest possible expansion among connected graphs (of order $1/|V|$, where $|V|$ is the number of nodes in the network), hence large expansion suggests that the graph is dissimilar to a tree.

3.1.3. Spectral gap

This measure of expansion is a real number that is positively correlated with tree-likeness (a larger value implies greater similarity to a tree). The spectral gap is d-λ_2 where d is the degree of the graph (we assume all vertices have the same degree) and λ_2 is the second largest eigenvalue in absolute value of the adjacency matrix of the graph. The spectral gap can be calculated efficiently. For a precise definition, see Appendix A.

While treewidth, expansion, and the spectral gap are likely to be positively correlated, there are differences between these predictors and thus we examined all of them. For example, the various expansion measures can be drastically affected by a single "bottleneck" (e.g., a graph composed of two connected components with no edges between them). This suggests a weakness of expansion-based predictors, as a graph might be composed of two disconnected "hard" graphs and nonetheless have low expansion. Treewidth is a more robust measure: for example, a union of two graphs with high treewidth that are disconnected (or connected by a small number of edges) results in a graph of large treewidth. The spectral gap admits an efficient polynomial algorithm as opposed to the other predictors which are NP-hard making it more practical to use.

3.1.4. Additional predictors

Recall that a forest is a graph with no cycles. A feedback vertex set (FVS) is a subset of vertices whose deletion from a given graph results in a forest. Alternatively, it is a set of vertices S such that every cycle in the graph contains at least one vertex from S. The size of a minimal FVS is an alternative measure to the similarity of a graph to a tree. Hence, we used this feature as well.[5] Previous empirical hardness models have found additional features of graphs to be useful: the diameter, the average eccentricity, and the average path length (Leyton-Brown et al., 2009). We thus included these features in our analysis as well. The diameter of a graph is the longest distance between two vertices in the graph (where the distance is the number of edges in a shortest path connecting two vertices; all graphs in this work are connected). The eccentricity $\varepsilon(v)$ of a vertex v in an undirected graph $G = (V,E)$ is the maximal distance of a vertex in V from v. The average eccentricity (AvgEcc) is defined as $\frac{1}{|V|}\sum_{v\in V}\varepsilon(v)$. The average path length (AvgPath) is $\frac{2}{|V||(V|-1)}\sum_{u,v\in V,u\neq v} d(u, v)$ where $d(u,v)$ is the distance between u and v (and every pair of vertices is counted in the summation once). Finally, we computed the number of paths in the *solution space* that lead to an optimal solution in both SC and MC. Here, a path to an optimal solution S is a sequence of subsets $A_0, A_1, A_2 \ldots A_r$ such that A_0 is the empty set and $A_r = S$ and for every $i<r$, A_{i+1} contains exactly one additional vertex not belonging to A_i. In other words, such a path is simply an increasing sequence of subsets

obtained by adding one vertex at a time starting with the empty set and ending in an optimal solution. We denote by NumOptimalPathsSC, (NumOptimalPathsMC) this number for an SC (MC) instance. For SC, we also consider the normalized version where NumOptimalPathsSC is divided by the total number of paths of length r that do not necessarily end in an optimal solution. We call this variable PropOptimalPathsSC. An analogous normalized quantity for MC (where the total number of paths includes those that do not end in an optimal coverage) is denoted by PropOptimalPathsMC.

3.2. *The connection between treelikeness and efficient algorithms*

Treewidth has found numerous algorithmic applications (Kloks, 1994; Robertson & Seymour, 1986). The crucial observation is that for many algorithmic problems, an n-vertex graph of treewidth w admits an exact divide-and-conquer-based algorithm that solves the problem in time proportional to $2^w n$ (Bodlaender, 1988; Cygan et al., 2015; Downey & Fellows, 2013). Both MC and SC are no exception: An algorithm with running time $O(2^w n)$ is known for n-vertex bipartite graphs of treewidth w (Alber & Niedermeier, 2002). On the other hand, graphs with large treewidth appear to be difficult to solve: For some problems, the exponential dependency of the running time on the treewidth w (for algorithms which find the optimal solution) is inherent under certain assumptions (Marx, 2007). Worst-case instances of MC and SC are hard even to approximate (Feige, 1998), and hard instances often have large treewidth (Clementi & Trevisan, 1999; Trevisan, 2001). Therefore, it is likely that finding nearly optimal solutions in instances of large treewidth will be intractable.

Intuitively, a tree-like goal system is one in which goals can be pursued nearly independently of each other making the selection of means a task that can be decided locally which results in an easier problem. In contrast, in goal systems that are *not* tree-like, there is a stronger connection between goals and the means chosen to satisfy them making means-selection a global problem, which is more difficult. As noted above, these intuitions can be made precise: Combinatorial optimization problems on graphs that are more tree-like admit more efficient algorithms.

Current algorithms that compute tree-decompositions are quite complicated and time-consuming. Therefore, it seems unlikely that people would use them to solve SC and MC problems. But treewidth might nevertheless affect the performance of people's heuristics: The similarity between low treewidth graphs and trees might make the kinds of algorithms that people use, such as greedy methods, much more effective. Similar reasoning applies to our expansion measures. For example, NP-hard problems over graph families which are guaranteed to have a sublinear edge or vertex separators (e.g., planar graphs, graphs excluding a fixed minor) are known to have better approximation algorithms compared to arbitrary instances (e.g., Baker, 1994; Klein, Plotkin, & Rao, 1993). Summarizing the discussion above, we derive the following hypothesis:

Hypothesis I. Treewidth, vertex-expansion, edge-expansion, and the spectral gap of G should be negatively correlated with the solution quality and probability that a person will find the optimal solution to the SC and MC problems.

3.3. Design of treewidth experiments

To test how our predictors can capture human performance in MC and SC, we conducted a series of behavioral experiments. To test whether human performance in means-selection deteriorates with increasing treewidth (Hypothesis I), we used a between-subjects design, where each participant was assigned randomly to 1 of 20 graphs. In Experiment 1, participants were asked to solve the SC problem, while in Experiment 2, participants were asked to solve the MC problem. In each case, the problem was graphically represented as a bipartite graph. To avoid potential influence of the problem size on the performance in these tasks, we used graphs with an identical number of vertices (48), and edges across all experiments. All graphs were 4-regular and thus had exactly 96 edges. The 24 vertices at the bottom represented the available means (activities A-Z) and 24 vertices at the top represented the goals. Each edge from a means vertex to a goal vertex implies that completing that activity is sufficient to achieve the goal. In the SC task, participants were asked to select a minimal number of activities (means) to achieve all goals. In the MC task, participants were asked to choose five activities that achieve as many goals as possible. Goals often have semantic content of desirable outcomes. Experiment 3 asked participants to solve an SC problem where goals are given semantic content and real values. Furthermore, a different visual display was used to eliminate visualization effects. The same family of bipartite graphs as in the first two experiments was used in Experiment 3. For a visual display of all 20 graphs, please see the Appendix.

Our experiments follow a common paradigm where people must choose a set of means to achieve a set of goals coupled with a verbal description of some of the possible means or goals. While this paradigm clearly does not capture all aspects of goal pursuit in the wild (which are often significantly more complex), it is similar to previous experiments done within the goal systems framework (Köpetz, Faber, Fishbach, & Kruglanski, 2011; Huang & Zhang, 2013).

All experiments reported below were approved by the institutional review board of the University in North America where the experiment was conducted.

3.4. Experiment 1: Human performance on SC

3.4.1. Methods

3.4.1.1. Participants: We recruited 655 participants on Amazon Mechanical Turk. Participants were paid \$1.25 and could earn a performance-dependent bonus of up to \$2.

3.4.1.2. Stimuli and procedure: We generated 20 graphs whose treewidth ranged from 5 to 14. Besides one random 4-regular bipartite graph, a 6×8 torus, and a 4×12 torus, we generated three families of graphs. For each of these families, we first generated a *base* graph of low treewidth by examining plausible candidate graphs and computing their treewidth; for two of the families, we used a graph of treewidth 6, and for the third family, we used a graph of treewidth 5. Then, we used two methods which take as input those base graphs and generate graphs of varying higher treewidth. The first method operated by performing random flips: we randomly choose two edges (a, b) and (c,d) for which we verify that neither (a,c) nor (b,d) already belongs to the graph on which the flip is to be performed. Then, we delete the edges (a,b) and (c,d) and replace them by the edges (a,c) and (b,d). By increasing the number of these random flips, we could gradually increase the treewidth. The second method identified a matching of size 8 in the graph and reconnecting the matched vertices by a random matching. Again, by repeating this process, we could gradually increase the treewidth. Each participant was randomly assigned to 1 of 20 conditions that differed only in the graph structure of the SC problem participants were asked to solve. Please see Appendix B for a visual display of the 20 graphs we have used. After consenting to participate, participants read a cover story about a person trying to choose which set of activities (e.g., volunteer to improve the company's website and work out at the gym) they should perform in order to achieve all their goals (e.g., earn more money, improve relationship with boss, get fit, etc.) with as few activities as possible because their time is limited. The story highlighted that some activities achieve multiple goals at the same time. Next, participants completed a simple practice trial involving only two means and two goals. Once they had solved this task successfully, participants proceeded to the SC problem they had been assigned to (see Fig. 4).

When participants hovered with their mouse over an activity, the interface highlighted the goals this activity would achieve and the corresponding edges of the graph in green. Goals that had already been achieved and activities that had already been selected were highlighted by checkmarks, and the number of selected activities was shown at the bottom of the screen. Participants were asked to help a person achieve their 24 goals with as few activities as possible, and they were motivated by the prospect of earning a financial bonus of $2 for achieving all goals as efficiently as possible. After submitting their solution, participants completed an exit questionnaire asking them for basic demographic information (age, gender, and primary language), and four 9-point Likert scales (anchors: "not at all," "somewhat," and "extremely") measuring the perceived difficulty of the task, motivation to achieve all goals, motivation to find the optimal solution, and their motivation to finish the task as quickly as possible.

3.4.2. Results

We excluded 30 participants (4.6%) because their responses did not achieve all goals, suggesting that they did not follow the instructions.

We found that treewidth alone explained 44.90% of the variance in the frequency with which people found the optimal solution across the 20 graphs ($F(1, 18) = 14.20$; $p = .0012$): As we increased the treewidth of the graph, the percentage of participants who discovered the optimal solution decreased significantly ($\rho = -0.59$; $p = .0058$) from more than 90% on the graph with treewidth 5 to only about 30% on the graph with treewidth 14 (Fig. 5A). We found that the average solution quality was negatively correlated with treewidth ($\rho = -0.44$; $p = .0525$), suggesting that our participants achieved fewer goals for goal systems with higher treewidth (Fig. 5B). Treewidth explained 17.59% of the variance in the median response time across problems ($F(1, 18) = 3.86$; $p = .0650$): The median amount of time people took to solve the problems tended to increase with treewidth ($\rho = 0.3426$) but this effect was not statistically significant ($p = .1393$), and when we restricted this analysis to correct solutions, the correlation was $\rho = 0.3825$ ($p = .1297$). Perceived difficulty also tended to increase with treewidth ($p = .37$) but this correlation was not statistically significant ($p = .1062$). Our participants were highly motivated to find the optimal solution (average rating 7.91 ± 0.06 out of 9). Thus, it appears unlikely that their motivation was a bottleneck to their performance.

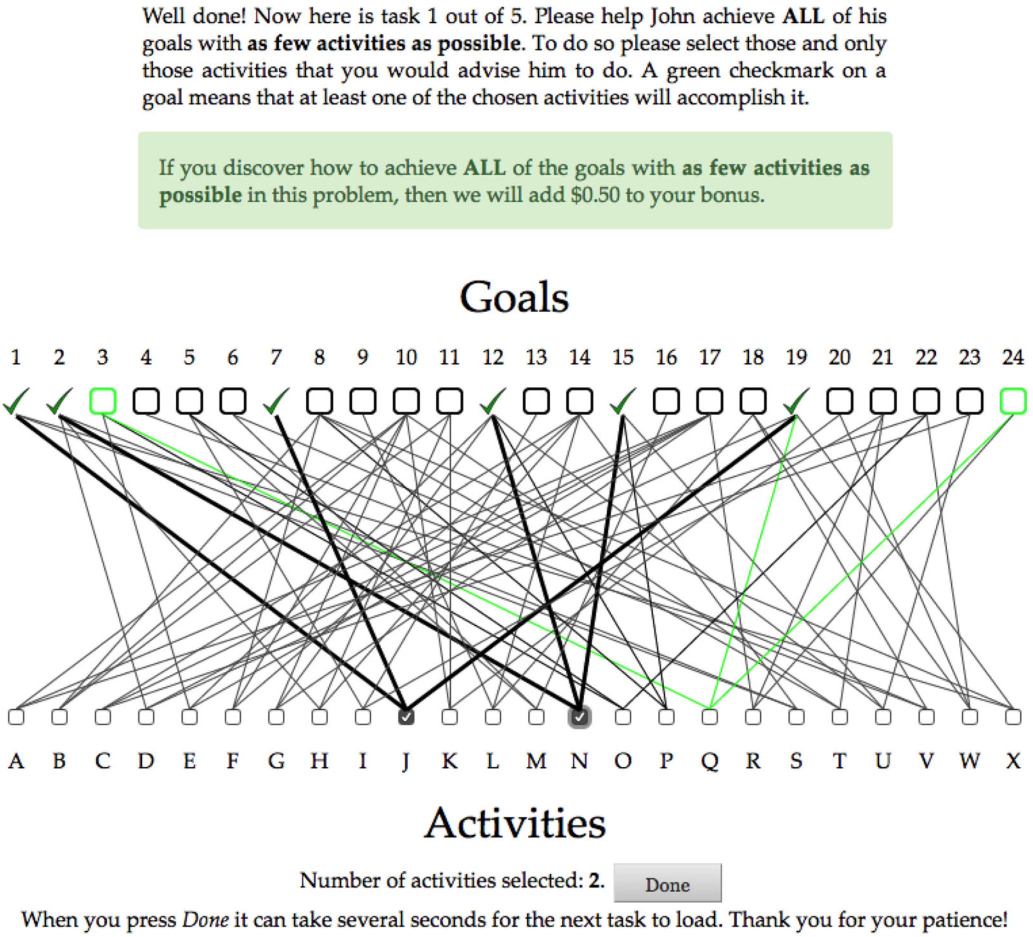


Fig. 4. Interface and *instructions* for Experiment 1.

Furthermore, motivation appeared to be unaffected by treewidth ($\rho = 0.23$; $p = .34$). Thus, the observed differences in performance appear to result from the inherent difficulties of the means-selection problems posed by different goal systems.

Of the additional predictors evaluated, we found that graph diameter, average shortest path, average graph eccentricity, the number of optimal paths, and the proportion of optimal paths all were significantly positively correlated with the frequency of optimal solutions identified by our participants (graph diameter: $\rho = 0.5691$; $p = .0088$, average shortest path: $\rho = 0.6516$; $p = .0019$, average eccentricity: $\rho = 0.6265$; $p = .0031$, number of optimal paths: $\rho = 0.5501$; $p = .0120$, proportion of optimal paths: $\rho = 0.5597$; $p = .0103$). In addition, the spectral gap (measured as d-λ_2) and vertex and edge expansions showed significant negative correlations with the frequency of optimal solutions (vertex expansion: $\rho = -0.5836$; $p = .0069$, edge expansion: $\rho = -0.4552$; $p = .0437$, spectral gap: $\rho = -0.6280$; $p = .0030$). We also found that the average shortest path, average graph eccentricity, spectral gap, and

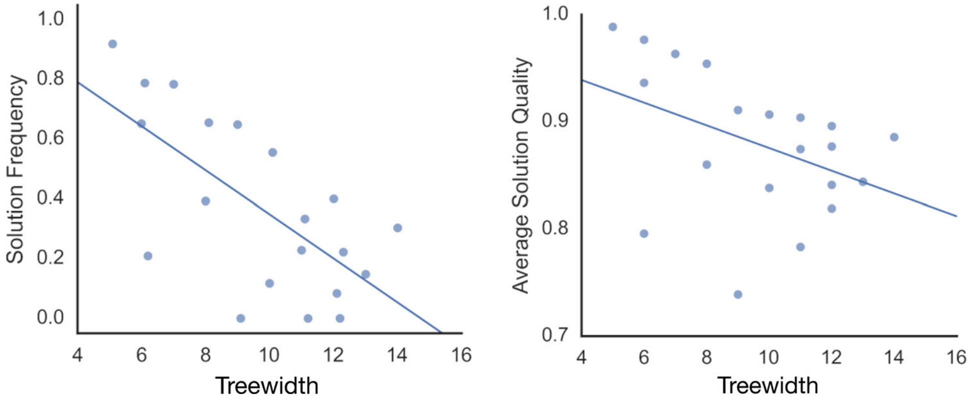


Fig. 5. Human performance in the set cover task of Experiment 1 as a function of treewidth. Left: Solution Frequency. Right: Solution Quality.

the graph vertex expansion were significantly correlated with the average participant solution qualities (average shortest path: $\rho = 0.4505$; $p = .0462$, average eccentricity: $\rho = 0.4316$; $p = .0574$, spectral gap: $\rho = 0.4496$; $p = .0467$, vertex expansion: $\rho = 0.4636$; $p = .0395$, number of optimal paths: $\rho = 0.4615$; $p = .0405$, proportion of optimal paths: $\rho = 0.4705$; $p = .0363$). Finally, only the graph edge expansions exhibited a significant correlation with the median response times on the SC task ($\rho = 0.4538$; $p = .0445$).

None of the effects reported above can be attributed to variability in the size of the optimal solution, since it was neither correlated with the average frequency with which people found the optimal solution ($\rho = -0.0667$, $p = .6827$) nor with the quality of their solutions ($\rho = 0.1356$, $p = .4043$).

3.5. Experiment 2: Human performance on MC

3.5.1. Methods

3.5.1.1. Participants: We recruited 545 participants on Amazon Mechanical Turk.

Participants were paid \$1.25 and could earn a bonus of up to \$2. The consent form specified that participants must not have participated in the previous version of this experiment.

3.5.1.2. Stimuli and procedure: Experiment 2 was identical to Experiment 1 except for the task: Participants were now instructed to achieve as many goals as possible subject to the constraint that the person's limited time does not permit them to complete more than five activities. The 20 graphs and financial incentives were the same as in Experiment 1. The interface of Experiment 1 (see Fig. 5) was modified to prevent participants from selecting more than five activities at a time. When a participant attempted to add a sixth activity, they were told they would first have to remove one or more of the activities they had already selected. The cover story and survey were modified slightly to match the change in the task.

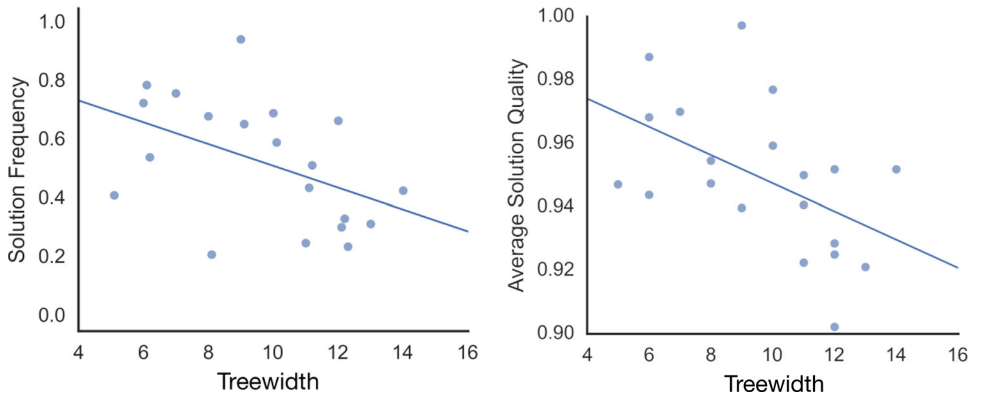


Fig. 6. Human performance in the Maximum Coverage task of Experiment 2 as a function of treewidth. Left: Solution Frequency. Right: Solution Quality.

3.5.2. Results

We excluded 23 participants (4.2%) because they had selected fewer than five means.

The frequency with which people found the optimal solution decreased significantly with treewidth ($\rho = -0.4828$; $p = .0311$; Fig. 6A). We found that treewidth alone explained 20.41% of the variance in the frequency with which people found the optimal solution across the 20 graphs ($F(1, 18) = 4.62$; $p = .0455$). We found that treewidth explained 25.25% of the variance in solution quality ($F(1, 18) = 6.08$; $p = .0240$) which significantly deteriorated as treewidth increased ($\rho = -0.4972$; $p = .0257$; Fig. 6B). The median amount of time people took to solve the problems did not increase significantly with treewidth ($\rho = 0.25$; $p = .28$) and treewidth explained only 0.6% of our participants' median response times ($F(1, 18) = 0.10$; $p = .76$). When we restricted the analysis to the time taken by optimal solutions, the relationship was still not statistically significant ($\rho = 0.2994$; $p = .1998$; $F(1, 18) = 0.10$; $p = .76$). Finally, treewidth explained only 8.8% of the variance in the perceived problem difficulty across the 20 graphs ($F(1, 18) = 1.74$; $p = .20$), and the correlation between treewidth and perceived difficulty was not statistically significant ($\rho = 0.26$; $p = .26$). Our participants were highly motivated to find the optimal solution (average rating 8.11 ± 0.06 out of 9). Thus, it appears unlikely that their motivation was a bottleneck to their performance. Furthermore, motivation appeared to be unaffected by treewidth ($\rho = 0.03$; $p = .91$). Thus, the observed differences in performance appear to result from the inherent difficulties of the means-selection problems posed by different goal systems.

In addition, we found that the edge expansion and the graph spectral gap (measured again as $d\text{-}\lambda_2$) were both significantly negatively correlated with the frequency of optimal solutions (edge expansion: $\rho = -0.4802$; $p = .0321$, spectral gap: $\rho = -0.4782$; $p = .0330$), while the average shortest path average graph eccentricity and the number and proportion of optimal paths showed a significant positive relationship (average shortest path: $\rho = 0.4912$; $p = .0279$, average eccentricity: $\rho = 0.4391$; $p = .0528$, number of optimal paths: $\rho = 0.7256$; $p = .0003$, proportion of optimal paths: $\rho = 0.7256$; $p = .0003$). The graph vertex and edge expansions and the number and proportion of optimal paths showed a significant correlation with the average solution quality (vertex expansion: $\rho = 0.4431$; $p = .0504$, edge expansion: $\rho = 0.4832$; $p = .0309$, number of optimal paths: $\rho = 0.6856$, $p = .0008$, proportion of optimal paths: $\rho = 0.6956$, $p = .0008$). None of the metrics surveyed were significantly correlated with median participant response times.

3.6. Experiment 3: A more concrete SC task

While Experiments 1 and 2 captured some of the computational challenges of goal achievement, the tasks were relatively abstract. Experiment 3 addresses this limitation by assigning semantic labels to the 24 goals. These labels were common new-year resolutions, such as "get in shape" and "earn more money." We also used a different interface to minimize the effect of visualization that arise from graph drawings in the first two experiments.

3.6.1. Methods

3.6.1.1. Participants: We recruited 600 participants on Amazon Mechanical Turk.

3.6.1.2. Stimuli and procedure: Participants were paid \$0.38 for about 5 min of work plus a performance-dependent bonus of \$0.50 if they found an optimal solution. Each participant was randomly assigned to 1 of the 20 graph structures used in Experiments 1 and 2. For each graph, the order in which the means were listed and the order in which the goals were listed was randomized between participants. The participants' task was to achieve all goals with as few means as possible. The graphical interface of the task was changed to reduce visual clutter. Instead of drawing edges between means and goals, the goals achieved by each means were listed next to it (see Fig. 7). The cover story was similar to the one used in Experiment 1. The consent form required that participants had not participated in any of our previous goal-pursuit experiments.

3.6.2. Results

As shown in Fig. 8A,B, Experiment 3 replicated the effects of treewidth on the solution frequency and the solution quality in a more realistic setting: Increasing the treewidth of the goal system significantly decreased both solution frequency ($\rho = -0.756$, $p = .0001$) and solution quality ($\rho = -0.6$, $p = .004$). Furthermore, we found that the magnitude of the graph spectral gap, graph edge and vertex expansions, average eccentricity, average shortest path, graph diameter, and the number and proportion of optimal paths were all significantly correlated with the frequency with which human participants identified the optimal solution (average eccentricity: $\rho = 0.583$; $p = .007$, average shortest path: $\rho = 0.651$; $p = .002$, graph diameter: $\rho = 0.525$; $p = .017$, spectral gap: $\rho = -0.708$; $p = .0005$, edge expansion: $\rho = -0.7$; $p = .0006$, vertex expansion: $\rho = -0.7303$; $p = .0003$, number of optimal paths: $\rho = 0.5596$; $p = .0103$, proportion of optimal paths: $\rho = 0.5697$, $p = .0087$).

Similarly, the magnitude of the graph spectral gap, the graph edge and vertex expansions, average eccentricity, and average shortest path were all significantly correlated with the average solution quality of human responses (average eccentricity: $\rho = 0.4872$; $p = .0293$, aver-

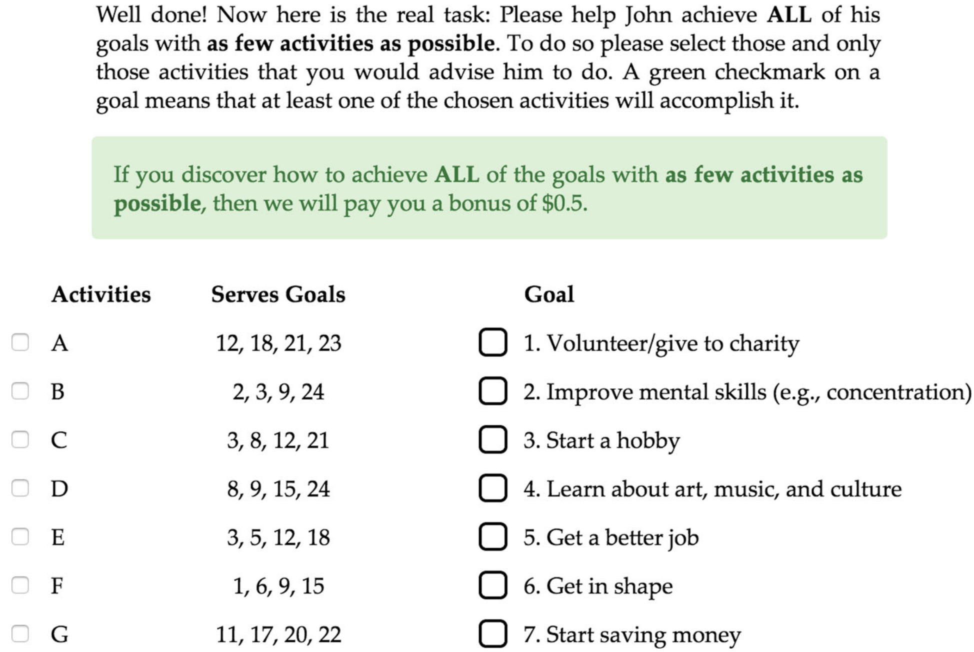


Fig. 7. Interface and instructions for Experiment 3.

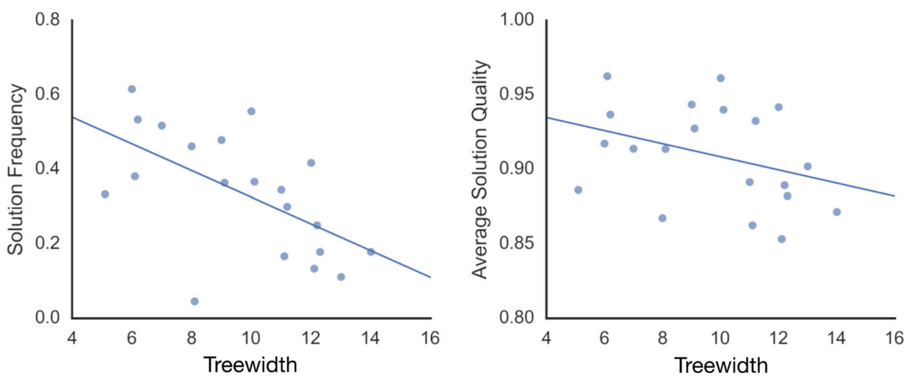


Fig. 8. Human performance in the semantic Set Cover task of Experiment 3 as a function of treewidth. Left: Solution Frequency. Right: Solution Quality.

age shortest path: $\rho = 0.5243$; $p = .0176$, spectral gap: $\rho = -0.5308$; $p = .0160$, edge expansion: $\rho = -0.5670$; $p = .0091$, and vertex expansion: $\rho = -0.6047$; $p = .0047$). The number and proportion of optimal paths were negatively correlated with the average solution quality of human responses (number of optimal paths: $\rho = -0.4955$; $p = .0263$, proportion of optimal paths: $\rho = -0.5052$; $p = .0231$). None of the features were significantly correlated with median participant response times.

Participants rated their motivation to find a solution that achieves all goals with the minimal number of means as 7.38 on a 9-point scale, their motivation to finish the task as quickly as possible and move on as 4.35, and the difficulty of the task as 5.67. The size of the optimal solution was positively correlated with the average solution quality ($\rho = 0.46, p = .0387$) but not with the average solution frequency $\rho = 0.25, p = .29$.

4. General discussion

We hypothesized that difficulties people experience in means selection tasks happen because the complex structure of a network interconnecting goals and means can make it difficult to choose the optimal set of means needed to pursue all goals simultaneously. We tested this hypothesis in a series of behavioral experiments and found that theoretically motivated structural properties of goal systems are highly predictive of human performance. Our findings are consistent with predictions we derived from the literature in theoretical computer science. This suggests that computational complexity theory and algorithm design can be useful theoretical tools for understanding what makes tasks hard for people to solve.

4.1. Implications for motivation and goal pursuit

The empirical results presented here indicate that the combinatorial optimization problems people need to solve to achieve their goals might be a critical bottleneck to success regardless of how motivated they are. Therefore, psychological theories seeking to explain failures of self-control and motivation should acknowledge the computational problems that people need to solve when pursuing multiple goals. For example, predictions of how successful a person will be should not solely be based on their motivation or skills but should also consider the structure of their goal system, as even highly motivated people may perform poorly when the computational burden involved in goal pursuit is high. More generally, our work is one of the first to investigate the *bounded rationality* of human goal pursuit. Future research looking into the connections between bounded rationality, computational complexity, and goal striving offers the opportunity to advance our understanding of both motivation and cognition (in particular, problem-solving).

Managing means selection tasks "in the wild" is likely to differ from our experiments in several aspects. People may not be aware of all their goals, all the means that are instrumental to their goals, and all connections between the desired goals and the available means. Furthermore, people's limited working memory capacity (Miller, 1956) can make it hard for them to maintain the entirety of their goal system in working memory when deciding which means to pursue. Hence, managing multiple goals and means in daily life is likely to be even more challenging to solve compared to our experiments. The additional difficulty of maintaining a network of goals and means in working memory along with the uncertainties of connection of goals and means strengthens our conclusion that a major source of difficulties in goal pursuit hinges in the computational hardness of the problems involved.

An interesting future direction is to measure people's goal systems and extract graph-theoretic properties of these networks to characterize their structure. Future work could examine whether the goal systems of people who are more successful in achieving their goals differ structurally from those of people who find it harder to accomplish their aims. This approach can shed light on the algorithmic processes that people apply toward more effective goal management. It would also enable more ecologically valid experiments that study means-selection in simulations of real-life goal pursuits.

Since our measures of tree-likeness predict goal achievement from the structure of a person's goal system, they could be used to detect overly complicated goal systems that set people up for ineffective goal management and help them restructure their goals to make it easier to achieve their goals. Our findings suggest that designing more tractable goal systems could be achieved by reducing the number of goals or reducing the interdependencies between different goals (i.e., reducing the treewidth of the goal system). Additionally, computerized tools could be used to help people execute algorithms to deal with their goal systems (although, computers are, to some extent, constrained by intractability considerations as well).

4.2. *Implications for human problem-solving*

There is extensive theorizing within cognitive psychology studying computational architectures that are designed to achieve goals in large problem spaces (e.g., Anderson, 1996; Newell & Simon, 1972) and this literature has influenced motivational accounts, such as goal system theory. For example, the notion that people use lower-level means to achieve higher-level goals has originated from Newell and Simon's work on human problem-solving. These theories study properties of solution spaces that can influence the difficulty of reaching an end state from a start position using a set of predefined operators (e.g., size of solution space, number of paths to the solutions). Importantly, they do not consider as we do here network properties of goal systems, and we are not aware of any previous work that has studied the connection between network properties connecting goals and means to human problem-solving. We believe that our empirical and theoretical findings are relevant to problem-solving and may offer new explanations why certain problems are harder for people to solve than others.

Two measures that are related to theories of human problem-solving: the number and proportions of paths to reach optimal solutions were found to be positively correlated with human performance in Experiments 1 and 2. These two variables were negatively correlated with human performance in Experiment 3. Such a negative correlation is in the "opposite direction" of theorizing of the problem-solving literature postulating that a larger number of paths should result with better performance rather than worse, as was found in Experiment 3. While more research is needed to explain this negative relationship, it does appear that tree-likeness is a more robust predictor of human performance in computational problems related to goal systems.

Unlike our treewidth measure, the path-related predictors are not supported by theoretical findings from computational complexity and algorithm design. For example, the number of paths to an optimal solution is typically large (exponential in the number of nodes) even for tractable (polynomial-time solvable) problems. Nevertheless, the empirical evidence here suggests that these measures can have predictive value for human performance in goal pursuit and human problem-solving. We believe that measures related to solution space might be especially effective for problems of smaller magnitude where asymptotic considerations are less relevant. Studying the factors that enhance or diminish the predictive power of the graph-theoretic variables studied in this work is an interesting direction for future research.

4.3. *Implications for cognitive science*

As noted, NP-hard problems are abundant in various domains, such as perception, decision-making, language and communication, categorization, and problem-solving. Our results suggest that computational intractability can constrain the algorithms people use to solve such challenging problems. Furthermore, intractability results can be used to reason about the cognitive and behavioral outcomes of bounded rational agents who face difficult problems. Structural properties can be used to reason about whether instances faced by the cognitive system are likely to be hard or easy to solve. This research program can be seen as a version of rational analysis (Anderson, 1990) that characterizes the structure of the problems people face in terms of properties that can be exploited by specialized algorithms. Leveraging theoretical insights from computational complexity toward classifying which problems are hard or easy to solve has the advantage of being *representation impendent*. Namely, it is believed that intractability results from computational complexity are universal in the sense that they apply for all reasonable computational models, such as finite automata, Turing machines, counter programs, and neural networks (Papadimitriou, 1994). Such universality is advantageous from a cognitive science perspective as it is often difficult or impossible to know which computational model or representation is used by the mind (Anderson, 1978).

Our work is focused on properties of computational problems that make them easy or hard to solve for people. Can our theorizing be used to better understand the nature of algorithms people use in goal pursuit, problem-solving, or other domains? First, our result lends support to the idea that people use *efficient* algorithms, narrowing the potential candidates for algorithms used by the mind. Moreover, our results indicate that people use algorithms that are sensitive to the treewidth of goal systems and perform better on problems of lower treewidth. We believe that similar findings may emerge on other NP-hard problems on networks. Finally, treewidth is by no means the only measure that can parameterize the complexity of solving NP-hard problems and there is a whole field of fixed-parameter tractability which studies such measures (Cygan et al., 2015; Downey & Fellows, 2013). The relevance of parameterized complexity to understanding which algorithms people use for solving effectively hard problems has been outlined before (van Rooij, 2008; Van Rooij, & Wareham, 2008). We believe that using these theoretical ideas to identify the algorithms people employ is an interesting future direction.

We have focused here on NP-hard problems. For algorithmic problems that admit polynomial-time solutions, the theoretical validity of treewidth as a predictor is less clear as a polynomial algorithm can solve the problem efficiently independently of how large or small the treewidth of the graph is. For example, finding the shortest path in a graph with positive weights on the edges can be done in nearly linear time for any network using Dijkstra's algorithm (Kleinberg & Tardos, 2005). A polynomial-time algorithm may be still complicated for people to implement and consume prohibited amounts of memory. Hence, it is possible that treewidth or other measures can be used to predict the hardness of algorithmic problems even if there is a polynomial-time algorithm for these. Investigating this possibility is left for future research.

4.4. Limitations

A limitation of the experiments reported here is that they consider a restricted form of goal systems. To allow clean and simple examination of our hypotheses, we focused on unweighted networks where all goals have the same value and all means have the same cost. However, in real life, some goals are much more valuable than others and some means are much more expensive than others. Future experiments should investigate the predictive power of computational complexity theory in more realistic settings where the goals differ in their values and the means differ in their costs.[6] As in goal-systems theory, we are mostly concerned with shallow goal systems. However, it is quite likely that people encounter goal systems of depth greater than 2. Examining empirically how people cope with such deeper systems is of potential interest. Future research could also consider problems where multiple means must be invested to achieve a goal and some means contribute more toward achieving a goal than others (namely, goal systems and goal systems with weighted edges). In particular, one would expect that in certain circumstances, some means would inhibit some goals (Kruglanski, Chernikova, Babush, Dugas, & Schumpe, 2015). Hence, it could be of interest to examine how people perform on goal systems with negative weights. Finally, goal pursuit often entails sequential decisions and actions over time. Looking into such sequential tasks and how people deal with the computational challenges they entail is another avenue for future research.

The term "goal" is very broad and our assumption of a clean computational model of goals, means, and the connection between them is likely to be more relevant to lower-level goals (such as those studied empirically in papers concerned with goal systems theory) that are often more explicit and easier to connect to concrete means in contrast to higher-level goals more ambiguous goals, such as "being a good person" or "making friends." Our theorizing and modeling are likely to apply more strongly to concrete low-level goals that people are trying to achieve.

4.5. Conclusion

Striving for goals is essential to the human experience, and progress toward one's goals is critical to attain satisfaction and well-being. Our work sheds new light on the factors which determine how difficult it is to juggle between multiple goals. While previous motivational accounts are silent with respect to the role of computational challenges involved in goal achievement, our theory suggests that whether a person will be able to achieve their goals partly depends on the structure of their goal system, as this structure determines how computationally demanding it is to find the right course of action. We believe that the theorizing and experiments presented here have the potential to advance our basic understanding of human performance and could lead to new methods for helping people restructure their goals so that they can be pursued more effectively.

Acknowledgments

We thank Rina Dechter for her advice on treewidth computation. We thank Kevin Leyton -Brown, Nira Liberman and Rika Antonova for useful discussions.

Notes

1 We assume basic familiarity with concepts from computer science such as the definition of running time of algorithms and NP-completeness whose presentation is beyond the scope of this work. Intuitively, NP-hardness of a computational problem means that no efficient algorithm for solving the problem is likely to exist. More background on these topics can be found in Garey and Johnson (1979), Kleinberg and Tardos (2005), and van Rooij (2008). For a precise definition of the term "NP-hard" and more background on this concept, we refer the interested reader to Goldreich (2008), Goldreich (2010), Harel and Feldman (2004).
2 For a formal definition of treewidth, see Section 3.1.1 and Appendix A.
3 We refer the reader to Appendix A for graph-theoretical terminology that we use.
4 For a *single* instance, there is a trivial algorithm that solves the optimization problem efficiently. For example, for an MC instance, the algorithm that simply prints the optimal sets of means solves the problem. However, having such an instance-specific algorithm in people's heads appears psychologically implausible and indeed our empirical results point against the possibility of people using such an algorithm.
5 We calculated FVS using an implementation based on Iwata (2016) and Wahlström (2014) (available at https://github.com/wata-orz/fvs).
6 In a follow-up experiment, we found that tasks where the goals differ in value are significantly more difficult than tasks where all goals have the same value. Concretely, when people were tasked to select five means to maximize the sum of the achieved goals' values, the frequency of the optimal solution dropped to 7%. This suggests that, as goal management tasks become more realistic, their computational difficulty becomes even more limiting. Due to low average performance in this challenging task, the effect of treewidth on solution frequency was no longer detectable ($\rho = -0.18$, $p = .46$). This suggests that treewidth is not the only factor determining computational difficulty.

References

Ainslie, G., & Haslam, N. (1992). Self-control. In G. Loewenstein & J. Elster (Eds.), *Choice over time* (pp. 177–209). New York: Sage Foundation.

Alber, J., & Niedermeier, R. (2002). Improved tree decomposition based algorithms for domination-like problems. In *Latin American Symposium on Theoretical Informatics* (pp. 613–627). Berlin: Springer-Verlag.

Alon, N., & Milman, V. D. (1985). λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, *38*(1), 73–88.

Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review*, *85*(4), 249–277.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, *51*(4), 355.

Ariely, D., & Wertenbroch, K. (2002). Procrastination, deadlines, and performance: Using precommitment to regulate one's behavior. *Psychological Science*, *13*, 219–224.

Baker, B. S. (1994). Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, *41*(1), 153–180.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W. H. Freeman/Times Books/Henry Holt & Co.

Baumeister, R. F., & Heatherton, T. F. (1996). Self-regulation failure: An overview. *Psychological Inquiry*, *7*, 1–15.

Bodlaender, H. L. (1988). Dynamic programming on graphs with bounded treewidth. In *International Colloquium on Automata, Languages, and Programming* (pp. 105–118). Berlin: Springer-Verlag.

Bossaerts, P., & Murawski, C. (2017). Computational complexity and human decision-making. *Trends in Cognitive Sciences*, *21*(12), 917–929.

Campbell, D. J. (1988). Task complexity: A review and analysis. *Academy of Management Review*, *13*(1), 40–52.

Carruthers, S., Masson, M. E., & Stege, U. (2012). Human performance on hard non-Euclidean graph problems: Vertex cover. *Journal of Problem Solving*, *5*(1), 34–55.

Carver, C. S., & Scheier, M. F. (1998). *On the self-regulation of behavior*. New York: Cambridge University Press.

Clementi, A. E., & Trevisan, L. (1999). Improved non-approximability results for minimum vertex cover with density constraints. *Theoretical Computer Science*, *225*(1), 113–128.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms* (2nd ed.). Cambridge, MA: MIT Press.

Cygan, M., Fomin, F. V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized algorithms* (Vol. *4*). Cham: Springer.

Diestel, R. (2012). Graph Theory, volume 173 of. Graduate texts in mathematics, 593.

Downey, R. G., & Fellows, M. R. (2013). *Fundamentals of parameterized complexity* (Vol. *4*). London: Springer.

Dworkin, L., & Kearns, M. (2015). From "in" to "over": Behavioral experiments on whole-network computation. In *Third AAAI Conference on Human Computation and Crowdsourcing* (pp. 52–61). Palo Alto, CA: AAAI Press.

Enemark, D. P., McCubbins, M. D., Paturi, R., & Weller, N. (2011). Does more connectivity help groups to solve social problems? In *Proceedings of the 12th ACM Conference on Electronic Commerce* (pp. 21–26). New York: ACM Press.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM*, *45*(4), 634–652.

Fishbach, A., Shah, J. Y., & Kruglanski, A. W. (2004). Emotional transfer in goal systems. *Journal of Experimental Social Psychology*, *40*(6), 723–738.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: Freeman.

Gigerenzer, G. (2008). *Rationality for mortals*. New York: Oxford University Press.

Gogate, V., & Dechter, R. (2004). A complete anytime algorithm for treewidth. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (pp. 201–208). Arlington, VA: AUAI Press.

Goldreich, O. (2008). *Computational complexity: A conceptual perspective*. New York: Cambridge University Press.

Goldreich, O. (2010). *NP, and NP-completeness*: The basics of complexity theory. New York: Cambridge University Press.

Gollwitzer, P. M. (1999). Implementation intentions: Strong effects of simple plans. *American Psychologist*, *54*, 493–503.

Green, L., Fristoe, N., & Myerson, J. (1994). Temporal discounting and preference reversals in choice between delayed outcomes. *Psychonomic Bulletin & Review*, *1*(3), 383–389.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, *7*(2), 217–229.

Harel, D., & Feldman, Y. (2004). *Algorithmics: The spirit of computing* (3rd ed.). Berlin: Springer-Verlag.

Huang, S. C., & Zhang, Y. (2013). All roads lead to Rome: The impact of multiple attainment means on motivation. *Journal of Personality and Social Psychology*, *104*(2), 236–248.

Hutter, F., Xu, L., Hoos, H. H., & Leyton-Brown, K. (2014). Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, *206*, 79–111.

Iwata, Y. (2016). Linear-time kernelization for feedback vertex set. *arXiv preprint arXiv:1608.01463*.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice-Hall.

Klein, P., Plotkin, S. A., & Rao, S. (1993). Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing* (pp. 682–690). New York: ACM Press.

Kleinberg, J., & Tardos, E. (2005). *Algorithm design*. London: Pearson Education.

Kloks, T. (1994). *Treewidth: Computations and approximations* (Vol. *842*). New York: Springer Science & Business Media.

Köpetz, C., Faber, T., Fishbach, A., & Kruglanski, A. W. (2011). The multifinality constraints effect: How goal multiplicity narrows the means set to a focal end. *Journal of Personality and Social Psychology*, *100*(5), 810.

Kruglanski, A. W., Bélanger, J. J., Chen, X., Köpetz, C., Pierro, A., & Mannetti, L. (2012). The energetics of motivated cognition: A force-field analysis. *Psychological Review*, *119*(1), 1–20.

Kruglanski, A. W., Chernikova, M., Babush, M., Dugas, M., & Schumpe, B. (2015). The architecture of goal systems: Multifinality, equifinality, and counterfinality in means-end relations. *Advances in Motivation Science*, *2*, 69–98.

Kruglanski, A. W., Shah, Y. J., Fishbach, A., Friedman, R., Chun, W. Y., & Sleeth-Keppler, D. (2002). A theory of goal-systems. In M. P. Zanna (Ed.), *Advances in experimental social psychology* (pp. 331–378). San Diego, CA: Academic Press.

Kruglanski, A. W., & Kopetz, C. (2009). What is so special (and non-special) about goals? A view from the cognitive perspective. In G. B. Moskowitz & H. Grant (Eds.), *The psychology of goals* (pp. 27–55). New York: Guilford Press.

Kruglanski, A. W., & Kopetz, C. (2010). The self control challenge in human action: Unpacking the dilemma. In R. Hassin, K. Ochsner, & Y. Trope (Eds.), *Self-control in society, mind, and brain* (pp. 297–312). Oxford: Oxford University Press.

Leyton-Brown, K., Nudelman, E., & Shoham, Y. (2009). Empirical hardness models: Methodology and a case study on combinatorial auctions. *Journal of the ACM*, *56*(4), 1–52.

Liberman, N., & Dar, R. (2009). Normal and pathological consequences of encountering difficulties in monitoring progress toward goals. In G. B. Moskowitz & H. Grant (Eds.), *The psychology of goals* (pp. 277–303). New York: Guilford.

Little, B. R., & Gee, T. (2007). The methodology of personal projects analysis: Four modules and a funnel. In B. R. Little, K. Salmela-Aro, & S. D. Philipps (Eds.), *Personal project pursuit: Goals, action, and human flourishing* (pp. 51-94). New York: Psychology Press.

Louro, M. S., Pieters, R., & Zeelenberg, M. (2007). Dynamics of multiple goal pursuit. *Journal of Personality and Social Psychology*, *93*, 174–193.

MacGregor, J. N., & Ormerod, T. (1996). Human performance on the traveling salesman problem. *Attention, Perception, & Psychophysics*, *58*(4), 527–539.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2000). A model of human performance on the travelling salesperson problem. *Memory and Cognition*, *28*, 1183–1190.

Marx, D. (2007). Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science* (pp. 169–179). Washington, DC: IEEE Computer Society.

Miller, G. A. (1956). The magic number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*, 91–97.

Mischel, W. (1974). Processes in delay of gratification. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (Vol. *7*, pp. 249–292). San Diego, CA: Academic Press.

Myerson, J., & Green, L. (1995). Discounting of delayed rewards: Models of individual choice. *Journal of the Experimental Analysis of Behavior*, *64*(3), 263–276.

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. *104* No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review*, *65*(3), 151–166.

Papadimitriou, C. H. (1994). *Computational complexity*. Chichester: Addison-Wesley.

Perarnau, G., & Serra, O. (2014). On the tree-depth of random graphs. *Discrete Applied Mathematics*, *168*, 119–126.

Robertson, N., & Seymour, P. D. (1986). Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, *7*(3), 309–322.

Sanborn, A., Griffiths, T., & Navarro, D. J. (2010). Rational approximation to rational models: Alternative algorithms for category learning. *Psychological Review*, *117*(4), 1144–1167.

Shah, J. Y., & Kruglanski, A. W. (2000). Aspects of goal networks: Implications for self-regulation. In M. Boekaerts & P. R. Pintrich (Eds.), *Handbook of self-regulation* (pp. 85–110). San Diego, CA: Academic Press.

Shenhav, A., Musslick, S., Lieder, F., Kool, W., Griffiths, T. L., Cohen, J. D., & Botvinick, M. M. (2017). Toward a rational and mechanistic account of mental effort. *Annual Review of Neuroscience*, *40*, 99–124.

Stajkovic, A. D., & Luthans, F. (1998). Self-efficacy and work-related performance: A meta-analysis. *Psychological Bulletin*, *124*(2), 240.

Steel, P. (2007). The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological Bulletin*, *133*, 65–94.

Thagard, P., & Verbeurgt, K. (1998). Coherence as constraint satisfaction. *Cognitive Science*, *22*, 1–24.

Trevisan, L. (2001). Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing* (pp. 453–461), New York: ACM Press.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science*, *32*, 939–984.

van Rooij, I., Stege, U., & Kadlec, H. (2005). Sources of complexity in subset choice. *Journal of Mathematical Psychology*, *49*(2), 160–187.

Van Rooij, I., & Wareham, T. (2008). Parameterized complexity in cognitive modeling: Foundations, applications and opportunities. *Computer Journal*, *51*(3), 385–404.

van Rooij, I., & Wareham, T. (2012). Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, *56*, 232–247.

van Rooij, I., Wright, C., & Wareham, H. T. (2012). Intractability and the use of heuristics in psychological explanations. *Synthese*, *187*, 471–487.

van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability: A guide to classical and parameterized complexity analysis*. Cambridge University Press.

Vazirani, V. V. (2003). *Approximation algorithms*. Atlanta, GA: Springer.

Vroom, V. H. (1964). *Work and motivation*. New York: Wiley.

Wahlström, M. (2014). Half-integrality, LP-branching and FPT algorithms. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1762–1781). New York: ACM Press.

Appendix A: Graph-theoretic definitions

In this section, we provide some graph-theoretic definitions. More details can be found in Diestel (2012). An *undirected* graph G is composed of a finite set of vertices, V, and a set of edges, E, which is a subset of the family of all subsets of V which are of size exactly 2. Namely, each edge is an *unordered* pair $\{u, v\}$ where both u, v are in V. We write $G = (V, E)$

to signify a graph G that consists of a vertex set V and edge set E. We say that a vertex y is a *neighbor* of x if $\{x, y\} \in E$. Alternatively, we say that y is adjacent to x. Note that x is adjacent (a neighbor) to y if and only if y is adjacent to x. The *degree* of x is defined as the number of neighbors of x. A *simple path* is a set of *distinct* vertices $v_1, v_2, \ldots, v_k$ such that for every $1 \leq i < k$, v_i is a neighbor of v_{i+1}. The *length* of the path is the number of vertices in the path minus 1 (i.e., $k - 1$). A graph is *connected* if every two vertices are connected by a path of finite length. A *simple cycle* is a set of distinct vertices $v_1, v_2, \ldots, v_k, v_1$ such that, for every i, v_i is adjacent tov_{i+1} and v_1 is adjacent to v_k. A connected graph with no cycles is called a *tree*. An *independent set* is a subset I of vertices that contains no edges. $G = (V, E)$ *is bipartite* if the vertex set of V is the union of two disjoint independent sets. Given a list of vertices, $v_1, v_2, \ldots, v_r$, the degree sequence of these vertices is the list of the degrees of $v_1, v_2, \ldots, v_r$. The average degree of a graph is the sum of the degrees normalized by the number of vertices. Finally, we comment that the sum of degrees (hence the average degree) is related to the number of edges in a graph: It is known to equal $2|E|$, where $|E|$ is the number of vertices of the graph.

A *tree decomposition* of a graph $G = (V,E)$ is a tree $T = (U,F)$ and a function $\chi : U \rightarrow 2^V$ (namely, for every $u \in U$, $\chi(u)$ is subset of vertices from V). We term the subsets of vertices of V associated with the vertices of T, bags. Tree decompositions are defined by two properties: (i) every edge $(u, v) \in E$ belongs to some bag $\chi(t), t \in U$ and (ii) for every vertex $v \in V$, the bags containing v form a connected subtree of T. The *width* of the tree decomposition is one less than the cardinality of the largest bag (e.g., $\max_{t \in T} |\chi(t) - 1|$ and the treewidth of G, denoted by tw(G), is the smallest width for which G has a tree decomposition. We shall omit G from this notation when clear from the context. See Fig. 3 for examples and further explanations of treewidth.

For the computation of the treewidth of our graphs, we used the QUICKBB library (http://graphmod.ics.uci.edu/group/quickbb) which is described in Gogate and Dechter (2004). Using a cluster of machines, each containing 96GB of RAM, the treewidth computation of most graphs was completed within 40 h.

Given a graph G and a nonempty (strict) subset S of V, a *cut* $(S, V \backslash S)$ is a partition of the vertices of G to two nonempty sets S and $V \backslash S$. An edge of G *crosses* the cut if exactly one of its endpoints belongs to S. We denote by$\partial(S)$ the set of edges crossing the cut $(S, V \backslash S)$ and define $N(S)$ to be the set of all vertices in $V \backslash S$ having a neighbor in S. The vertex-expansion of G is defined as: $\min_{\phi \neq S \subseteq V, |S| \leq |V|/2} \frac{|N(S)|}{|S|}$. The edge-expansion of G is $\min_{\phi \neq S \subseteq V, |S| \leq |V|/2} \frac{|\partial(S)|}{|S|}$.

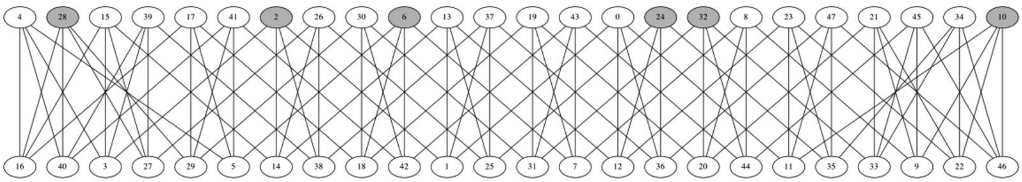
We computed both vertex and edge expansion by formulating these quantities as integer linear programs and used IBM's CPLEX to solve these programs.

Given an undirected graph $G = (V, E)$ over n vertices, it can be represented by a symmetric matrix A_G of dimension n with 0–1 entries which is called the adjacency matrix of G. We have that $A_G(i, j) = 1$if the ith vertex is connected to the jth vertex and 0 otherwise. The adjacency matrices are symmetric, hence have n real eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. For bipartite d-regular graphs, it is known that $\lambda_1 = d$ and $\lambda_n = -d$. The classical (discrete) Cheeger's inequality (Alon & Milman, 1985) implies that λ_2 and the edge expansion are related: The larger $d - \lambda_2$ is, the larger the edge expansion of the graph. The *spectral gap* of G is defined as $d - \lambda_2$.

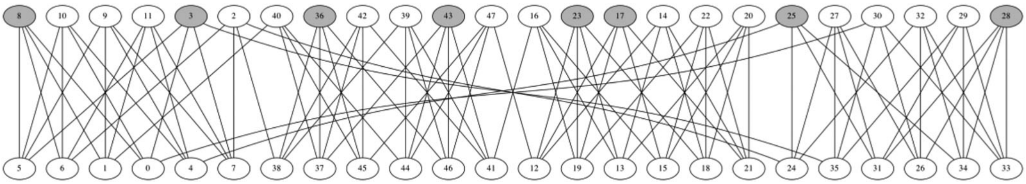
Appendix B: Problem graphs

In the following figures, we provide a visualization of the bipartite graphs used for the set cover and max-coverage task along with an optimal (minimum cardinality) set cover (shaded vertices).

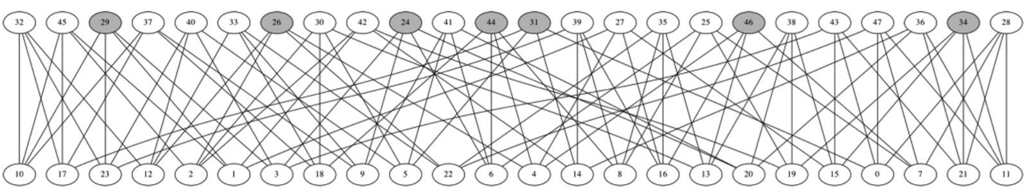
TW 13



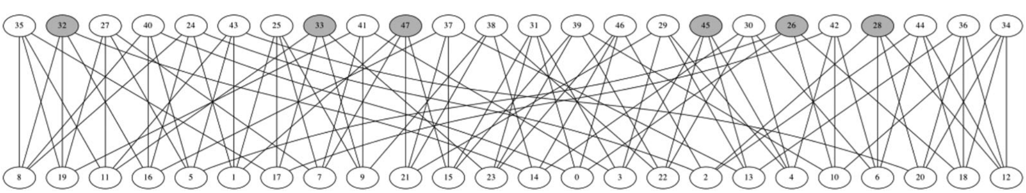
TW 6



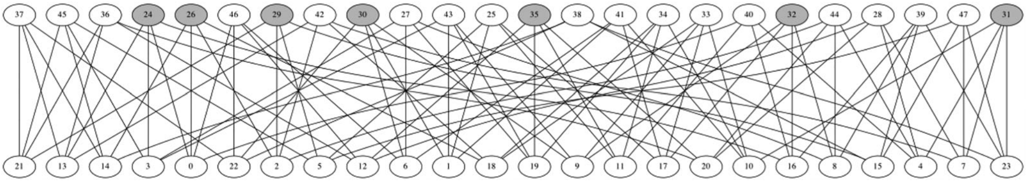
TW 12



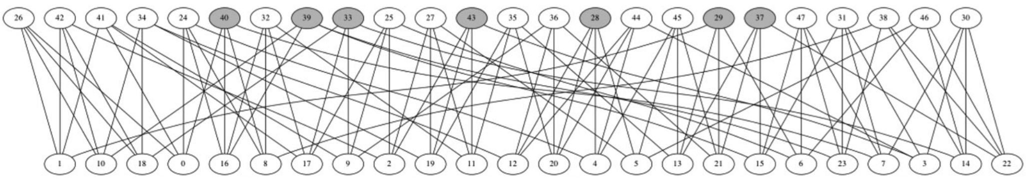
TW 11



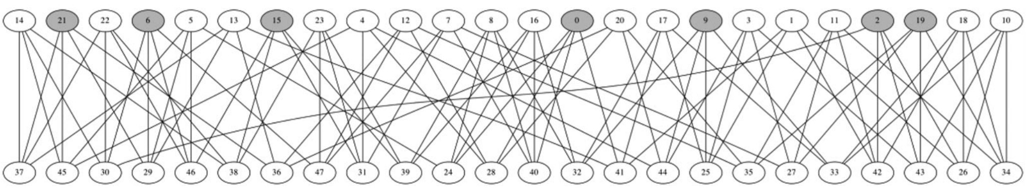
TW 12



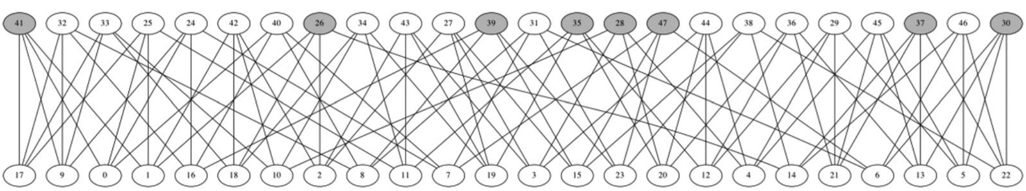
TW 11



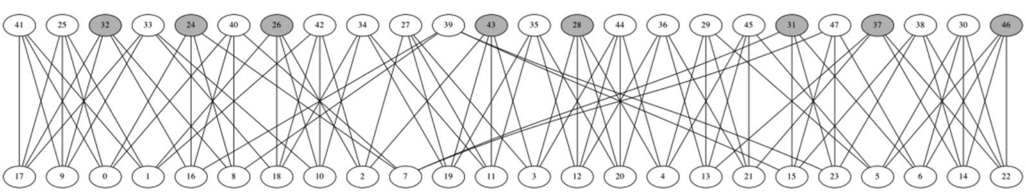
TW 10



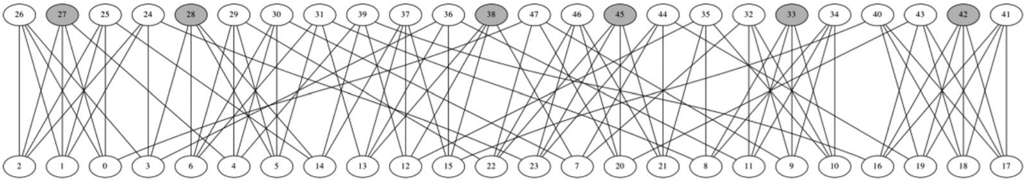
TW 8



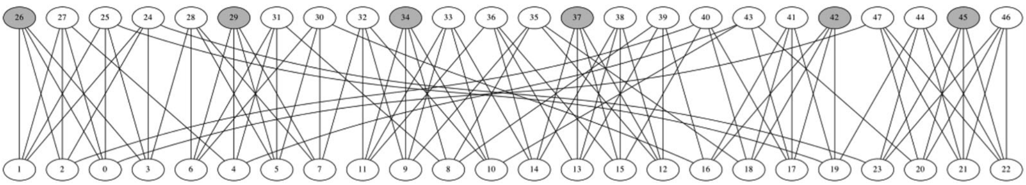
TW 5



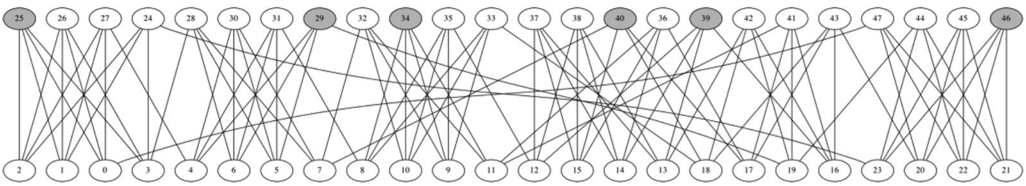
TW 9



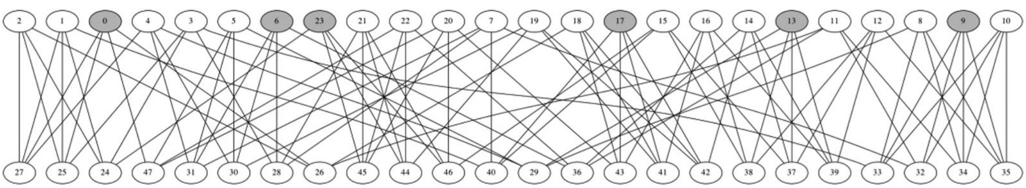
TW 7



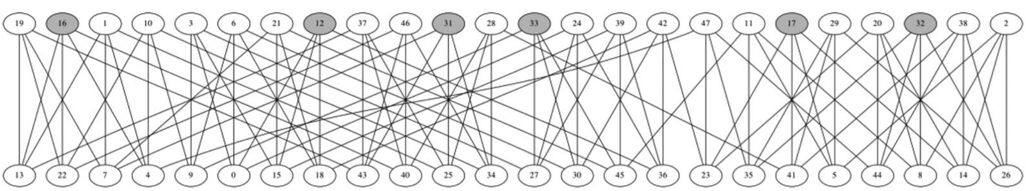
TW 6



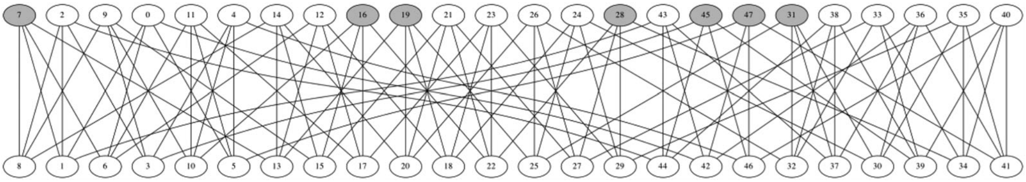
TW 10



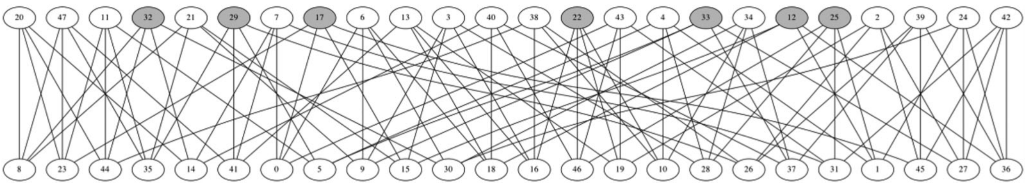
TW 6



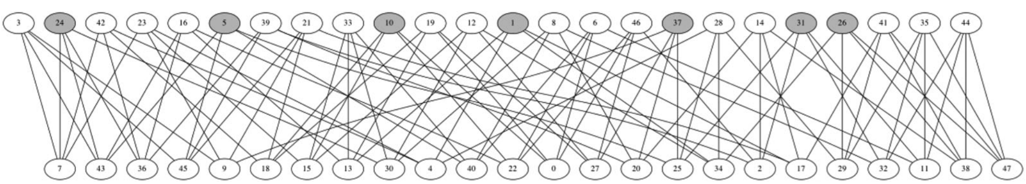
TW 12



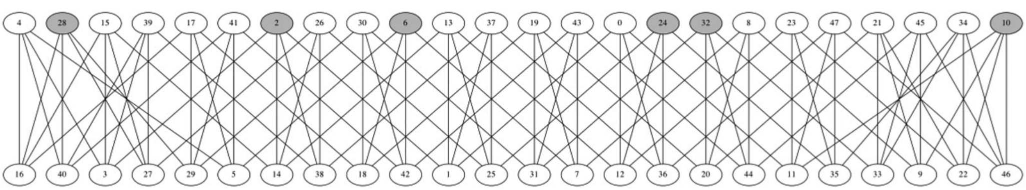
TW 14



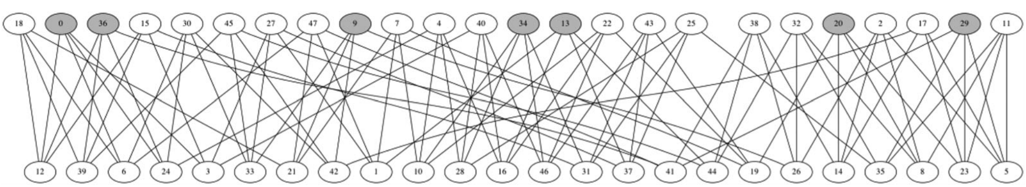
TW 12



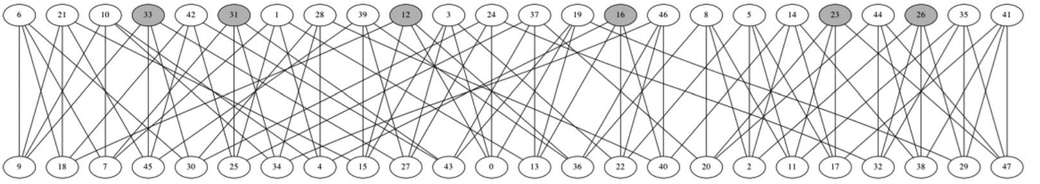
TW 8



TW 11



TW 9


6
21
10
33
42
31
1
28
39
12
3
24
37
19
16
46
8
5
14
23
44
26
35
41
9
18
7
45
30
25
34
4
15
27
43
0
13
36
22
40
20
2
11
17
32
38
29
47