Sensitivity analysis over polymatroids with applications to game theory

Michael Krause

Bachelor Thesis

First Examiner: Prof. Dr. rer. nat. Britta Peis
Second Examiner: Prof. Dr. Ir. Gerhard Woeginger

RWTH Aachen University, Aachen, Germany
March 29, 2017

Zusammenfassung

Diese Arbeit behandelt einige kombinatorische Strukturen und ihre Anwendungen in der Spieltheorie. Es wird eine Reihe von klassischen Resultaten und aktuellen Veröffentlichungen besprochen und mit Beispielen erläutert.

Neben einer Einführung in Matroid- und Polymatroidtheorie werden grundlegende Konzepte aus der kooperativen und nicht-kooperativen Spieltheorie wiederholt. Insbesondere werden Anwendungen der Sensitivätsanalyse von Optimierungsproblemen über Polymatroiden im Hinblick auf Matroid Congestion Games und verwandte Spiele vorgestellt.

Zusätzlich wird gezeigt, dass die Polymatroidstruktur bestimmter konvexer kooperativer Spiele eine effiziente Neuberechnung von Corevektoren erlaubt.

Abstract

This thesis deals with certain combinatorial structures and their applications to game theoretic problems. As part of this, we will review several classical results and recent publications on these topics and give illustrating examples.

We will give an introduction to matroid and polymatroid theory as well as basic concepts from non-cooperative and cooperative games. Our focus will be on applications of sensitivity results for polymatroid optimization to matroid congestion games and related concepts.

In addition, we will demonstrate how the polymatroid structure of certain convex cooperative games can be exploited to recompute core allocations efficiently.

Contents

List of Figures

Chapter 1

Introduction

How should traffic be routed through a road network? How should a bankrupt firm distribute its remaining estate among its creditors? These are two questions that are addressed by game theory. In this thesis, we will see that results from combinatorics can provide elegant answers to these questions. In particular, our aim is to give an accessible introduction to applications of polymatroid optimization to game theory. We will start with an overview of all topics that will be covered in later chapters.

Combinatorial optimization is the maximization or minimization of functions over countable sets. An example is the problem of finding a minimum cost path in a graph. These optimization problems may be difficult to solve algorithmically depending on the nature of the objective function and the structure of the underlying set. For example, when finding a minimum cost path, the objective function is linear: the cost of a path is the sum of the costs of its elements.

In this thesis, we study optimization problems on structures that arise from *submodular* set functions (and their siblings, *supermodular* set functions). Intuitively, with a submodular function, adding an element to a larger set does not increase the function value more than adding the same element to a smaller set. Such behavior is common in everyday situations: Imagine buying several items of the same type from the same store. The 20th item will not be more expensive than the 10th, however it might in fact be cheaper due to discounts offered by the store on large orders.

Two of the structures we will discuss in this thesis are called *matroids* and *polymatroids*. They are generalizations of matrices known from linear algebra and share variants of the basis exchange property (i.e., replacing a column vector in the basis of a matrix with another column vector of the same matrix, while retaining independence, yields another basis). This property will be of great importance when we look at applications of these structures in game theory.

Game theory is the mathematical study of interactions between self-interested, rational agents. It was originally a domain of mathematicians and economists, but solving game theoretic problems increasingly called for efficient algorithms. This has motivated the study of *algorithmic game theory*, which lies at the intersection of game theory and computer science.

Non-cooperative game theory studies scenarios with competing, self-interested agents that choose among a set of options in order to maximize their personal payoff. A special case of non-cooperative games are congestion games: Here, the options available to the agents correspond to resources, e.g. processor cores or paths through a network. The more agents use a resource, the more costly it becomes. The system reaches an equilibrium when none of the agents can improve their payoff by switching resources. Later, we will see that sensitivity analysis for polymatroid optimization (i.e., the analysis of changes to optimal solutions under changes to problem parameters) allows to rule out certain anomalies for a special case of congestion games related to matroids. Furthermore, it allows us to prove the existence of "pure" equilibria for a type of non-cooperative games called polymatroid games, where pure means that the agents do not randomize their choices of resources.

Cooperative game theory deals with situations where several agents work together to achieve some task and share the cost they experience when doing so. After completing their task, the agents must allocate the cost among themselves in such a way that no one is treated unfairly. Finding such a cost allocation can be algorithmically hard. In this thesis, we are particularly interested in recomputing a valid cost allocation under small changes to the task parameters.

We will begin by introducing the necessary background from combinatorial optimization before we move on to its applications in game theory.

Throughout this work, the pronoun "we" shall refer to "the author and the reader". In chapters 2 and 3 as well as in sections 4.1 to 4.3, we cover existing literature. Section 4.4 contains original work.

Chapter 2

From Matrices to Polymatroids

In this chapter, we introduce combinatorial structures that are fundamental for the later discussion. The definitions and results in this chapter are found in standard textbooks like [8], [9], [16] and [22].

2.1 Submodular functions

We begin with two types of set functions that will appear frequently in this work.

Definition 2.1 (Sub- and supermodular functions)
Let N be a finite set and 2^N its power set. A function $f\colon 2^N \to \mathbb{R}$ is called submodular *if for all $A, B \subseteq N$,*

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{2.1}$$

and supermodular *if*

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B). \tag{2.2}$$

As mentioned in the introduction, sub- and supermodular functions appear frequently in mathematical economics and can be interpreted intuitively: Consider a function that is both submodular and supermodular, i.e., the inequalities above hold with equality. Such a function is called *modular* and it assigns the same values to elements of N, no matter which subset of N they are part of. In other words, the valuation of set A plus the valuation of set B are exactly the valuations of the elements of both sets combined, counting duplicate elements twice. Then sub- and supermodular functions are two relaxations of this property.

Another intuition is given by the following lemma: a function is submodular (supermodular) if adding an element to a set adds more (less) valuation than adding the same element to a superset of that set.

Lemma 2.2
For a finite set N, $f\colon 2^N \to \mathbb{R}$ is submodular if and only if for all $S \subseteq L \subseteq N$ and $n \in N \setminus L$,

$$f(S \cup \{n\}) - f(S) \geq f(L \cup \{n\}) - f(L). \tag{2.3}$$

The converse inequality holds if f is supermodular.

Proof (see also Theorem 44.1 in [22])
We focus on the submodular case and note that the supermodular case can be proved similarly. For (2.1) $\implies$ (2.3), observe that for all $S \subseteq L \subseteq N, n \in N \setminus L$ by choosing $A = S \cup \{n\}$ and $B = L$ equation (2.1) gives

$$\begin{aligned} & f(S \cup \{n\}) + f(L) \geq f(S \cup \{n\} \cup L) + f((S \cup \{n\}) \cap L) \\ \Leftrightarrow\ & f(S \cup \{n\}) + f(L) \geq f(L \cup \{n\}) + f(S) \\ \Leftrightarrow\ & f(S \cup \{n\}) - f(S) \geq f(L \cup \{n\}) - f(L) \end{aligned}$$

For (2.3) $\implies$ (2.1), let $A, B \subseteq N$. If $A = \emptyset$, we have

$$\begin{aligned} & f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \\ \Leftrightarrow\ & f(\emptyset) + f(B) \geq f(B) + f(\emptyset) \end{aligned}$$

and if $A = N$, we have

$$\begin{aligned} & f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \\ \Leftrightarrow\ & f(N) + f(B) \geq f(N) + f(B) \end{aligned}$$

(similarly for $B = \emptyset$ or $B = N$). Furthermore, if $B \subseteq A$, we get

$$\begin{aligned} & f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \\ \Leftrightarrow\ & f(A) + f(B) \geq f(A) + f(B) \end{aligned}$$

and similarly for $A \subseteq B$.

Thus, we can now assume $A, B \subset N$ to be non-empty and $A \nsubseteq B, B \nsubseteq A$. Let $A \triangle B = (A \setminus B) \cup (B \setminus A)$ be the symmetric difference of A and B. Then $|A \triangle B| \geq 2$. We prove (2.1) by induction for $|A \triangle B| \geq 2$.

Assume $|A \triangle B| = 2$ and let $P = A \cap B$, then we can write $A = P \mathbin{\dot\cup} \{a\}$ and $B = P \mathbin{\dot\cup} \{b\}$ for elements $a \in A, b \in B$. Now, choosing $S = P, n = b$ and $L = P \cup \{a\}$, our assumption (2.3) yields

$$\begin{aligned} & f(P \cup \{b\}) - f(P) \geq f(P \cup \{a\} \cup \{b\}) - f(P \cup \{a\}) \\ \Leftrightarrow\quad & f(P \cup \{b\}) + f(P \cup \{a\}) \geq f(P \cup \{a\} \cup \{b\}) + f(P) \\ \Leftrightarrow\quad & f(P \cup \{b\}) + f(P \cup \{a\}) \geq \\ & \qquad f((P \cup \{b\}) \cup (P \cup \{a\})) + f((P \cup \{b\}) \cap (P \cup \{a\})) \\ \Leftrightarrow\quad & f(B) + f(A) \geq f(B \cup A) + f(B \cap A) \end{aligned}$$

For the induction step, assume $|A \triangle B| = i + 1, i \geq 2$. Then we have $|A \setminus B| \geq 2$ or $|B \setminus A| \geq 2$. Without loss of generality assume the former and let $n \in A \setminus B$ so that we can apply the induction hypothesis and get

$$\begin{aligned} & f(A \setminus \{n\}) + f(B) \geq f((A \setminus \{n\}) \cup B) + f((A \setminus \{n\}) \cap B) \\ \Leftrightarrow\quad & f(A \setminus \{n\}) + f(B) \geq f((A \setminus \{n\}) \cup B) + f(A \cap B) \\ \Leftrightarrow\quad & f(A \cap B) - f(B) \leq f(A \setminus \{n\}) - f((A \setminus \{n\}) \cup B) \qquad (\diamondsuit) \end{aligned}$$

In addition, choosing $S = A \setminus \{n\}, L = B \cup (A \setminus \{n\})$ and applying (2.3) gives us

$$\begin{aligned} & f((A \setminus \{n\}) \cup \{n\}) - f(A \setminus \{n\}) \geq f(B \cup (A \setminus \{n\}) \cup \{n\}) - f(B \cup (A \setminus \{n\})) \\ \Leftrightarrow\quad & f(A) - f(A \setminus \{n\}) \geq f(B \cup A) - f(B \cup (A \setminus \{n\})) \\ \Leftrightarrow\quad & f(A \setminus \{n\}) - f(B \cup (A \setminus \{n\}) \leq f(A) - f(B \cup A) \qquad (\clubsuit) \end{aligned}$$

Combining the two, we get

$$\begin{aligned} & f(A \cap B) - f(B) \\ \leq\ & f(A \setminus \{n\}) - f((A \setminus \{n\}) \cup B) && \text{by } (\diamondsuit) \\ \leq\ & f(A) - f(A \cup B) && \text{by } (\clubsuit) \end{aligned}$$

and we are done. □

2.2 From matrices to matroids

We now look at at a structure called "matroid" that generalizes matrices from linear algebra and explain its relationship with submodular functions.

Consider any matrix with real-valued entries A. From linear algebra we know several key facts about sets of column vectors of A. For example, every subset of a set of independent column vectors is also independent. Also, there are maximal independent sets of column vectors that are called bases. The rank of a matrix is the dimension of the span of its column vectors and thus the size of its bases.

A *matroid* is a structure containing a finite set E and subsets of that set which are called "independent". Independent sets that are inclusion-wise maximal are called "bases". It is also possible to define a rank function on the subsets of E. Thus, matroids are a generalization of the aforementioned concepts from linear algebra.

Definition 2.3 (Matroid)
Let E be a finite set and $\mathcal{I}$ a family of subsets of E, so that

(I1) $\emptyset \in \mathcal{I}$

(I2) $I_1 \subseteq I_2 \in \mathcal{I} \implies I_1 \in \mathcal{I}$

(I3) $I_1, I_2 \in \mathcal{I}$ *and* $|I_1| < |I_2| \implies \exists e \in I_2 \setminus I_1 \colon I_1 \cup \{e\} \in \mathcal{I}$

We call $(E, \mathcal{I})$ a matroid, *E its* ground set *and the $I \in \mathcal{I}$ its* independent sets.

An element $I \in \mathcal{I}$ is called a basis *if there is no $J \in \mathcal{I}$ with $I \subset J$. Thus, the set of bases of a matroid is given by*

$$\mathcal{B} = \{I \in \mathcal{I} \mid \forall J \in \mathcal{I} \colon I \subseteq J \implies I = J\}.$$

Furthermore, we define the rank function *of a matroid to be*

$$p \colon 2^E \to \mathbb{N}, p(X) = \max\{|I| \mid I \in \mathcal{I}, I \subseteq X\}$$

and the rank of the matroid *as $p(E)$.*

Example 2.4
As explained above, we can think of the column vectors of a matrix as elements of the ground set E of a matroid and of the linearly independent sets of these vectors as members of $\mathcal{I}$. For instance

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 3}, \text{ so } E = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix} \right\}.$$

Then $p(E) = 2$ *and*

$$\mathcal{B} = \left\{ \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix} \right\} \right\}.$$

Similar to bases in linear algebra, we can exchange elements between the bases of a matroid to obtain a new basis.

Lemma 2.5
For pairs of bases $B_1, B_2 \in \mathcal{B}$ *of a matroid, it holds that* $|B_1| = |B_2|$ *and for an element* $e_1 \in B_1 \setminus B_2$ *there exists* $e_2 \in B_2 \setminus B_1$ *so that* $(B_1 \setminus \{e_1\}) \cup \{e_2\} \in \mathcal{B}$.

Proof
B_1 and B_2 must be of equal size, otherwise assume $|B_1| < |B_2|$. Then by (I3) there is $e \in B_2 \setminus B_1$ so that $B_1 \cup \{e\} \in \mathcal{I}$, which contradicts that B_1 is a basis.

Now, $|(B_1 \setminus \{e_1\})| < |B_2|$ and again e_2 exists by (I3). □

The next lemma establishes the connection to submodular functions, which were discussed in the previous section. In fact, instead of defining matroids via their independent sets, it is possible to equivalently define matroids via functions satisfying the properties in this lemma, see e.g. [8].

Lemma 2.6
The rank function of a matroid has the following properties:

(Subcardinality) $\forall X \subseteq E \colon 0 \leq p(X) \leq |X|$

(Monotony) $X \subseteq Y \subseteq E \implies p(X) \leq p(Y)$

(Submodularity) p *is submodular*

Proof (see also Lemma 5.1.3 in [8])
(Subcardinality) and (Monotony) clearly hold by the definition of p. Thus, it remains to show that p fulfills the condition in equation 2.1.

To that end, let $A, B \subseteq E$ and take a subset $F \subseteq A \cap B$ that is maximally independent, i.e., $F \in \mathcal{I}$ and there is no $J \in \mathcal{I}$ with $J \subseteq A \cap B$ and $F \subset J$. Likewise, let N' be a maximal independent subset of $A \cup B$. Through repeated applications of (I3), we can add elements of N' to F to obtain a maximal independent subset N of $A \cup B$ with $F \subseteq N$.

For these sets we have $|F| = p(A \cap B)$ and $|N| = p(A \cup B)$ by the definition of p. Observe also that $p(A) \geq |N \cap A|$ since $N \cap A \in \mathcal{I}$, and likewise $p(B) \geq |N \cap B|$.

Now note that $|N \cap A| + |N \cap B| = 2|N \cap A \cap B| + |(N \cap A) \setminus B| + |(N \cap B) \setminus A|$. Furthermore, $N \cap A \cap B = F$. Also, $((N \cap A) \setminus B) \mathbin{\dot\cup} ((N \cap B) \setminus A) = N \setminus F$, since every element of $N \setminus F$ comes from either $A \setminus B$ or $B \setminus A$ (otherwise, it would already be included in F). Thus, we get

$$\begin{aligned} p(A) + p(B) &\geq |N \cap A| + |N \cap B| = 2\,|F| + |N \setminus F| \\ &= |F| + |N| = p(A \cap B) + p(A \cup B). \end{aligned}$$ □

Matroids and similar structures are frequently used in combinatorial optimization. In the next section, we will see that it is possible to use a greedy algorithm to optimize linear objective functions over bases of a matroid.

2.3 Linear optimization over matroids

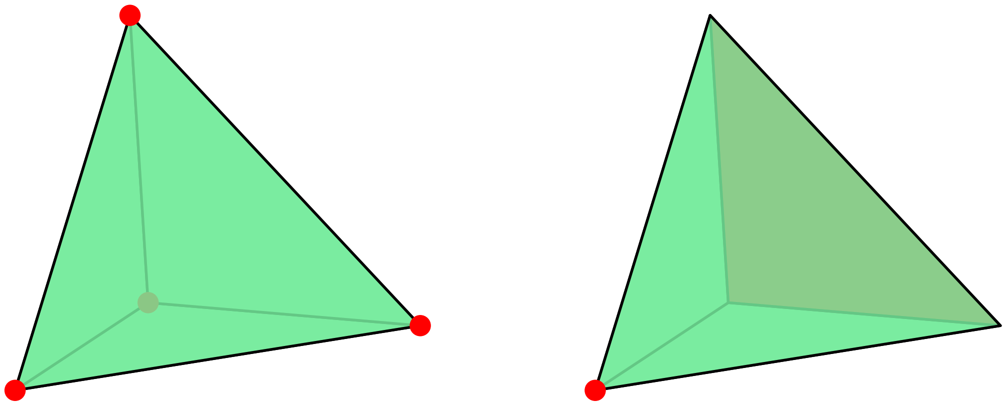
Optimization is the problem of finding some element x of a set of *feasible* elements X that minimizes or maximizes the value of an *objective function* f. Depending on the properties of X and f (e.g. whether elements in X are real- or integer-valued), there are different approaches to solve an optimization problem.

A frequently occurring special case of optimization is *linear programming*, where the set X consists of solutions to linear inequalities and f is a linear function. As there are efficient procedures to solve linear programming problems, they are used in many domains. In particular, linear programs occur frequently in game theory. There is also a geometric interpretation of the feasible elements of linear programs:

Definition 2.7 (Polyhedron and polytope)
Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ a matrix and a vector respectively. Then the set of feasible elements

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

of the associated linear program is called a polyhedron*. A bounded polyhedron is called* polytope.



(a) Bounded case

(b) Unbounded case

Figure 2.1: Two polyhedra. They are defined by the linear inequalities: $x_1 + x_2 + x_3 \leq 1, x_1 \geq 0, x_2 \geq 0$, and for 2.1a only: $x_3 \geq 0$. The redly shaded area in 2.1b illustrates the polyhedron reaching into infinity.

In the remainder of this section, we will look at linear programs whose feasible regions have matroid structure. We begin by motivating this area of focus.

Example 2.8 (Minimum spanning tree)
A well known problem in optimization consists of finding a minimum weight spanning tree for a given connected undirected graph $G = (V, E)$ with weights $w \colon E \to \mathbb{N}$ associated with each edge.

In terms of our formulation of the optimization problem above, the set of feasible elements X is the set of all spanning trees of G (i.e., all connected, circle-free subgraphs $(V, S \subseteq E)$ so that for all $v \in V$ there exists some $e = \{p, q\} \in S$ with $v = p$ or $v = q$) and we want to minimize $f((V, S)) = \sum_{e \in S} w(e)$ (the sum of weights of the edges in the spanning tree) over all spanning trees $(V, S) \in X$.

We know that the problem can be solved by Kruskal's algorithm *(see e.g. [3]), which starts with an empty set S and iteratively adds edges of lowest possible weight so that the resulting graph remains circle-free. The algorithm terminates after adding $|V| - 1$ edges with a minimum weight spanning tree of G. See Figure 2.2 for an example.*

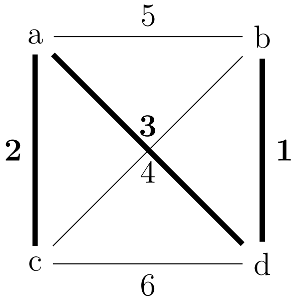


Figure 2.2: A graph with a minimum spanning tree. Kruskal's algorithm adds, in order, the edges of weight 1, 2 and 3. Afterwards it terminates, as adding another edge would introduce a circle in the subgraph. Thus, the total sum of weights in the spanning tree is 6.

Kruskal's algorithm is an example of a greedy algorithm*, since in each iteration it chooses a locally optimal element (the edge with lowest cost among all remaining edges) to ultimately arrive at a globally optimal solution (a minimum weight spanning tree).*

We also observe that the minimum spanning tree problem has matroid structure: Take E to be the the ground set of the matroid and choose as members of $\mathcal{I}$ the edge sets of all subgraphs of $G = (V, E)$ that are circle free. Then the maximal independent sets of $(E, \mathcal{I})$ correspond to spanning trees of G. Finding a minimum weight spanning tree of G thus translates to the problem of finding a minimum weight basis of $(E, \mathcal{I})$.

Generalizing from the previous example, we may consider the problem of optimizing a linear function over the bases of a matroid, namely:

Problem 2.9 (Linear optimization over bases of a matroid)
Let $(E, \mathcal{I})$ be a matroid with bases $\mathcal{B}$ and $w\colon E \to \mathbb{R}$ a weight function. Find a base $B \in \mathcal{B}$ such that $\sum_{e \in B} w(e)$ is minimal.

Note that Problem 2.9 can be written as an integer linear program: the objective function $B \mapsto \sum_{e \in B} w(e)$ is a linear map of the characteristic vector $\chi_B \in \{0,1\}^E$ of B (where $(\chi_B)_e = 1$ if $e \in B$ and $(\chi_B)_e = 0$ otherwise). Additionally, we can reformulate the constraint that feasible solutions are bases of a matroid as linear inequalities so that the feasible region becomes

$$\left\{ x \in \{0,1\}^E \;\middle|\; \sum_{e \in X} x_e \leq p(X) \;\forall X \subset E, \sum_{e \in E} x_e = p(E) \right\}, \tag{2.4}$$

where p is the rank function of $(E, \mathcal{I})$.

Thus, it is not immediately obvious how Problem 2.9 can be solved efficiently. However, it turns out that there is a very elegant solution thanks to the problem's matroid structure, see Algorithm 1.

Algorithm 1 Greedy algorithm for Problem 2.9, see e.g. section 5.5.2 in [8]

Input: $(E, \mathcal{I})$ a matroid, $\mathcal{B}$ its bases and $w\colon E \to \mathbb{R}$ an associated weight function
Output: $B \in \mathcal{B}$ with $\sum_{e \in B} w(e)$ minimal

```
1: $B \Leftarrow \emptyset$
2: $[e_1, \dots, e_n] \Leftarrow$ a list of all elements in $E$ such that $w(e_1) \le w(e_2) \le \dots \le w(e_n)$
3: $i \Leftarrow 1$
4: while $i \le n$ do
5:     if $B \cup \{e_i\}$ is independent then
6:         $B \Leftarrow B \cup \{e_i\}$
7:     end if
8:     $i \Leftarrow i + 1$
9: end while
```

Theorem 2.10 (Correctness of the matroid greedy algorithm)
The output B of algorithm 1 is a basis of $(E, \mathcal{I})$ of minimal weight.

Proof (see also Theorem 40.1 in [22])
The algorithm only adds elements such that the resulting set remains independent (step 5). When the loop terminates, by (I2) there is no element that may be added to retain independence. Thus there is no strict superset of B also in $\mathcal{I}$, so B is a basis.

It remains to show optimality. Denote by B_i the set B at the beginning of the ith iteration of the algorithm. Furthermore, let $\mathcal{B}_{min}$ be the set of bases of minimal weight. We will show that if $B_i \subseteq B_{min}$ for a $B_{min} \in \mathcal{B}_{min}$ and $1 \le i < n$, then also $B_{i+1} \subseteq B'_{min}$ for some $B'_{min} \in \mathcal{B}_{min}$. This will give us, with $i = n - 1$, that B is of minimal weight.

Thus, fix some i and choose a $B_{min} \in \mathcal{B}_{min}$ so that $B_i \subseteq B_{min}$. If $B \cup \{e_i\}$ is not independent, then $B_i = B_{i+1} \subseteq B_{min}$. If, on the other hand, $e_i \in B_{min}$, then $B_i \cup \{e_i\} = B_{i+1} \subseteq B_{min}$ and we have the claim. Otherwise, $e_i \notin B_{min}$ and by repeatedly applying (I3) (starting with $I_1 = B_i \cup \{e_i\} = B_{i+1}, I_2 = B_{min}$ and iterating until I_1 cannot be enlarged anymore) we get a basis $B' \supseteq B_{i+1}$. By its construction, there is an element $x \in B_{min} \setminus B_i$ such that $B' = B_{min} \setminus \{x\} \cup \{e_i\}$. But e_i has minimal weight among elements that can be added to B_i while preserving independence, so $w(e_i) \le w(x)$ and thus $\sum_{e \in B'} w(e) \le \sum_{e \in B_{min}} w(e)$. Therefore, B' is a basis of minimal weight and we are done. □

Regarding runtime, the only loop in Algorithm 1 runs exactly $|E|$ times but requires an oracle to check for independence in step 5 and a list of elements in E sorted by weight. A similar algorithm can be used to find a basis of maximal weight: in that case, the list in step 2 should be in non-increasing order. Also note that the problem of finding minimal (maximal) independent sets can be solved by deleting elements with positive (negative) weight and then applying Algorithm 1.

2.4 From matroids to polymatroids

There are many generalizations and variations of matroids that can be obtained e.g. by modifying the constraints in the independence axioms or by using different kinds of rank functions. In this section, we look at polymatroids, which are structures where the subcardinality of the rank functions is relaxed.

Definition 2.11 (Polymatroid axioms)
Let E be a finite set and $p\colon 2^E \to \mathbb{R}_+$ a function over subsets of E satisfying

(Normalization) $p(\emptyset) = 0$

and (Monotony) and (Submodularity) from Lemma 2.6.
We call (E, p) a polymatroid *and p its* rank function*. If p is in fact integer-valued (so $p\colon 2^E \to \mathbb{N}$), we call (E, p) an* integral polymatroid*.*

We will assume that the rank function is given by an oracle, i.e., some efficient procedure that calculates $p(X)$ for $X \subseteq E$. Although we introduced polymatroids through their rank functions, the concepts of independence and bases from matroids still carry over in some way. The rank function can be used to define a polyhedron, similarly to the one describing the feasible solutions of linear optimization over matroid bases (see equation 2.4). We consider elements of that polyhedron to be independent and elements on a certain extreme face of the polyhedron to be bases (see Figure 2.3 for an illustration).

From now on, we will use a shorthand notation common in literature, where for a vector $x = (x_e)_{e\in E} \in \mathbb{R}^E$ and a subset $X \subseteq E$ we write $x(X)$ for the sum of vector elements corresponding to members of X, so

$$x(X) = \sum_{e\in X} x_e.$$

Definition 2.12 (Polymatroid (base) polytope)
Let (E, p) be a polymatroid. The polymatroid polytope *associated with (E, p) is the set*

$$P(p) = \left\{x \in \mathbb{R}_+^E \;\middle|\; x(X) \le p(X)\ \forall X \subseteq E\right\}$$

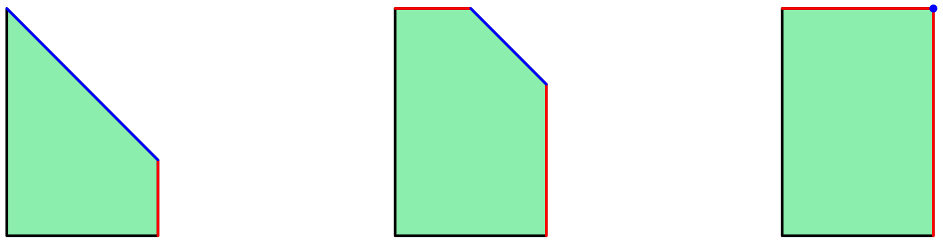
and the corresponding polymatroid base polytope *is*

$$B(p) = \{x \in P(p) \mid x(E) = p(E)\}.$$

If (E, p) is an integral polymatroid, the sets $P(p)$ and $B(p)$ are instead defined with vectors of natural numbers $x \in \mathbb{N}^E$.

Note that $P(p)$ and $B(p)$ are upper bounded by the rank function and lower bounded by 0 in all dimensions, so they are indeed polytopes.

There is also something akin to the basis exchange property from matroids for the components of a vector in the polymatroid polytope. For some such vectors,



(a) $x_1 + x_2 \leq 3$ **(b)** $x_1 + x_2 \leq 4$ **(c)** $x_1 + x_2 \leq 5$

Figure 2.3: Three polymatroid polyhedra and their associated base polytopes. They are obtained from the inequalities: $0 \leq x_1 \leq 2, 0 \leq x_2 \leq 3$ together with the inequality given under the respective subfigure. Areas of all colors constitute the polymatroid polytope, while the base polytopes are marked blue.

there are components whose value cannot be increased without exceeding the bounds of the polytope. These components are called *saturated*. They may only be increased after the value of another component is reduced. For example, in the polytopes in Figure 2.3, any point in the greenly shaded areas may be moved slightly up or to the right (thus having its second or first component increased) without leaving the bounds of the polytope. Points on the red lines have one saturated component, while points on the blue lines (the base polytopes) are saturated in all their components.

Definition 2.13 (Saturated components, exchangeable pairs)
Let (E, p) be a polymatroid, $P(p)$ the associated polytope and $x \in P(p)$ a vector in the polytope. Additionally, denote by $\chi_e \in \mathbb{R}_+^E$ the characteristic vector for an element $e \in E$ (so $(\chi_e)_e = 1$ and $(\chi_e)_f = 0$ for all $e \neq f \in E$). Then the set of saturated components *of x is given by*

$$\operatorname{sat}(x) = \{e \in E \mid x + \alpha\chi_e \notin P(p) \ \forall \alpha > 0\}.$$

For x and $e \in \operatorname{sat}(x)$, the set

$$\operatorname{dep}(x, e) = \{e' \in E \setminus \{e\} \mid \exists \alpha > 0 \colon x + \alpha(\chi_e - \chi_{e'}) \in P(p)\}$$

contains all $e' \in E$ so $x_{e'}$ can be reduced to increase x_e. The tuples (e, e'), $e' \in \operatorname{dep}(x, e)$, are called exchangeable pairs *associated with x.*

By definition, $\operatorname{sat}(x) = E$ for a vector $x \in B(p)$, so all components of a vector in the basis polytope are saturated. Moving from x to another $x' \in B(p)$ therefore requires reallocating some value between the exchangeable pairs of x. The following lemma states that for two bases of a polymatroid, mutually exchangeable pairs of components always exist. We refer to [17] for its proof.

Lemma 2.14 (Lemma 3.17 in [17], see also [16])
Let (E, p) be a polymatroid and $x, y \in B(p)$ two vectors in its base polyhedron. Then for all components $e \in E$ with $x_e > y_e$, there exists a component $f \in E$ with $x_f < y_f$ and an $\alpha \in \mathbb{R}_{>0}$ such that

$$x - \alpha(\chi_e + \chi_f) \in B(p) \text{ and } y + \alpha(\chi_e - \chi_f) \in B(p).$$

Polymatroid polytopes have some very desirable properties in optimization. As for matroids, it is possible to optimize a linear function over a polymatroid using a greedy algorithm, as we shall see shortly. Later, we will encounter optimization problems over polymatroids with convex separable objective functions. These can also be solved efficiently using an algorithm described in section 2.5.

The final theorem in this section contains a result on the complexity of computing an element of a polymatroid base polytope. This will be relevant when we discuss core allocations in cooperative games. The theorem is also essential for the polymatroid greedy algorithm.

Theorem 2.15
Let (E, p) be a polymatroid, $|E| = n$, and $\pi = (e_1, \ldots, e_n)$ any permutation of elements in E. Also, write $E_i^\pi = \{e_1, \ldots, e_i\}$ for all $1 \leq i \leq n$ and $E_0^\pi = \emptyset$. Define a vector $x \in \mathbb{R}^E$ by

$$x_{e_i} = p(E_i^\pi) - p(E_{i-1}^\pi)$$

for all $1 \leq i \leq n$. Then $x \in B(p)$.

Proof (see also Lemma 5.5.10 in [8])
Due to (Normalization) and (Monotony), we have that p is nonnegative and nondecreasing and thus $x_e \geq 0$ for all $e \in E$. Additionally, we obtain

$$x(E) = \sum_{i=1}^{n} x_{e_i} = \sum_{i=1}^{n} p(E_i^\pi) - p(E_{i-1}^\pi) = p(E) - p(\emptyset) = p(E),$$

where the final equality follows from (Normalization).

Thus, it remains to show that $x(X) \leq p(X)$ for all $X \subset E$. We proceed by induction on the size of X. To begin, let $|X| = 1$. Then $X = \{e_j\}$ for some $1 \leq j \leq n$ and

$$x(X) = x_{e_j} = p(E_j^\pi) - p(E_{j-1}^\pi) \leq p(\{e_j\}) - p(\emptyset) = p(X),$$

where the inequality follows from the submodularity of p (cf. equation 2.3).

Now, let $|X| = i + 1$, $i \geq 1$, and choose the subset $X' = X \setminus \{e_j\}$ obtained by removing from X the element e_j with the largest index j among its elements. Thus, $|X'| = i$ so that we can apply the induction hypothesis (IH): $x(X') \leq p(X')$. We get

$$x(X) = x(X') + x_{e_j} \overset{(IH)}{\leq} p(X') + x_{e_j} = p(X') + p(E_j^\pi) - p(E_{j-1}^\pi).$$

But note that

$$p(X') + p(E_j^\pi) \leq p(X) + p(E_{j-1}^\pi)$$

by the submodularity of p (cf. equation 2.1, choosing $A = X$ and $B = E_{j-1}^\pi$) and we are done. □

Theorem 2.15 gives an efficient algorithm to compute one of up to $|E|!$ (the number of permutations of E) vectors in the polymatroid base polytope: Provided we can compute $p(X)$ for all $X \subseteq E$ efficiently, we can compute the elements x_{e_i} one by one (potentially using dynamic programming) after fixing a permutation π. Both operations take time linear in $|E|$. We will use this result later when discussing core allocations for cooperative games with submodular cost functions.

The theorem also admits an algorithm that can be used for linear optimization over a polymatroid base polytope. Let $w\colon E \to \mathbb{R}$ be a weight function and $\sum_{e\in E} w(e)x_e$ shall be minimized for $x \in B(p)$. Now let $\pi = (e_1, \dots, e_n)$ be a permutation of E, such that $w(e_1) \leq \cdots \leq w(e_n)$. It can be shown that the vector $x^* \in B(p)$ constructed with π according to the formula in Theorem 2.15 is optimal for this minimization problem. The proof proceeds by constructing an appropriate dual problem and a corresponding feasible solution y so that the respective objective values for x^* and y are equal, see e.g. [8] or [22].

Intuitively, the algorithm is "greedy", since for all $1 \leq i \leq n$ it maximizes $x^*(X)$ for the cheapest subsets $X \subseteq E$ of size $|X| = i$. Or in other words: a total of $p(E)$ must be assigned to x^*, so the algorithm allocates as much as possible to components x^*_e with low $w(e)$ while preserving the submodularity constraints $x(X) \leq p(X)$ for all $X \subseteq E$.

Furthermore, by ordering the elements of E by decreasing weight, the same algorithm can be used for maximization. When optimizing over the whole polymatroid polytope (as opposed to the base polytope), one can ignore $e \in E$ with negative or positive weight (for maximization or minimization respectively) and then apply the algorithm.

2.5 Convex separable minimization over polymatroids

In later chapters, we will encounter minimization problems involving convex separable functions and polymatroid base polytopes. This means that the set of feasible elements is a polymatroid base polytope and the objective function is a sum of univariate convex cost functions per element of the ground set. Lemma 2.16 states the problem formally.

The so-called "decomposition algorithm" introduced in [11] can be used to maximize a concave separable function over a polymatroid base polytope efficiently. Minimizing a convex separable function can be done similarly, since a function f is minimized by maximizing $-f$ and f is convex if and only if $-f$ is concave.

We now briefly give the optimality conditions for minimizing a convex separable function over a polymatroid and sketch the decomposition algorithm. For a rigorous explanation, see [11] and [9].

Lemma 2.16 (Optimality conditions, Theorem 8.1 in [9])
Let (E, p) be a polymatroid, $c_e\colon \mathbb{R}_+ \to \mathbb{R}$ convex functions for all $e \in E$ and c_e^+, c_e^- their respective right and left derivatives. Define by

$$f\colon \mathbb{R}_+^E \to \mathbb{R}, f(x) = \sum_{e\in E} c_e(x_e)$$

a convex separable function to be minimized for elements $x \in B(p)$.

Then $x^* \in B(p)$ is optimal if and only if for all exchangeable pairs (e, f) associated with x^* it holds that

$$c_f^-(x_f^*) \leq c_e^+(x_e^*).$$

It can be easier to think of c_e as being differentiable, so that c_e^+ and c_e^- coincide. Note that even if c_e is not differentiable, the right and left derivatives always exist because c_e is a convex function in one variable (see e.g. [21]). Intuitively, Lemma 2.16 says that x^* is optimal when the objective function cannot be decreased by transferring some amount from one vector component of x^* to another (while staying inside $B(p)$). Clearly, the condition implies that x^* is a local optimum. For the proof of global optimality we refer to [11] and [9].

The decomposition algorithm can be used to find x^*. Roughly speaking, the algorithm proceeds by a divide-and-conquer approach: Firstly, f is minimized for $y \in \mathbb{R}_+^E$ under the sole constraint $y(E) \leq p(E)$. The algorithm finds a maximal element $v \in B(p)$ such that $v \leq y$ and proceeds to partition E into saturated and non-saturated components of v. For this partition, two subproblems are constructed by restricting f to the saturated and non-saturated components respectively and the algorithm is then applied recursively. The recursion ends when there are only saturated components in a subproblem.

This algorithm runs in time polynomial in $|E|$, provided the single constraint problem in the first recursive step can be solved efficiently. For the full statement and analysis of the algorithm we refer again to [11] and [9].

Chapter 3

From Congestion Games to Polymatroid Games

Congestion games provide a mathematical model of how self-interested actors, called players, compete for resources. In this chapter, we introduce the game theoretic framework for congestion games and a special case related to matroid theory called matroid congestion games. Additionally, we discuss some major results for matroid congestion games and a generalization called polymatroid games, involving sensitivity analysis over polymatroids.

The concepts and results in this chapter are from various sources which are indicated at the beginning of each section.

3.1 Non-cooperative games

This section includes the necessary background from game theory for the following discussion. The material is found in standard textbooks on the topic, e.g. [18]. To begin, we formalize the intuitive notion of "game" and illustrate the formalization with some examples.

Definition 3.1 (Strategic game)
A strategic game *consists of a tuple* $\Gamma = (N, (S_i)_{i\in N}, (u_i)_{i\in N})$ *where* N *is a finite set of* $|N| = n$ players *participating in the game and* S_i *is a finite set of* strategies *available to player* i*. Denote by* $S = S_1 \times \cdots \times S_n$ *all combinations of strategies of all players. Then* $u_i : S \to \mathbb{R}$ *describes player* i*'s* payoff *under each combination.*

A match *of the game corresponds to choices* $s = (s_1, \ldots, s_n) \in S$ *for all players and* $u_i(s)$ *is player* i*'s payoff after the match. We assume that players aim to maximize their payoff.*

Example 3.2 (Rock, Paper, Scissors)
The popular "Rock, Paper, Scissors" provides a simple example of how a real-life game can be represented as a strategic game.

There, we have $N = \{1, 2\}$ *players who can choose among the same strategies* $S_1 = S_2 = \{\text{Rock}, \text{Paper}, \text{Scissor}\}$*. Payoffs are assigned symmetrically: if the players choose the same strategy, they both get payoff* 0*. Otherwise, rock beats scissor, paper* *beats rock and scissor beats paper, meaning that the winning player gets payoff* 1 *and their opponent* -1. *For instance,* $u_1((\text{Paper}, \text{Rock})) = 1$.

Example 3.3 (Prisoner's dilemma)
In this famous example, two prisoners are being questioned separately and have the option to either cooperate with their partner and stay silent or defect by confessing their crimes. A prisoner that confesses may get a shorter sentence if their partner stays silent, but if both prisoners cooperate their sentences are shorter than if they both confess.

Formally, we again have $N = \{1, 2\}$ *players who can choose among strategies:* $S_1 = S_2 = \{\text{Cooperate}, \text{Defect}\}$. *The following table gives the payoffs to each prisoner in each possible match. Entries* (i, j) *correspond to payoff* i *for player* 1 *and payoff* j *for player* 2. *We can interpret a payoff* $-y$ *as a sentence of* y *years. For example,* $u_2((\text{Defect}, \text{Cooperate})) = -5$, *meaning that prisoner 2 is sentenced to* 5 *years if they cooperate and prisoner 1 defects.*

		Prisoner 1	
		Cooperate	Defect
Prisoner 2	Cooperate	*(-2,-2)*	*(-1,-5)*
	Defect	*(-5,-1)*	*(-4,-4)*

In the previous Example 3.3, the players can always increase their individual payoff by switching their own strategy (assuming their partner's strategy stays the same), unless both choose the "Defect" strategy. This motivates the concept of an *equilibrium*: if both prisoners confess, neither prisoner has an incentive to cooperate with their partner, because it will decrease their own payoff.

Observe that in Example 3.2, no pair of strategies results in such an equilibrium, because there is always one player who can increase their payoff by switching strategies. To adequately define an equilibrium for Example 3.2, the players must be allowed to play randomized strategies, i.e., to choose a strategy according to some probability distribution over their S_i.

Definition 3.4 (Mixed strategy, best response, Nash equilibrium)
A mixed strategy *for player* i *is a probability distribution* $x^i = (x^i_s)_{s \in S_i}$ *over* S_i. *So* $x^i_s \geq 0$ *is the probability of player* i *playing* $s \in S_i$ *and* $\sum_{s \in S_i} x^i_s = 1$. *A mixed strategy* x^i *is called* pure, *if* $x^i_s = 1$ *for some* $s \in S_i$ *and* $x^i_w = 0$ *for all other* $s \neq w \in S_i$.

Let X_i *be the set of all mixed strategies of player* i *and* $X = X_1 \times \cdots \times X_n$ *the set of all* profiles *of mixed strategies. The* expected payoff *for player* i *under profile* $x \in X$ *is given by the sum of payoffs for player* i *of all matches multiplied by their probability under* x, *so*

$$U_i(x) = \sum_{s=(s_1,\ldots,s_n) \in S} \left(u_i(s) \cdot \prod_{j \in N} x^j_{s_j} \right).$$

A strategy x^i is called a best response *for player i to a profile of opponents' strategies $(x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^n)$ if there is no better alternative strategy that player i could choose, meaning that for all $y^i \in X_i$,*

$$U_i((x^1, \dots, x^i, \dots, x^n)) \geq U_i((x^1, \dots, y^i, \dots, x^n)).$$

A profile $x \in X$ is called Nash equilibrium *if every player's strategy is a best response to their opponents' strategies in that profile. A Nash equilibrium is called pure, if every player's strategy in the equilibrium is pure.*

Intuitively, a Nash equilibrium is a profile where no player can increase their payoff by switching to another strategy. It is not hard to see that for Example 3.2, a Nash equilibrium consisting of mixed strategies is given by the profile in which both players play each of their strategies with equal probability $\frac{1}{3}$. Indeed, a seminal result by Nash states that such an equilibrium always exists. However, computing it may be hard.

Theorem 3.5 (Nash's Theorem)
Every strategic game has at least one Nash equilibrium.

3.2 Atomic congestion games

In the remainder of this chapter, we focus on non-cooperative games where players compete for resources. We begin with a particular type of these, called *atomic congestion game*. See again [18] and also [20]. Intuitively, in an atomic congestion game, players choose between different combinations of resources and each resource has a cost proportional to the number of all players that claim it. The game is called "atomic", because players are elements of finite sets. We will see another type of congestion game later.

Definition 3.6
An atomic congestion game is a tuple $\Gamma_{\text{ACG}} = (N, E, (S_i)_{i \in N}, (c_e)_{e \in E})$ where N is a finite set of players, E a finite set of resources*, strategies are sets of resources (so $S_i \subseteq 2^E$ for every player i) and $c_e \colon \mathbb{N} \to \mathbb{Z}$ is a* cost function *for resource $e \in E$.*

A profile of pure strategies $s = (s_1, \dots, s_n) \in S_1 \times \dots \times S_n$ admits a congestion

$$n_e(s) = |\{i \mid e \in s_i\}|$$

on each resource $e \in E$, i.e., exactly the number of players that use e in profile s. Now, $c_e(n_e(s))$ is the cost of using resource e in one's strategy under profile s.

The payoff u_i for player i is the negative total sum of costs for each resource in their chosen strategy, so $u_i(s) = -\sum_{e \in s_i} c_e(n_e(s))$.

Example 3.7 (Network congestion games)
We can use congestion games to model the interaction of actors who want to find the fastest path through a network. The network is given by a directed graph $G = (V, E)$.

Each player has an origin and destination node in V and tries to find paths of minimum cost between these nodes. Thus, edges correspond to resources and strategies are paths in the graph. Every edge has an associated cost that depends on the congestion on that edge. Figure 3.1 shows an example network.

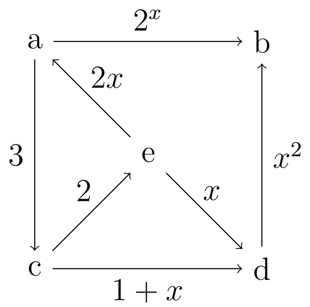


Figure 3.1: A network congestion game. The edges are labeled with their cost as a function of the congestion on that edge. Players may have different origins and goals and choose paths between them. For instance, some player may want to reach d from a and another b from e. If two players use the edge $e \to d$, the congestion on that edge is 2.

Network congestion games can be used to model various real world problems like traffic control or centralized package routing in computer networks.

The following central result states that we can find a pure Nash equilibrium for any atomic congestion game by repeatedly applying *improvement steps*. Such a step for a profile of pure strategies $s = (s_1, \dots, s_n) \in S_1 \times \cdots \times S_n$ is a new strategy $s'_i \in S_i$ for some player i, so that i strictly reduces their total cost (i.e., increases their payoff) by playing the new strategy over their previous choice. Formally: $u_i(s) < u_i((s_1, \dots, s'_i, \dots, s_n))$. Note that the improvement step that yields minimal total cost for player i is, by definition, a best response.

Theorem 3.8 (Sequences of improvement steps)
Every sequence of improvement steps in every congestion game is finite and terminates in a pure Nash equilibrium.

Proof (Sketch, see also [20] and [18])
For profiles of pure strategies $s \in S_1 \times \cdots \times S_n$, define a potential function

$$\Psi(s) = \sum_{e \in E} \sum_{i=1}^{n_e(s)} c_e(i). \tag{3.1}$$

It can be shown that each improvement step for any player reduces the value of the potential function by at least 1. Note that $S_1 \times \cdots \times S_n$ is finite, so Ψ has a minimum. Therefore, any sequence of improvement steps must be finite. Once no more improvement steps are possible, the game has - by definition - reached a pure Nash equilibrium. □

Computing a Nash equilibrium through improvement steps may take a long time, because of the number of such steps required. It can be shown (see [6]), that finding pure Nash equilibria in atomic congestion games through improvement steps is as hard as solving any problem in a class called "local search problems". For example, finding a maximum cut in a graph by repeatedly exchanging nodes between the two sets constituting the cut also belongs to this class. Importantly, through reductions to certain local search problems, it is possible to construct a congestion game for which any sequence of improvement steps has exponential length (see [6] and [1]). In the next section, however, we will see special congestion games where best responses (as special cases of improvement steps) quickly converge to Nash equilibria.

3.3 Matroid congestion games

There is a special type of atomic congestion games related to matroid theory, defined as follows:

Definition 3.9
An atomic congestion game is a matroid congestion game if for each player $i \in N$ the set of strategies S_i forms the bases of a matroid $(E, \mathcal{I}_i)$, so $S_i = \mathcal{B}$ from Definition 2.3.

In this section, we refer to [1], where these games have originally been studied. An important result given in this work is that the basis exchange property of matroids allows to upper bound the length of a sequence of best responses polynomially in the number of players and resources:

Theorem 3.10 (Theorem 2.5 in [1])
Let Γ_{ACG} be a matroid congestion game with n players and $|E| = m$ resources. Then players reach a Nash equilibrium after at most n^2m^2 best response improvement steps.

Proof
We consider the potential function from equation 3.1 with regard to alternative cost functions $\tilde{c_e}$: Construct a list of all possible cost values for all c_e and sort the list in non-decreasing order. Specifically, the list contains all distinct values $c_e(i)$ for $e \in E$ and $1 \leq i \leq n$, which are sorted non-decreasingly. Then $\tilde{c_e}(i)$ is defined to be the index of $c_e(i)$ in that list. Now, $\tilde{c_e}(i)$ is higher when $c_e(i)$ appears later in the list. Intuitively, $\tilde{c_e}(i)$ gives the "rank" of $c_e(i)$ relative to all possible costs.

Note that the list of cost values has length at most nm, the alternative cost functions are positive-valued and the congestion on any edge can not exceed the number of players. Using these observations, we can bound the potential function with regard to the alternative costs for profiles $s \in S_1 \times \cdots \times S_n$:

$$0 \leq \tilde{\Psi}(s) = \sum_{e \in E} \sum_{i=1}^{n_e(s)} \tilde{c_e}(i) \leq \sum_{e \in E} \sum_{i=1}^{n_e(s)} nm \leq n^2m^2.$$

To complete the proof, it remains to show that each best response improvement step decreases $\tilde{\Psi}$ by at least 1. To that end, fix a profile $s = (s_1, \dots, s_n)$ that is not a Nash equilibrium, let $s'_i \in S_i$ be a best response to s for player i and write $s' = (s_1, \ldots, s'_i, \ldots, s_n)$. Then for all $e \in s_i \setminus s'_i$ and $e' \in s'_i \setminus s_i$ it holds that

$$c_{e'}(n_{e'}(s')) \leq c_e(n_e(s))$$

since otherwise, by Lemma 2.5, player i could choose another strategy $s'_i \setminus \{e'\} \cup \{e\} \in S_i$ with lower costs and s'_i would not be a best response. Additionally, for at least one pair (e, e'), the inequality must be strict, because the best response strictly decreases the total costs for player i.

Thus, we have that the best response decreases costs c_e. The alternative cost functions $\tilde{c}_e$ are only the "ranks" of the original cost functions and therefore decrease as well. To conclude, the best response must decrease the potential $\tilde{\Psi}$. But since the potential is bounded from below by 0 and from above by n^2m^2, this means that any sequence of best responses has length at most n^2m^2. □

Example 3.11 (Minimum spanning tree congestion game)
Recall Example 2.8. There, we saw that the set of all spanning trees in a connected, undirected graph forms the bases of a matroid over the edge set of that graph.

We can imagine the following atomic congestion game: n players each choose a spanning tree in the graph and want to minimize their total cost, which is the sum of costs per edges in their chosen tree. The edge costs are functions of the congestion on the respective edge. So after all players have chosen their spanning tree, it might be beneficial for some players to switch to another tree. Now Theorem 3.10 tells us that the number of such best responses can be bounded polynomially in the number of players and edges in the graph.

3.4 Nonatomic congestion games

Often, we are not interested in the conduct of a single actor in a system but rather in the behavior of large groups of actors. As such, we can analyze player populations instead of individual players.

Consider, for example, a network congestion game where the network corresponds to a road map of a large city and there are hundreds of thousands of drivers, some of whom have the same origins and goals. Now, instead of analyzing best response dynamics between hundreds of thousands of players, we can think of all drivers with the same start and end points as belonging to the same population. The strategies available to a population are paths from their origins to their destinations. The exact number of drivers in a population may not matter, but the relation between population sizes does. We can analyze these scenarios as nonatomic congestion games. These games are described e.g. in [18] and [10], which we refer to.

Definition 3.12
A nonatomic congestion game is a tuple $\Gamma_{\text{NCG}} = (N, E, (S_i)_{i \in N}, (c_e)_{e \in E}, (d_i)_{i \in N})$ where N is now a finite set of populations. *Players in the same population $i \in N$ share the same set of strategies $S_i \subseteq 2^E$, $d_i \geq 0$ can be interpreted as the* demand *or size of population i and we require for the strategy distribution $x^i = (x^i_s)_{s \in S_i}$ of population i that $\sum_{s \in S_i} x^i_s = d_i$, meaning that players in a population may choose different strategies, but exactly the total demand is allocated among the chosen*

strategies. We require the cost functions $c_e \colon \mathbb{R}_+ \to \mathbb{R}_+$ *for resources* $e \in E$ *to be continuous and non-decreasing.*

Let $x = (x^i)_{i \in N}$ *be the strategy distributions of all populations. The congestion on a resource* $e \in E$ *is the sum of the demands that all populations allocate to that resource, so*

$$n_e(x) = \sum_{i \in N} \sum_{s \in S_i \colon e \in s} x^i_s.$$

Similar to atomic congestion games, the cost of using resource e *under distributions* x *is* $c_e(n_e(x))$ *and the total payoff of playing a strategy* $s \in 2^E$ *is* $-\sum_{e \in s} c_e(n_e(x))$*. Therefore, populations want to allocate their demand so that the strategies they use cause as little cost as possible.*

The notion of matroid congestion games from Definition 3.9 carries over to the nonatomic case: the game has matroid structure if each strategy set is the set of bases of some matroid.

As populations are continuums $[0, d_i]$, every player in a nonatomic congestion game is infinitesimally small. Thus, we cannot directly apply the terms from Definition 3.4 to analyze equilibrium situations for these games. Instead, we define another type of equilibrium:

Definition 3.13 (Wardrop equilibrium)
A strategy distribution $x = (x^i)_{i \in N}$ *of the populations in a nonatomic congestion game is called a* Wardrop equilibrium *if for all populations* $i \in N$ *all strategies* $s \in S_i$ *that are actually in use by that population (meaning* $x^i_s > 0$*) have minimum cost, therefore*

$$\sum_{e \in s} c_e(n_e(x)) \leq \sum_{e \in s'} c_e(n_e(x))$$

for all $s' \in S_i$*.*

These types of equilibria were first introduced for road planning tasks (akin to the example given at the beginning of this section, see also [19]). Their connection to game theory was established later, for example in [5]. There, it is shown that a traffic network in Wardrop equilibrium corresponds to a strategic game in Nash equilibrium where players are pairs of origin and destination nodes in the network and their strategies are a continuum of possible demand allocations between these nodes.

Another analogy between Wardrop and Nash equilibria is given e.g. in [19]. There, Nash equilibria are defined for nonatomic congestion games as follows: the game is in equilibrium if no (possibly infinitesimally small) amount of players (demand) can be reallocated to another strategy without increasing their cost. It can be shown that such Nash equilibria are equivalent to Wardrop equilibria for games with continuos, non-negative and non-decreasing cost functions (as in Definition 3.12).

For atomic congestion games, we presented an efficient method to compute a Nash equilibrium. We have not yet discussed how a Wardrop equilibrium may be computed. It turns out that this is possible by solving an optimization problem:

Theorem 3.14 (Beckmann potential)
For a strategy distribution x of the populations in a nonatomic congestion game we define the Beckmann potential

$$\Phi(x) = \sum_{e \in E} \int_0^{n_e(x)} c_e(t)dt. \tag{3.2}$$

Then x is a Wardrop equilibrium if and only if it minimizes $\Phi(x)$.

For the proof, we refer to [2]. Intuitively, the Beckmann potential function is a continuos analogue of the Rosenthal potential (equation 3.1). If a population shifts some amount of their demand to a strategy with lower costs, then the potential decreases. Once this is no longer possible, the strategy distribution is, by definition, a Wardrop equilibrium.

The Beckmann potential is a sum of terms which are integrals over non-decreasing functions c_e and thus convex. Therefore, finding a Wardrop equilibrium reduces to minimizing a convex separable function. For nonatomic matroid congestion games, it is shown in [10] that the set of all strategy distributions x can be represented as a polymatroid base polytope. Therefore, the decomposition algorithm from [11] (see section 2.5) can be used to minimize the Beckmann potential and find a Wardrop equilibrium in a nonatomic matroid congestion game.

To see this, we will present the argument from [10]. Assume we have a nonatomic matroid congestion game, so the strategy set S_i of population i is exactly the bases $\mathcal{B}_i$ of a matroid $(E, \mathcal{I}_i)$ with rank function p_i. A strategy distribution x^i induces a congestion vector $y^i = (y^i_1, \ldots, y^i_m) \in \mathbb{R}^E_+$ with entries $y^i_e = \sum_{s \in S_i :\, e \in s} x^i_s$ for all resources $e \in E$ (meaning, the congestion on resource e created by population i). The set of all possible congestion vectors created by population i is a polymatroid base polytope

$$P_i = \{y^i \in \mathbb{R}^E_+ \mid y^i(X) \leq d_i \cdot p_i(X) \; \forall X \subseteq E, y^i(E) = d_i \cdot p_i(E)\}.$$

The inequalities are obtained, because the congestion on a subset of resources $X \subseteq E$ can be at most the total demand of population i multiplied by the maximal number of resources that the total demand can be imposed on (which is $p_i(X)$, the maximum size of an independent subset of X). The polytope P_i is indeed a polymatroid base polytope, because p_i is a matroid rank function and therefore also a polymatroid rank function. Scaling it with d_i preserves this property.

Now, the set P of all possible congestion vectors after all populations have allocated their demand consists of the sums of individual elements of all P_i, therefore

$$P = \{p_1 + \ldots + p_n \mid p_1 \in P_1, \ldots, p_n \in P_n\}.$$

The set P is also known as the *Minkowski sum* of all P_i, see for instance [22]. It is shown there that P can be written as

$$P = \left\{ y \in \mathbb{R}^E_+ \;\middle|\; y(X) \leq \sum_{i \in N} d_i \cdot p_i(X) \; \forall X \subseteq E, y(E) = \sum_{i \in N} d_i \cdot p_i(E) \right\}.$$

Since summing up polymatroid rank functions preserves submodularity, we get that P is a polymatroid base polytope as well.

Altogether, a Wardrop equilibrium may be computed as an optimal solution to

$$\min_{y \in P} \left(\sum_{e \in E} \int_0^{y_e} c_e(t)dt \right), \tag{3.3}$$

which is a convex separable minimization problem over a polymatroid base polytope that can be solved using the decomposition algorithm from section 2.5.

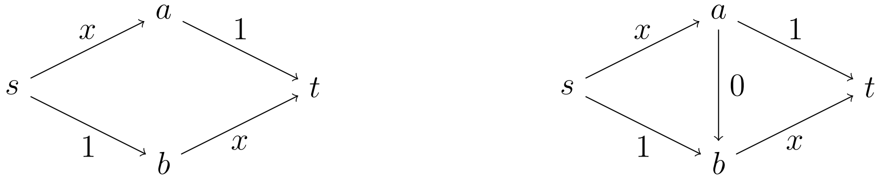
3.5 Sensitivity analysis: Matroid congestion games and Braess paradox

In this section, we will see a first application of *sensitivity analysis*. This is the study of how the optimal solution of an optimization problem changes when some parameters of the problem are adjusted. In other words, we assume that we have obtained an optimal solution for some optimization problem and want to see if we can change the solution slightly to accommodate different parameters.

For nonatomic matroid congestion games, we will see that decreasing game parameters (demands of the populations and cost functions of resources) results in a decrease of costs in a Wardrop equilibrium. This can be used to prove that a certain anomaly does not occur in matroid congestion games. The results in this section are from [10].

The anomaly is called "Braess paradox": For some congestion games, reducing demands or resource costs can negatively affect the cost experienced by some or all players in an equilibrium. We will illustrate both cases with examples.

Example 3.15 (Braess paradox: cost reductions)
This is a canonical example found e.g. in [18], [19] and [10]. Observe the network given in Figure 3.2a. We interpret it as a nonatomic network congestion game with a single population $1 \in N$ with demand $d_1 = 1$ whose set of strategies are the two paths from s to t. The cost of each edge is given as a function of the edge's congestion x. The game is in a Wardrop equilibrium when one half of the population chooses the upper path through the network and the other half chooses the lower path. In that case, the cost of each strategy is $\frac{1}{2} + 1 = \frac{3}{2}$ and since there are no others, all strategies in use by the population have minimum cost (cf. the definition of Wardrop equilibria 3.13).



(a) (b)

Figure 3.2: Braess paradox for cost reductions. The network in 3.2b features a low cost edge not present in 3.2a, but the cost on its equilibrium path is higher.

Now consider the network in Figure 3.2b. It is obtained from the network in Figure 3.2a by adding another edge with cost 0 *(equivalently, we could say that* *the edge is already included in Figure 3.2a and its cost was reduced from ∞ to 0). Intuitively, adding this edge should improve or at least not negatively affect network performance. However, the new Wardrop equilibrium routes all demand through the path $s \to a \to b \to t$, resulting in a total cost of $1 + 1 = 2$ on that path. In this case, either of the other two paths also impose a cost of 2, so this is indeed a Wardrop equilibrium.*

Thus, adding a new edge with low costs or reducing the cost of an existing edge in a network congestion game may increase the total cost for some players.

Example 3.16 (Braess paradox: demand reductions)
The previous example can be adapted to demonstrate that reducing the demand of some population in a nonatomic congestion game can increase the total cost for some players. Observe the network given in Figure 3.3. This time, the cost of the edge from a to b is given by the function

$$f(x) = \max\{x - 1, 0\}.$$

There are two populations $\{1, 2\} \in N$ with demands $d_1 = 1, d_2 = 2$. Population 1 may use either of the three paths from s to t. Population 2 only uses the path from a to b. In this scenario, the game is in a Wardrop equilibrium when one half of population 1 chooses the upper path through the network and the other half chooses the lower path. Population 2 imposes all of its demand on the edge from a to b, on the only path it has available. Then, as in the previous example, the cost of each strategy used by population 1 is $\frac{1}{2} + 1 = \frac{3}{2}$.

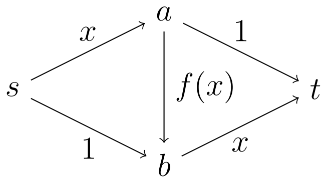


Figure 3.3: Braess paradox for demand reductions. Reducing the demand on the edge from a to b can negatively affect the total costs in an equilibrium.

Now assume the demand of population 2 is reduced to $\bar{d}_2 = d_2 - 2 = 0$. Afterwards, the network available to population 1 is equivalent to the one in Figure 3.2b. Thus, the cost of the strategy $s \to a \to b \to t$ used by population 1 in the new Wardrop equilibrium is 2. Altogether, reducing the demand of population 2 has increased costs for players in population 1.

The authors in [10] differentiate between two kinds of the paradox: the *strong Braess paradox* occurs when a cost or demand reduction increases the cost of strategies that are actually used in a Wardrop equilibrium. For the *weak Braess paradox*, it suffices when there is some resource with increased cost, but the congestion on that resource may be 0. Note that in examples 3.15 and 3.16 both the weak and the strong paradox occur. Definition 3.17 states them formally.

We begin by observing that whenever a population i has strategies $s, s' \in S_i$ available so that $s \subseteq s'$, then i will never be worse of by choosing s over s' since cost functions are non-negative. Therefore, from here on we assume that the strategy sets S_i of all populations i form a *clutter*, i.e, none of their strategies are contained in another.

Definition 3.17 (Braess paradox, Definition 2.6 in [10])
A family of strategy sets $(S_i)_{i\in N}$ admits Braess paradox if there is a nonatomic congestion game $\Gamma_{\text{NCG}} = (N, E, (S_i)_{i\in N}, (c_e)_{e\in E}, (d_i)_{i\in N})$ with these strategies and another instance obtained from Γ_{NCG} by reducing costs and/or demands, $\bar{\Gamma}_{\text{NCG}} = (N, E, (S_i)_{i\in N}, (\bar{c}_e)_{e\in E}, (\bar{d}_i)_{i\in N})$, with

$$\bar{c}_e(t) \leq c_e(t) \ \forall t \geq 0 \ \textit{and}$$

$$\bar{d}_i \leq d_i \ \forall i \in N$$

so that for Wardrop equilibria x of Γ_{NCG} and $\bar{x}$ of $\bar{\Gamma}_{\text{NCG}}$ it holds:

Weak Braess paradox: $\exists e \in E\colon c_e(n_e(x)) < \bar{c}_e(n_e(\bar{x}))$

Strong Braess paradox: $\exists i \in N, s, \bar{s} \in S_i\colon x^i_s > 0, \bar{x}^i_{\bar{s}} > 0$ *and*

$$\sum_{e\in s} c_e(n_e(x)) < \sum_{e\in \bar{s}} \bar{c}_e(n_e(\bar{x}))$$

Here, we will focus on the following result from [10]: if for all populations, the strategies S_i are the bases of matroids, then the weak Braess paradox cannot occur. This implies that the strong Braess paradox does not occur either, since the costs of strategies cannot rise when the costs of all resources do not increase.

To show this, two lemmata are needed. The first implies that minimal solutions of the Beckmann potential over a polymatroid decrease in all components under cost reductions. The second lemma states the same under demand reductions. From the previous section, we know that feasible strategy distributions of nonatomic matroid congestion games induce congestion vectors that form a polymatroid. Furthermore, minimum solutions of the Beckmann potential are Wardrop equilibria. Therefore, the lemmata imply that cost and demand reductions do not increase the cost for resources in Wardrop equilibria of these games. In other words: strategy sets that are bases of matroids are immune to the weak and strong Braess paradox.

Lemma 3.18 (Lemma 3.2 in [10])
Let $x, \bar{x} \in \mathbb{R}^E_+$ be two optimal solutions of equation 3.3, where the corresponding problem instances are obtained from nonatomic congestion games Γ_{NCG} and $\bar{\Gamma}_{\text{NCG}}$ that differ only in their cost functions such that $\bar{c}_e(t) \leq c_e(t)$ for all $t \geq 0$. Then it holds that

$$\bar{c}_e(\bar{x}_e) \leq c_e(x_e)$$

for all $e \in E$.

Proof
The key idea of this proof is to use Lemma 2.14 on a component of the minimal solution that is assumed, by contradiction, to have increased after the cost reduction.

Formally, assume $c_e(x_e) < \bar{c}_e(\bar{x}_e)$ for some $e \in E$. Then $\bar{x}_e > x_e$, because c_e is monotone. Since $x, \bar{x}$ are elements of the same base polytope, using the lemma gives a component $f \in E$ with $\bar{x}_f < x_f$ so that (e, f) is an exchangeable pair for x and (f, e) is an exchangeable pair for $\bar{x}$.

Now consider the optimality conditions of minimizing a convex separable function (such as the Beckmann potential) over a polymatroid from Lemma 2.16. The derivatives of the terms in the Beckmann potential are exactly the cost functions. Therefore, we obtain for the two exchangeable pairs that $c_f(x_f) \leq c_e(x_e)$ and $\bar{c}_e(\bar{x}_e) \leq \bar{c}_f(\bar{x}_f)$. This is sufficient to obtain the contradiction

$$c_f(x_f) \leq c_e(x_e) < \bar{c}_e(\bar{x}_e) \leq \bar{c}_f(\bar{x}_f) \leq c_f(x_f),$$

where the last inequality follows from our assumptions that cost functions are monotone and were only decreased. □

Lemma 3.19 (Lemma 3.3 in [10])
Let $x, \bar{x} \in \mathbb{R}_+^E$ be two optimal solutions of equation 3.3, where the corresponding problem instances are obtained from nonatomic congestion games Γ_{NCG} and $\bar{\Gamma}_{\text{NCG}}$ that differ only in the demand of one player such that $\bar{d}_j < d_j$ for a $j \in N$. Then it holds that

$$\bar{c}_e(\bar{x}_e) \leq c_e(x_e)$$

for all $e \in E$.

Proof (Sketch)
As with the previous lemma, the proof proceeds by contradiction. Here, however, the two minimal solutions are not elements of the same base polytope, so Lemma 2.14 is not directly applicable: if x is an element of a base polytope P as constructed in section 3.4, then $\bar{x}$ is an element of a similar polytope $\bar{P}$, where the summand P_j of the Minkowski sum is changed to

$$\bar{P}_j = \left\{ y^j \in \mathbb{R}_+^E \mid y^j(X) \leq \bar{d}_j \cdot p_j(X)\ \forall X \subseteq E, y^j(E) = \bar{d}_j \cdot p_j(E) \right\},$$

implying that $\bar{P}_j = \left\{ \frac{\bar{d}_j}{d_j} y^j \;\middle|\; y^j \in P_j \right\} = \frac{\bar{d}_j}{d_j} P_j$. The other summands in the Minkowski sum remain unchanged.

The key idea to circumvent this issue is to construct an element $x' \in \bar{P}$ so that, assuming $c_e(x_e) < \bar{c}_e(\bar{x}_e)$ for some $e \in E$, one gets $\bar{x}_e > x'_e$. Note that $x'_j = \frac{\bar{d}_j}{d_j} x_j \in \bar{P}_j$ and therefore $x' = x'_j + \sum_{i \in N \setminus \{j\}} x_i \in \bar{P}$. Then $\bar{x}_e > x_e \geq x'_e$. Thus, Lemma 2.14 can be applied for $x', \bar{x} \in \bar{P}$. The authors then obtain a contradiction similarly to the proof of the previous lemma. □

Lemma 3.19 refers to only one change in demand. If several populations reduce their demand, this can be interpreted as consecutive individual changes.

There are several more results in [10]. Importantly, it is shown that if $(S_i)_{i \in N}$ are immune to the weak Braess paradox, then all S_i must be bases of matroids. In the proof, a polytope per population i is constructed, containing all characteristic vectors of strategies in S_i. Then it is shown that the Minkowski sum of these polytopes must be a polymatroid base polytope arising from matroid bases. This is the converse direction of what was shown previously, thus yielding the characterization:

Theorem 3.20 (Theorem 3.1 in [10])
A family of strategy sets $(S_i)_{i \in N}$ is immune to the weak Braess paradox if and only if $(S_i)_{i \in N}$ are bases of matroids.

Finally, the authors show that when there are at least two populations, any $(S_i)_{i \in N}$ that are "universally immune" to the strong Braess paradox must consist of bases of matroids. Universally immune means the following: note that if the strategies are bases of matroids per population, this does not imply any information on how the strategy distribution of one population affects the resources used by other populations. For example, the S_i might be bases of matroids over disjunct ground sets, so that the populations are essentially independent from each other in their choices of resources. If $(S_i)_{i \in N}$ are universally immune to the strong Braess paradox, then the paradox cannot occur regardless of how the resources used by the populations correspond to each other.

To prove universal immunity, it is assumed that one population strategy set S_j does not contain the bases of a matroid. Then, the authors construct a game that invokes the strong Braess paradox: two populations (including j) are separated from the rest (meaning they operate on separate resource sets). Then certain properties of the non-matroid set system S_j are exploited which guarantee that the strong Braess paradox involving cost reductions can occur between these two populations.

These latter results do not involve sensitivity analysis, which is why we refer to [10] for the details.

3.6 Sensitivity analysis: Polymatroid games have pure Nash equilibria

In this section, we will see another application of sensitivity analysis. We will discuss the results from [12], where the following optimization problem is considered: a separable convex function that may depend on parameters and is subject to some regularity constraints regarding these parameters shall be minimized over an integral polymatroid (cf. definitions 2.11 and 2.12). We will see the regularity assumptions shortly. The authors in [12] show that minor changes to the parameters of the objective function or the polymatroid result in minor changes to the optimal solution. These results are used to prove that a particular class of non-cooperative games, called "polymatroid games", has pure Nash equilibria. Polymatroid games are quite similar to congestion games and it can be shown that atomic matroid congestion games are one of their special cases. Thus, the existence of pure Nash equilibria for polymatroid games generalizes Theorem 3.10.

Formally, the following optimization problem is analyzed:

Problem 3.21
*Let $p\colon 2^E \to \mathbb{N}$ be an integral polymatroid rank function and $P(p)$ the polymatroid polytope associated with p. For a $d \in \mathbb{N}$ with $d \leq p(E)$, the d-*truncated *integral polymatroid base polytope is defined as*

$$B(p,d) = \{x \in P(p) \mid x(E) = d\}\,.$$

Furthermore, let $c_e\colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ be cost functions for all $e \in E$ that are discrete convex *in their first parameter and* regular *(will be explained later).*

These cost functions depend on parameters $t_e \in \mathbb{N}$ for all $e \in E$. We want to find a solution to

$$\min_{x \in B(p,d)} \sum_{e \in E} c_e(x_e, t_e).$$

For fixed parameters t_e, this optimization problem bears resemblance to continuous convex separable optimization problems over polymatroids that were discussed earlier. It is, however, not immediately clear how convexity should be defined for functions on an integer domain. Consider a function $f\colon \mathbb{N} \to \mathbb{R}$. It's left and right derivatives are defined as

$$f^+(x) = f(x+1) - f(x) \text{ and } f^-(x) = f(x) - f(x-1)$$

for all $x \in \mathbb{N}$. Note that these are the difference quotients from continuous functions without taking the limit. Intuitively, the right derivative corresponds to the slope of a line drawn between the points $(x, f(x))$ and $(x+1, f(x+1))$ (similarly for the left derivative).

f is defined to be *discrete convex* if its left derivative is not greater than the right derivative, i.e.,

$$f^-(x) \leq f^+(x)$$

for all $x \in \mathbb{N}$. It can be shown (see [16]), that with this notion of discrete convexity, certain desirable properties known from continuous convex functions hold as well. In particular, any discrete convex function can be extended to a continuous convex function that coincides with it on the domain $\mathbb{N}$. Additionally, recall that a differentiable function is convex if its derivative is monotonically non-decreasing. For discrete functions, this holds similarly since $f^+(x) = f^-(x+1)$ and so $f^-(x) \leq f^-(x+1)$ and $f^+(x) \leq f^+(x+1)$ for discrete convex functions. Thus, the notion of discrete convexity also fulfills the intuitive idea: f grows quicker on larger values.

In fact, the optimality conditions for continuous convex separable optimization over polymatroids from section 2.5 hold for Problem 3.21 as well if the parameters are fixed. The left and right derivatives are with respect to the first parameter, since that parameter is being optimized over. So

$$c_e^+(x_e, t_e) = c_e(x_e+1, t_e) - c_e(x_e, t_e) \text{ and } c_e^-(x_e, t_e) = c_e(x_e, t_e) - c_e(x_e-1, t_e).$$

The decomposition algorithm can be applied to piecewise linear extensions of the c_e to get an integral solution for Problem 3.21. See [9] or [11] for proofs of these statements.

It remains to determine how the cost functions c_e may depend on their parameters t_e. If minimal changes to t_e can create arbitrary changes in c_e, then any sensitivity analysis will be fruitless. Therefore, the authors in [12] require the cost functions to be *regular*, meaning they fulfill additional constraints on their second parameter. Specifically, these constraints are

$$c_e^-(x_e, t_e) \leq c_e^-(x_e, t_e+1) \text{ and}$$

$$c_e^-(x_e, t_e+1) \leq c_e^+(x_e, t_e)$$

for all $x_e, t_e \in \mathbb{N}$. Intuitively, the second parameter of a regular cost function does not impact the cost too drastically, since the increase it causes to the marginal contribution of the first parameter is bounded by the right derivative. A regular function is also discrete convex in its first parameter, which can be seen by combining the two inequalities.

Two sensitivity results are given in [10]. The first refers to changes to the parameter t_e of cost functions c_e. The second considers changes to the polymatroid's parameter d. They state that minimal changes (meaning ±1) to one parameter result in elementary changes to optimal solutions of Problem 3.21 in at most two components. The results in full are:

Theorem 3.22 (Theorem 3.2 in [12])
Let x be a minimal solution for Problem 3.21 given parameters $(t_e)_{e\in E}$ and d. Consider a change in parameter $\bar{t}_e = t_e + 1$ for some $e \in E$.

Then either x is an optimal solution for the new parameters or there exists an exchangeable pair (g, e) associated with x such that $x + \chi_g - \chi_e$ is optimal for the new parameters.

Proof (Sketch)
Recall the optimality conditions from Lemma 2.16. If there is no exchangeable pair for e associated with x, then x is necessarily optimal for the new parameters. Otherwise, if for all exchangeable pairs (f, e) associated with x its holds that

$$c_e^-(x_e, t_e + 1) \leq c_f^+(x_f, t_f),$$

then intuitively, shifting some value from x_e to another component of x will not decrease costs under the $\bar{t}_e$. In this case, the authors use the regularity assumptions to show that x is still optimal for the new parameters.

If the above inequality does not hold, one chooses the exchangeable pair (g, e) associated with x where g has minimal marginal cost increase, so

$$c_g^+(x_g, t_g) \text{ minimal among exchangeble pairs } (g, e).$$

The authors proceed by showing that $x + \chi_g - \chi_e$ is optimal for the new parameters by proving that the optimality conditions from Lemma 2.16 are fulfilled. □

Theorem 3.23 (Theorem 3.4 in [12])
Let x be a minimal solution for Problem 3.21 given parameters $(t_e)_{e\in E}$ and d. Consider a change in parameter $\bar{d} = d + 1$, provided $\bar{d} \leq p(E)$.

Then there exists a $g \in E$ such that $x + \chi_g$ is optimal for the new parameters.

Proof (Sketch)
g is chosen such that

$$c_g^+(x_g, t_g) \text{ minimal among } \left\{ g \in E \;\middle|\; x + \chi_g \in B(p, \bar{d}) \right\}.$$

Note that such a g must always exist. Otherwise, we would have $\operatorname{sat}(x) = E$ and thus $x(E) = x(\operatorname{sat}(x)) = p(\operatorname{sat}(x)) = p(E)$ by the definition of saturated components. But since $x \in B(p, d)$, we have that $x(E) = d < \bar{d} \leq p(E)$. See also [9].

Again, the authors in [12] show that $x + \chi_g$ is optimal using the optimality conditions from Lemma 2.16. □

Both theorems also hold for the symmetric case that a parameter is decreased instead of increased. Combining them gives the following: when some parameters in Problem 3.21 change, an optimal solution x can be recomputed through at most two unit changes to x per unit change in the parameters. The authors in [12] use this idea to construct an algorithm that calculates pure Nash equilibria for a class of non-cooperative games related to polymatroid theory. We begin by introducing this class of games:

Definition 3.24 (Polymatroid games)
A polymatroid game is a tuple $\Gamma_{\mathrm{PG}} = (N, E, (p_i)_{i\in N}, (d_i)_{i\in N}, (c_{i,e})_{i\in N, e\in E})$ where N is a finite set of players, E a finite set of resources, $d_i \in \mathbb{N}$ are integral demands and $p_i \colon 2^E \to \mathbb{N}$ integral polymatroid rank functions per player. The strategy set of player i is $B(p_i, d_i)$, the d_i-truncated integral base polytope associated with p_i. Cost functions $c_{i,e} \colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ are specific to resources $e \in E$ and players $i \in N$. The first parameter shall contain the congestion that player i creates on resource e and the second parameter the congestion of all other players on e. The functions are required to be regular.

A profile contains vectors of all the players' polytopes, so $x = (x^1, \dots, x^n) \in B(p_1, d_1) \times \cdots \times B(p_n, d_n)$. Note that, since $x^i(E) = d_i$, all of player i's demand gets allocated among resources in E. We can consider x^i_e to be the congestion that player i creates on resource e and $\sum_{i\in N} x^i_e$ the total congestion created on e by all players. Therefore, the cost that resource e imposes on player i under profiles x is exactly

$$c_{i,e}\left(x^i_e, \sum_{j\in N\setminus\{i\}} x^j_e\right)$$

and the payoff for player i under profile x is $-\sum_{e\in E} c_{i,e}\left(x^i_e, \sum_{j\in N\setminus\{i\}} x^j_e\right)$.

Theorem 3.25
Polymatroid games have pure Nash equilibria that can be computed in time polynomial in the number of players, resources and the maximum d_i.

Theorem 3.25 follows from the correctness and runtime analysis of an algorithm proposed in [12] that constructs a pure Nash equilibrium. The algorithm initializes all players' demands to zero. Then all players' polytopes contain only the zero vector and a profile of zero vectors constitutes a pure Nash equilibrium for these demands. Afterwards, the demand of some player is increased by one and a Nash equilibrium for the new scenario is obtained through repeated best responses of all players. This is iterated until the original demands of all players are met.

In each iteration, players apply best responses until no player can improve anymore. Thus, once the original demands of the players have been reached, a Nash equilibrium of the polymatroid game is computed. Therefore, the algorithm is correct, provided that the sequence of best responses terminates. To show this, the two sensitivity results are essential: When the player whose demand is increased chooses a best response, they can do so by adding one unit to a component of their strategy. This follows from Theorem 3.23, since their strategy is a vector in a truncated polymatroid base polytope and they wish to minimize a sum of regular cost functions over that polytope.

Additionally, when the other players apply best responses after the initial increase of demand, they can do so by shifting a unit in their strategy vector from one component to another. This follows from Theorem 3.22, since the initial player's best response results in an increase of the congestion on some resource. Therefore, after each best response, there is exactly one resource with increased congestion compared to before the initial increase of demand. The key difficulty in proving the correctness of the algorithm is to show that this "shifting of units" does not loop endlessly. To that end, the authors show that a certain vector of marginal costs decreases after each best response (for a suitably defined order). As an analogy, this vector can be thought of as a potential associated with each profile of strategies.

In total, the algorithm iterates over every unit of demand of all players, the number of which can be bounded in the number of players and the maximum d_i over all $i \in N$. Likewise, the number of best responses in each iteration can be bounded polynomially in the number of resources, players, and player demands. Altogether, Theorem 3.25 follows. We refer to [12] for the rigorous proof.

As mentioned before, Theorem 3.25 generalizes Theorem 3.10. To see this, we need to prove that atomic matroid congestion games are polymatroid games. This is stated in Lemma 3.26.

Lemma 3.26 (Proposition 4.2 in [12])
Under the additional assumption that cost functions be non-decreasing, atomic matroid congestion games are polymatroid games.

Proof
The authors in [12] show this for atomic matroid congestion games where cost functions are both player and resource specific and the players' matroids can have different ground sets. Our definitions 3.6 and 3.9, in which cost functions are solely resource specific and all players' matroids share a ground set, are a special case.

Let $\Gamma_{\text{ACG}} = (N, E, (S_i)_{i \in N}, (c_e)_{e \in E})$ be an atomic matroid congestion game, where S_i is the bases of a matroid $(E, \mathcal{I}_i)$ for each player $i \in N$. We construct from this a polymatroid game $\Gamma_{\text{PG}} = (N, E, (p_i)_{i \in N}, (d_i)_{i \in N}, (c_{i,e})_{i \in N, e \in E})$, where p_i is simply the rank function of player i's matroid and the demand of each player is set to the rank of their matroid, so $d_i = p_i(E)$.

The cost function $c_{i,e}$ of player $i \in N$ and resource $e \in E$ in Γ_{PG} is obtained from the cost function c_e of resource e in Γ_{ACG} by

$$c_{i,e}(x_e^i, t_e) = \begin{cases} c_e(x_e^i + t_e) & \text{if } x_e^i = 1 \\ 0 & \text{if } x_e^i = 0 \end{cases}$$

for $x^i \in B(p_i, d_i)$ and all $t_e \in \mathbb{N}$. Intuitively, player i only experiences cost for resources they actually use and that cost is given by the original cost functions of the resources in the congestion game. Note that either $x_e^i = 1$ or $x_e^i = 0$, since x^i is a vector in the integral base polytope of player i, so $x^i \geq 0$, and for the matroid rank function p_i defining the polytope it holds: $p_i(\{e\}) \leq |\{e\}| = 1$ by (Subcardinality).

To finish the proof, it is sufficient to verify that all $c_{i,e}$ fulfill the regularity constraints. To that end, fix $i \in N, e \in E$ and a $t_e \in \mathbb{N}$. For $x_e^i = 0$, we do not need to check the constraints, as the left derivatives $c^-_{i,e}(0, t_e), c^-_{i,e}(0, t_e + 1)$ are not defined. For $x^i_e = 1$, the right derivative $c^+_{i,e}(1, t_e)$ is not defined, but we can check

$$\begin{gathered} c^-_{i,e}(x^i_e, t_e) = c^-_{i,e}(1, t_e) = c_{i,e}(1, t_e) - c_{i,e}(0, t_e) = c_e(1 + t_e) \leq \\ c_e(1 + t_e + 1) = c_{i,e}(1, t_e + 1) - c_{i,e}(0, t_e + 1) = c^-_{i,e}(1, t_e + 1) = c^-_{i,e}(x^i_e, t_e + 1), \end{gathered}$$

where the inequality holds because c_e is non-decreasing. □

Note the different approaches in showing the existence of pure Nash equilibria. In the proof of Theorem 3.10, this is done via a potential function. For polymatroid games, the key ideas were the sensitivity results from theorems 3.22 and 3.23.

The remaining results in [12] show that submodularity of the function defining the feasible region is necessary for the sensitivity results regarding Problem 3.21 to hold. Likewise, there is a game with regular convex cost functions and where the players' strategies do not form a polymatroid that does not have a pure Nash equilibrium. Importantly, these statements are only shown for the case of regular cost functions. For their proofs, we refer to [12].

Chapter 4

Cooperative Games

In chapter 3 we looked at games with selfish players who wish to maximize their payoff. In contrast, players in cooperative games may work together to improve their benefits or reduce their losses.

As in the previous chapter, we indicate the sources for all definitions and results at the beginning of each section. Section 4.4 consists of original work.

4.1 Core allocations

We now introduce some basic concepts from cooperative game theory that are found in standard textbooks. In particular, we will consider the problem of finding core allocations for cooperative games. The material in this section and the next can be found e.g. in [18] and [20].

Definition 4.1
A cooperative game *consists of a tuple* $\Gamma_{\text{coop}} = (N, c)$ *where* N *is a finite set of* players *participating in the game. We call subsets of players* $C \subseteq N$ coalitions. $c \colon 2^N \to \mathbb{R}_+$ *defines a* cost *for each coalition and we require* c *to satisfy*

(Normalization) $c(\emptyset) = 0$

(Monotony) $c(A) \leq c(B)$ *for* $A \subseteq B \subseteq N$

(Subadditivity) $c(A) + c(B) \geq c(A \cup B)$ *for* $A, B \subseteq N$ *with* $A \cap B = \emptyset$

Equivalently, cooperative games may be tuples (N, v) *where* $v \colon 2^N \to \mathbb{R}_+$ *is a* benefit *function satisfying normalization, monotony and*

(Superadditivity) $v(A) + v(B) \leq v(A \cup B)$ *for* $A, B \subseteq N$ *with* $A \cap B = \emptyset$

We will mostly focus on games defined using cost functions. Due to the subadditivity of c, players in a cooperative game are encouraged to work together (i.e., form coalitions) to minimize the cost they experience. Forming coalitions, however, opens up the question of how the cost $c(C)$ experienced by a coalition should be distributed to its members. Under an unfair distribution of costs, some players might prefer to defect from their coalition and form their own group to obtain lower costs.

A large part of cooperative game theory deals with the question of how the *grand coalition* N can distribute the cost $c(N)$ fairly among its members. Several solutions for this have been proposed, see for example [18]. We will deal with a particular solution concept called the *core*. It is motivated by the idea that no coalition wants to incur more cost than it contributes to the grand coalition.

Definition 4.2
The set of cost distributions $x \in \mathbb{R}^N$ with

(Budget balance) $\sum_{i \in N} x_i = c(N)$

(Core property) $\sum_{i \in C} x_i \leq c(C) \ \forall C \subseteq N$

is called the core *of the cooperative cost game (N, c). For benefit games (N, v), the core property becomes $\sum_{i \in C} x_i \geq v(C) \ \forall C \subseteq N$.*

Under a cost (benefit) distribution $x \in \mathbb{R}^N$, player i pays cost (gets benefit) x_i. The budget balance condition means all the cost gets distributed. The core property implies that no coalition pays higher cost (or gets less benefit) than it would pay (or get) if it defected from the grand coalition.

The core of a cooperative game may be empty, so competitive behavior may sometimes be preferable to cooperative options. This provides a motivation for other cost distribution concepts. In the next section, however, we will see a type of game where the core is guaranteed to be non-empty.

4.2 Convex cooperative games

It turns out that *convex cooperative games* have core elements that can be found efficiently.

Definition 4.3
A cooperative game is convex if its cost function is submodular or if its benefit function is supermodular.

Note that a submodular function is also subadditive and a supermodular function is also superadditive. To see that a core element for convex cost games can be found efficiently, observe: a submodular cost function is indeed the rank function of a polymatroid (cf. definitions 4.1 and 2.11). Therefore, the core is exactly the polymatroid base polytope (cf. definitions 4.2 and 2.12) and an element of that polytope can be found quickly, as described in Theorem 2.15.

For a convex benefit game (N, v), we can construct from the benefit function v a polymatroid rank function $v' \colon 2^N \to \mathbb{R}_+$ so that the associated polymatroid base polytope coincides with the core of the benefit game (see [9], Lemma 2.4). For all $C \subseteq N$ define

$$v'(N \setminus C) = v(N) - v(C), \text{ or equivalently: } v'(C) = v(N) - v(N \setminus C).$$

v' is clearly normalized and monotone and its submodularity results from the supermodularity of v, since for all $A, B \subseteq N$,

$$\begin{aligned} & v'(A) + v'(B) \geq v'(A \cup B) + v'(A \cap B) \\ \Leftrightarrow\ & 2v(N) - v(N \setminus A) - v(N \setminus B) \geq 2v(N) - v(N \setminus (A \cup B)) - v(N \setminus (A \cap B)) \\ \Leftrightarrow\ & v(N \setminus A) + v(N \setminus B) \leq v(N \setminus (A \cup B)) + v(N \setminus (A \cap B)) \\ & = v((N \setminus A) \cap (N \setminus B)) + v((N \setminus A) \cup (N \setminus B)). \end{aligned}$$

We can now see that the core of the benefit game and the base polytope for v' are equal since for a vector $x \in \mathbb{R}_+^N$ with $x(N) = v(N) = v'(N)$ we have

$$\begin{aligned} & x(C) \leq v'(C)\ \forall C \subseteq N \\ \Leftrightarrow\ & x(N \setminus C) \leq v'(N \setminus C) = v(N) - v(C)\ \forall C \subseteq N \\ \Leftrightarrow\ & x(C) = x(N) - x(N \setminus C) \geq v(C)\ \forall C \subseteq N. \end{aligned}$$

We will illustrate the concepts discussed so far with a cooperative game that has a supermodular benefit function and a real-world analogue. In this game, the players in N are *creditors* of a firm that has failed. Let d_i be the money in € owed by the bankrupt firm to creditor $i \in N$. Furthermore, let E be the firm's remaining *estate*, i.e., the monetary equivalent in € of all assets of the firm that are to be divided among its creditors. We assume that the combined claims of all creditors exceed the estate. We wish to find a fair allocation of E among the players, in other words, we want to find a core allocation for the following cooperative game:

Definition 4.4 (Bankruptcy game, see [4])
Let $N = \{1, \dots, n\}, E \in \mathbb{N}_+$ and $d = (d_1, \dots, d_n)$ with $d_i \in \mathbb{N}_+$ so that $\sum_{i \in N} d_i > E$. Furthermore, define a function $v \colon 2^N \to \mathbb{R}_+$ by

$$v(C) = \max \left\{ \left(E - \sum_{i \in N \setminus C} d_i \right), 0 \right\}$$

for all $C \subseteq N$. Then the cooperative benefit game $\Gamma_{\text{bk}} = (N, v)$ is called a bankruptcy game.

$v(C)$ is exactly the amount of money that a coalition $C \subseteq N$ of creditors gets without going to court since it is the amount left after all other creditors $N \setminus C$ have had their claim fulfilled. Importantly, v is supermodular (see Lemma 4.6). We illustrate bankruptcy games and the computation of a core vector using Theorem 2.15 in the following example:

Example 4.5
A firm has gone bankrupt and leaves an estate of $E = 1\,000\,000$ €. There are four creditors $N = \{1, 2, 3, 4\}$ with claims $d_1 = 300\,000$ €, $d_2 = 200\,000$ €, $d_3 = 700\,000$ € and $d_4 = 200\,000$ €. For the benefit function we get e.g. $v(\{2, 3\}) = 500\,000$ €.

To compute a core allocation, we fix the permutation $\pi = (1, 2, 3, 4)$ of N and get

$$\begin{aligned} x_1 &= v(\{1\}) - v(\emptyset) = 0 €, \\ x_2 &= v(\{1,2\}) - v(\{1\}) = 100\,000 €, \\ x_3 &= v(\{1,2,3\}) - v(\{1,2\}) = 700\,000 €, \\ x_4 &= v(\{1,2,3,4\}) - v(\{1,2,3\}) = 200\,000 €. \end{aligned}$$

The allocation (x_1, x_2, x_3, x_4) of E among the creditors thus obtained is in the core of the cooperative game. Therefore, no sub-coalition of creditors would be able to get a better allocation for themselves if they chose to defect from the grand coalition.

Lemma 4.6
The bankruptcy game is convex, i.e., its benefit function is supermodular.

Proof
Let $S \subseteq L \subseteq N$ and $n \in N \setminus L$. For the valuation function of the bankruptcy game we need to show

$$v(S \cup \{n\}) - v(S) \leq v(L \cup \{n\}) - v(L).$$

Firstly, observe that $v(X) \geq 0$ for all $X \subseteq N$ and that $v(A) \leq v(B)$ for all $A \subseteq B \subseteq N$, because $\sum_{i \in N \setminus A} d_i \geq \sum_{i \in N \setminus B} d_i$. Now, if $v(L) = 0$ then also $v(S) = 0$, because of these observations. But then also $v(S \cup \{n\}) - v(S) \leq v(L \cup \{n\}) - v(L) \Leftrightarrow v(S \cup \{n\}) \leq v(L \cup \{n\})$ holds.

Thus, assume $v(L) \neq 0$, therefore $v(L) = E - \sum_{i \in N \setminus L} d_i$ and

$$v(L \cup \{n\}) - v(L) = E - \sum_{i \in N \setminus (L \cup \{n\})} d_i - E + \sum_{i \in N \setminus L} d_i = d_n.$$

If both $v(L) \neq 0$ and $v(S) \neq 0$, we get $v(S \cup \{n\}) - v(S) = d_n = v(L \cup \{n\}) - v(L)$.

It remains the case $v(L) \neq 0$ and $v(S) = 0$. But then $\sum_{i \in N \setminus S} d_i \geq E$ and therefore $v(S \cup \{n\}) \leq d_n = v(L \cup \{n\}) - v(L)$ so we obtain the claim. □

4.3 The joint replenishment game

We now introduce another, more complex cooperative game that is also convex. This section is based on [13], [23] and [7].

Consider the following scenario: a group of *retailers* $N = \{1, \dots, n\}$ sell the same product, which they obtain from the same supplier. They are interested in ordering their units (instances of the product) together, so as to benefit from volume discount offered by the supplier. However, they have different levels of demand for the product at their stores and also varying costs associated with holding the units before they are sold. Therefore, they need to solve two problems in order to cooperate: firstly, they will want to know at which intervals each retailer requests their units (to maximize joint orders) and secondly, how to fairly distribute costs, preventing retailers from forming competing coalitions.

This scenario, and variations, has been extensively studied as the “joint replenishment problem” (a comprehensive survey is given in [14]). Here, we make some additional assumptions (in accordance with [13] and [23]): We assume that time progresses infinitely in steps $t \in \mathbb{N}_+$ and that each retailer $i \in N$ faces a constant *demand* $d_i \in \mathbb{R}_{>0}$ for the product at these time steps. Furthermore, we assume that when an order is placed to the supplier, it instantaneously arrives at the retailers. Finally, we restrict the retailers to fixed-interval reorders, meaning: each retailer i has an associated *replenishment interval* T_i, so that they order at times $0, T_i, 2T_i, 3T_i, 4T_i$ etc.

Now, denote by $h_i \in \mathbb{R}_{>0}$ the *holding cost* for a unit at retailer i and fix a function $p\colon 2^N \to \mathbb{R}_+$ that gives for each group of retailers $C \subseteq N$ a cost $p(C)$ they incur when ordering together (irrespective of the amount of units being ordered). We will call this the *setup cost.* The function p shall satisfy the properties in Definition 2.11, so p is a polymatroid rank function. Due to submodularity, this means that retailers have an incentive to order together.

Now we turn towards answering the first question: at which intervals should retailers place their orders? It can be shown (see [7]), that among fixed-interval reorder strategies the "power-of-two policies" are never bad: The cost of choosing a power-of-two policy is at most 6 % higher than choosing the optimal among all fixed-interval strategies. By choosing the optimal power-of-two policy, this decreases to 2 %. Formally, a power-of-two policy is defined by a retailer independent *base planning period* $\mathscr{L} \in \mathbb{R}_{>0}$ and retailer specific parameters $m_i \in \mathbb{Z}$, so that the replenishment intervals per $i \in N$ become

$$T_i = 2^{m_i}\mathscr{L}.$$

For the remainder, we fix an $\mathscr{L}$. We still need to find appropriate m_i. Furthermore, we need to answer question two: how should the cost be allocated among players? For this, we construct a cooperative game. Its characteristic function $c(C)$ will describe the long-run average cost of retailers $C \subseteq N$ ordering together using an optimal power-of-two policy. Define

$$\mathcal{T}^{\mathscr{L}} = \{2^m\mathscr{L} \mid m \in \mathbb{Z}\},$$

the set of all power-of-two replenishment intervals and

$$\mathcal{T}^{\mathscr{L}}_C = \left\{(T_i)_{i\in C} \;\middle|\; T_i \in \mathcal{T}^{\mathscr{L}} \; \forall i \in C\right\}$$

as a shortcut for the set containing all combinations of power-of-two replenishment intervals for retailers in C. Then the average cost in the long run is

$$c(C) = \min_{T_C \in \mathcal{T}^{\mathscr{L}}_C} g_C(T_C) + h_C(T_C),$$

where $g_C(T_C)$ is the *average setup cost* and $h_C(T_C)$ the *average holding cost* per time step for C given replenishment intervals T_C. For the latter, consider a player $i \in N$. Observe that in the first time step of the replenishment interval T_i, the cost for holding player i's entire inventory is $h_i d_i T_i$. In the second step it is $h_i d_i (T_i - 1)$, then $h_i d_i (T_i - 2)$ etc. Thus, the total holding cost over the entire interval T_i is $\frac{1}{2} h_i d_i T_i (T_i + 1)$, and for the average cost per time step we obtain

$$\frac{h_i d_i T_i (T_i + 1)}{2T_i} = \frac{1}{2} h_i d_i T_i + \frac{1}{2}.$$

In accordance with [13] and [23], we omit the constant term, use the shortcuts $H_i = \frac{1}{2} h_i d_i$ for all $i \in N$ and get

$$h_C(T_C) = \sum_{i \in C} H_i T_i.$$

The average setup cost g_C is the sum of all possible setup costs multiplied by the frequency with which the corresponding order occurs. Formally, fix some combination of replenishment intervals $T_C = (T_i)_{i\in C} \in \mathcal{T}_C^{\mathscr{L}}$ and a permutation $\pi = (\pi_1, \dots, \pi_{|C|})$ of C that orders the intervals non-decreasingly:

$$T_{\pi_1} \leq \cdots \leq T_{\pi_{|C|}}$$

Now, since for every $1 \leq i \leq |C|$ there is an m with $T_{\pi_i} = 2^m \mathscr{L}$ we have for all $1 \leq j < i$ an $y < m$ with $T_{\pi_j} = 2^y \mathscr{L}$. So retailer π_j orders 2^{m-y} times as often as π_i and when π_i orders the pth time, π_j orders the $(2^{m-y} \cdot p)$th time. Importantly, when π_i orders, all π_j with $1 \leq j < i$ order as well.

The frequency of retailers $\{\pi_1, \dots, \pi_i\}$ ordering is $\frac{1}{T_{\pi_i}}$. Consequently, the frequency of $\{\pi_1, \dots, \pi_i\}$ ordering when π_{i+1} does not order is $\frac{1}{T_{\pi_i}} - \frac{1}{T_{\pi_{i+1}}}$ (with the convention that $\frac{1}{T_{\pi_{|C|+1}}} = 0$). Now we can write down the average setup cost, which is

$$\begin{aligned}
g_C(T_C) &= \sum_{i=1}^{|C|} p\left(\{\pi_1, \dots, \pi_i\}\right) \left(\frac{1}{T_{\pi_i}} - \frac{1}{T_{\pi_{i+1}}} \right) \\
&= \sum_{i=1}^{|C|-1} \left(\frac{1}{T_{\pi_i}} p\left(\{\pi_1, \dots, \pi_i\}\right) - \frac{1}{T_{\pi_{i+1}}} p\left(\{\pi_1, \dots, \pi_i\}\right) \right) + \\
&\qquad \frac{1}{T_{\pi_{|C|}}} p\left(\left\{\pi_1, \dots, \pi_{|C|}\right\}\right) \\
&= \sum_{i=1}^{|C|-1} \frac{1}{T_{\pi_i}} \Big(p\left(\{\pi_1, \dots, \pi_i\}\right) - p\left(\{\pi_1, \dots, \pi_{i-1}\}\right) \Big) + \\
&\qquad \frac{1}{T_{\pi_{|C|}}} p\left(\left\{\pi_1, \dots, \pi_{|C|}\right\}\right) + \frac{1}{T_{\pi_{|C|}}} p\left(\left\{\pi_1, \dots, \pi_{|C|-1}\right\}\right) \\
&= \sum_{i=1}^{|C|} \frac{1}{T_{\pi_i}} \Big(p\left(\{\pi_1, \dots, \pi_i\}\right) - p\left(\{\pi_1, \dots, \pi_{i-1}\}\right) \Big)
\end{aligned}$$

Comparing the formula for $g_C(T_C)$ with our discussion of linear optimization over polymatroids from section 2.4, we can see similarities: Consider a linear maximization task where the weight per element $i \in C$ is $\frac{1}{T_i}$. Writing $x_i = p(\{\pi_1, \dots, \pi_i\}) - p(\{\pi_1, \dots, \pi_{i-1}\})$, we can see that $g_C(T_C)$ is the optimal value of maximizing the function $f\colon \mathbb{R}_+^{|C|} \to \mathbb{R}, f(x) = \sum_{i=1}^{|C|} \frac{1}{T_i} x_i$ over the polymatroid base polytope of p restricted to C, which is

$$B(p)|_C = \left\{ x \in \mathbb{R}_+^{|C|} \;\middle|\; x(X) \leq p(X) \; \forall X \subseteq C, x(C) = p(C) \right\}.$$

But note that if $x \in B(p)|_C$, then $x' \in \mathbb{R}_+^{|N|}$ with

$$x'_n = \begin{cases} x_n & \text{if } n \in C \\ 0 & \text{otherwise} \end{cases}$$

for all $n \in N$ satisfies $x' \in P(p)$. Furthermore, if x maximizes f over $B(p)|_C$, then x' maximizes $f'\colon \mathbb{R}_+^{|N|} \to \mathbb{R}, f(x) = \sum_{i=1}^{|C|} \frac{1}{T_i} x_i$ over $P(p)$, since any component not in $\operatorname{sat}(x')$ does not affect the function value of f'. So we can rewrite $g_C(T_C)$ as

$$g_C(T_C) = \max_{x \in P(p)} \sum_{i \in C} \left(\frac{x_i}{T_i} \right).$$

We summarize the discussion so far in the following definition:

Definition 4.7 (Joint replenishment game)
A joint replenishment game (with power-of-two reorder policies) consists of a tuple $\Gamma_{\text{JRG}} = (N, (d_i)_{i \in N}, (h_i)_{i \in N}, p\colon 2^N \to \mathbb{R}_+, \mathscr{L})$ *with retailers (players)* $N = \{1, \dots, n\}$*, demands* $d_i \in \mathbb{R}_{>0}$ *and holding costs* $h_i \in \mathbb{R}_{>0}$ *per player, setup costs given by a polymatroid rank function* p *and a base planning period* $\mathscr{L}$*.*

The cost function of the game is given by

$$c(C) = \min_{(T_i)_{i \in C} \in \mathcal{T}_C^{\mathscr{L}}} \max_{x \in P(p)} \sum_{i \in C} \left(\frac{x_i}{T_i} + H_i T_i \right)$$

for coalitions $C \subseteq N$*.*

Observe that calculating the cost for a coalition involves optimizing over a polymatroid. This will be the key insight to prove that the joint replenishment game is, like the bankruptcy game, convex. Thus, efficiently calculating a core allocation of the joint replenishment game is possible using Theorem 2.15, provided the individual costs $c(C)$ can be computed efficiently. We will see that this is also the case.

To prove convexity of the joint replenishment game, we need another theorem first: Theorem 4.8 implies that the objective function (i.e., the objective value as a function of the set to optimize for) of maximizing a concave separable function over a polymatroid is submodular. Thus, if the valuation function of a cooperative game is given, for each coalition, by a concave separable optimization problem over a polymatroid, then the game is convex.

Theorem 4.8 (Theorem 3 in [13])
Let N *be a finite set,* $p\colon 2^N \to \mathbb{R}_+$ *a polymatroid rank function and* $P(p)$ *the associated polymatroid polytope over* N*. Also, let* $f_i\colon \mathbb{R} \to \mathbb{R}$ *be concave functions for* $1 \leq i \leq |N|$*. Now, define for each* $C \subseteq N$ *the optimization problem*

$$h(C) = \max_{x \in P(p)} \sum_{i \in C} f_i(x_i).$$

Then the function of optimal values $h\colon 2^N \to \mathbb{R}$ *is submodular.*

Proof (Sketch)
The authors in [13] first prove an analogous result for linear maximization over a polymatroid. They use that to prove the property from equation 2.1 for h and any two sets $A, B \subseteq N$. Let $x^{A \cup B}$, $x^{A \cap B}$ be optimal solutions of the optimization problems $h(A \cup B)$ and $h(A \cap B)$. Now construct a family of linear functions $\hat{f}_i$ that coincide with f_i at $x_i^{A \cup B}$ and $x_i^{A \cap B}$. Note that the concavity of f_i implies that f_i lies above $\hat{f}_i$ in the range between $x_i^{A \cup B}$ and $x_i^{A \cap B}$.

Denote by $\hat{h}\colon 2^N \to \mathbb{R}$ the function of optimal values of the optimization problems where all f_i are replaced by their counterparts $\hat{f}_i$. Also, denote by $\hat{\hbar}$ and $\hbar$ similar functions, where only elements in the interval from $x^{A \cup B}$ to $x^{A \cap B}$ can be feasible solutions. By the aforementioned analogous result, $\hat{h}$ and $\hat{\hbar}$ are submodular. The claim follows, because

$$h(A) + h(B) \geq \hbar(A) + \hbar(B) \geq \hat{\hbar}(A) + \hat{\hbar}(B) \geq \\ \hat{\hbar}(A \cup B) + \hat{\hbar}(A \cap B) \geq h(A \cup B) + h(A \cap B),$$

where the last inequality follows because the solutions $x^{A \cup B}$ and $x^{A \cap B}$ are in that interval and there, $\hat{h}$ coincides with h. See [13] for the rigorous argument. □

Theorem 4.9 (Theorem 5 in [13])
The joint replenishment game is convex, i.e., its cost function is submodular.

Proof
It is shown in [23] and [7], that the value of $c(C)$ remains the same when changing the optimization order, so

$$c(C) = \max_{x \in P(p)} \sum_{i \in C} \min_{T_i \in \mathcal{T}^{\mathscr{L}}} \left(\frac{x_i}{T_i} + H_i T_i \right).$$

By Theorem 4.8, we need to show that $\min_{T_i \in \mathcal{T}^{\mathscr{L}}} \left(\frac{x_i}{T_i} + H_i T_i \right)$ is concave in x_i. But this is clear, since $\frac{x_i}{T_i} + H_i T_i$ is linear in x_i for each T_i and minimizing over all possible T_i gives a concave function. Thus, we are done. □

From the proof of Theorem 4.9 we get that computing $c(C)$ is equivalent to maximizing a separable concave function over a polymatroid. From section 2.5 we know that this is efficiently possible using the decomposition algorithm in [11]. Therefore, we can efficiently compute a core allocation of the joint replenishment game using Theorem 2.15.

4.4 Sensitivity analysis: Recomputing core allocations

We now turn towards the following question: given a core allocation for a cooperative game, can we quickly adjust the allocation after changes to the game parameters occurred so that the resulting vector is still in the core? For convex games, we know that a core allocation can be computed in linear time. Therefore, we would like to know whether there are special cases where recomputing the core allocation requires only constantly many steps.

Example 4.10
Recall Example 4.5. There, we had a bankruptcy scenario with parameters $E = 1\,000\,000$ €, $d_1 = 300\,000$ €, $d_2 = 200\,000$ €, $d_3 = 700\,000$ € *and* $d_4 = 200\,000$ €. *We calculated a core allocation* x *with* $x_1 = 0$ €, $x_2 = 100\,000$ €, $x_3 = 700\,000$ € *and* $x_4 = 200\,000$ €.

Now imagine the estate of the bankrupt firm turned out to be larger than originally thought, so $\bar{E} = E + 100\,000$ €. *This increases the benefit for coalitions*

$\{1,2\},\{1,2,3\},\{1,2,3,4\}$ *but not for* $\{1\}$. *Therefore, a core allocation* $\bar{x}$ *for the modified scenario can be obtained from the previous allocation by setting* $\bar{x}_2 = x_2 + 100\,000$ € *and* $\bar{x}_1 = x_1, \bar{x}_3 = x_3, \bar{x}_4 = x_4$.

If, on the other hand, the claim of some creditor happened to be overstated, e.g. $\bar{d}_3 = d_3 - 100\,000$ €, *this may increase the benefit for coalitions that creditor 3 is not a member of, namely* $\{1\},\{1,2\}$. *Here, only* $v(\{1,2\})$ *increases, so we can recompute the core allocation with* $\bar{x}_2 = x_2 + 100\,000$ €, $\bar{x}_3 = x_3 - 100\,000$ € *and* $\bar{x}_1 = x_1, \bar{x}_4 = x_4$.

Generalizing from this example, we want to find cost or benefit functions of convex games in some parameters such that changes to a parameter result in equal changes to the function values of consecutive coalitions. We formalize these requirements in the following definition:

Definition 4.11 (Consecutive changes)
Let N *be a finite set of size* $|N| = n$ *and for a permutation* $\pi = (1, \ldots, n)$ *of* N *write* $C_i^\pi = \{1, \ldots, i\}$ *for all* $1 \le i \le n$ *and* $C_0^\pi = \emptyset$.

We say that a normalized, monotone and sub- or supermodular function $f_d \colon 2^N \to \mathbb{R}_+$ *in parameters* $d = (d_1, \ldots, d_l) \in \mathbb{R}^l$ *allows for* consecutive changes *if for some permutation* π *of* N *and all parameters* d *the following conditions hold: if* $\bar{d} = d \pm \chi_j$ *for some* $1 \le j \le l$, *then either* $f_{\bar{d}} = f_d$ *or there exist* $\alpha \in \mathbb{R}_+$ *and indices* $1 \le p_j \le q_j \le n$ *so that*

$$f_{\bar{d}}(C_i^\pi) = \begin{cases} f_d(C_i^\pi) & \textit{if } i < p_j \textit{ or } q_j < i \\ \textit{1.) } f_d(C_i^\pi) + \alpha \textit{ or 2.) } f_d(C_i^\pi) - \alpha & \textit{if } p_j \le i \le q_j \end{cases}$$

for all $1 \le i \le n$ *and additionally*

$$f_d(C_{q_j+1}^\pi) - f_d(C_{q_j}^\pi) \ge \alpha \textit{ if } q_j < n \textit{ in case 1.)}$$
$$f_d(C_{p_j}^\pi) - f_d(C_{p_j-1}^\pi) \ge \alpha \textit{ in case 2.)}$$

The first condition requires that changing parameter j affects the consecutive coalitions $C_{p_j}^\pi, C_{p_j+1}^\pi, \ldots, C_{q_j}^\pi$ equally. The additional conditions ensure monotony after the parameter change. For example, if $f_d(C_{p_j}^\pi) - f_d(C_{p_j-1}^\pi) < \alpha$ in case 2, we would have $f_{\bar{d}}(C_{p_j}^\pi) < f_{\bar{d}}(C_{p_j-1}^\pi)$, so $f_{\bar{d}}$ would not be monotone.

Now assume a family of functions f_d satisfies the conditions in Definition 4.11 and a core allocation x of the associated convex game was calculated for parameters d and a permutation π according to the formula in Theorem 2.15. After a change of parameters $\bar{d} = d \pm \chi_j$ for some $1 \le j \le l$, a core allocation $\bar{x}$ can be calculated as

$$\bar{x}_i = x_i + \begin{cases} \alpha & \text{if } i = p_j \text{ in case 1.) or } i = q_j + 1 \text{ in case 2.)} \\ -\alpha & \text{if } i = p_j \text{ in case 2.) or } i = q_j + 1 \text{ in case 1.)} \\ 0 & \text{otherwise} \end{cases}$$

for all $1 \le i \le n$. Note that computing the core allocation using Theorem 2.15 requires a number of operations linear in $|N|$. In contrast, recomputing $\bar{x}$ using the formula above requires at most two changes to the components of x.

Example 4.12
Consider a cooperative cost game where every player $i \in N$ is assigned a parameter $d_i \in \mathbb{N}_+$ and the cost of a coalition is the maximum d_i among members of the coalition (possibly scaled by a constant). Thus, for a constant $\alpha \in \mathbb{R}_+$,

$$c_d(C) = \alpha \max_{i \in C}\{d_i\}$$

for all $C \subseteq N$. Note that c_d is normalized, monotone and submodular. Also, Definition 4.11 applies to this family of cost functions: Consider an increase of player j's parameter, $\bar{d}_j = d_j + 1$. Then the cost for coalitions where d_j was maximal is increased by α. Note that when $\bar{d}_j$ is increased again, the costs may change for other coalitions. In terms of the definition above, the indices p_j, q_j may be different on repeated increases.

Similarly, when player j's parameter decreases, $\bar{d}_j = d_j - 1$, the cost of coalitions where d_j was maximal and $\bar{d}_j$ is still maximal is reduced by α.

Example 4.13
Definition 4.11 applies to bankruptcy games, too. Assume $\bar{d}_j = d_j + 1$ for a player $j \in N$ (again, the case $\bar{d}_j = d_j - 1$ is similar). Then p_j is the lowest index in the permutation π such that $j \notin C^\pi_{p_j}$ and $E - \sum_{i \in N \setminus C^\pi_{p_j}} d_i \geq 1$. Furthermore, q_j is the highest index such that $j \notin C^\pi_{q_j}$. These indices need not exist (for example, if j is the first entry in π). Then the cost for all coalitions remains the same.

Now assume there is a change to the estate $\bar{E} = E + 1$ (once more, $\bar{E} = E - 1$ follows similarly). Then $q_j = n$ and p_j is the lowest index in π such that $\sum_{i \in N \setminus C^\pi_{p_j}} d_i \leq E$.

We can generalize the benefit function of the bankruptcy game to obtain more functions to which Definition 4.11 applies. Consider the following classes of functions in player-specific parameters $d = (d_1, \dots, d_n) \in \mathbb{N}^n_+$:

$$g_d \colon 2^N \to \mathbb{R}_+, g_d(C) = \sum_{i \in C} f_i(d_i) \text{ for component functions } f_i \colon \mathbb{N} \to \mathbb{R}_+$$

$$h_d \colon 2^N \to \mathbb{N}, h_d(C) = \min\{\sum_{i \in C} d_i, \gamma\} \text{ for } \gamma \in \mathbb{N}$$

g_d and h_d are clearly normalized and monotone. The functions g_d are in fact modular, cf. section 2.1, which implies both sub- and supermodularity: changes to d_j affect all coalitions with $j \in C$ equally. For h_d, submodularity can be shown similarly to Lemma 4.6. Intuitively, h_d are modular up to a “cut-off” at γ. Because of this cut-off, adding an element to a smaller coalition always increases the function value at least as much as adding it to a larger coalition. Thus, submodularity is also guaranteed for h_d.

Definition 4.11 also applies to g_d and h_d: Fix a permutation π of N, compute a core allocation for parameters d and then consider $\bar{d}_j = d_j + 1$. For g_d, $q_j = n$ and p_j is the lowest index such that $j \in C^\pi_{p_j}$. For h_d, p_j is defined the same way, unless $\sum_{i \in C^\pi_{p_j}} d_i \geq \gamma$, in which case the allocation does not change at all. The index q_j is also defined as before, unless $\sum_{i \in N} d_i \geq \gamma$, in which case q_j is the highest index such that $\sum_{i \in C^\pi_{q_j}} d_i < \gamma$.

So far, all presented functions fulfilling Definition 4.11 depended on one parameter per player. We will now introduce a game where the relationship between parameters and players is a bit more intricate. Efficiently recomputing core vectors will also be possible. The game is inspired by weighted coverage functions, which are mentioned in [15].

The game can be described intuitively as follows: A group of people want to buy a customizable product, e.g. an expensive machine. They do not need more than one machine in total, but all have different customization requirements. In other words, there is a finite set of machine features F that may be chosen as customization options, each of which has an associated price $w_f \in \mathbb{R}_+$ for $f \in F$. Player $i \in N$ requires a set of features $F_i \subseteq F$, which they expect the purchased machine to have. When a group of players orders together, they pay for each requested feature only once. After the purchase, the total cost of the order shall be allocated fairly among players. Such an allocation is in the core of the following cooperative game:

Definition 4.14 (Coverage game)
Let N be a finite set of $|N| = n$ players and F a finite set of $|F| = m$ features and $F_1, \dots, F_n \subseteq F$ subsets for each player. Furthermore, let $w = (w_1, \dots, w_m) \in \mathbb{R}_+^F$ be prices per feature and define a function $c\colon 2^N \to \mathbb{R}_+$,

$$c(C) = \sum_{f \in \cup_{i \in C} F_i} w_f$$

for all $C \subseteq N$. Then the cooperative cost game $\Gamma_{\mathrm{cv}} = (N, c)$ is called a coverage game.

Lemma 4.15
The coverage game is convex, i.e., its cost function is submodular.

Proof
Let $S \subseteq L \subseteq N$ and $i \in N \setminus L$. For the cost function of the coverage game we need to show

$$c(S \cup \{i\}) - c(S) \geq c(L \cup \{i\}) - c(L).$$

Notice that for any $C \subseteq N$, it holds that $c(C \cup \{i\}) - c(C)$ is the sum of weights of all $f \in F_i$ that are not also elements of some F_j for $j \in C$, thus $c(C \cup \{i\}) - c(C) = \sum_{f \in F_i \setminus (\cup_{j \in C} F_j)} w_f$. Furthermore, note that since $S \subseteq L$ also $\cup_{j \in S} F_j \subseteq \cup_{j \in L} F_j$. Altogether we get

$$c(S \cup \{i\}) - c(S) = \sum_{f \in F_i \setminus (\cup_{j \in S} F_j)} w_f \geq \sum_{f \in F_i \setminus (\cup_{j \in L} F_j)} w_f = c(L \cup \{i\}) - c(L). \qquad \square$$

We can identify the feature sets per player with their corresponding characteristic vectors and write them as rows of a matrix. That way, we obtain a matrix $M \in \{0,1\}^{N \times F}$ where $M_{if} = 1 \Leftrightarrow f \in F_i$. Thus, the parameters of a coverage game are the price vector w and the matrix M.

We can now see that the cost function of the game fulfills Definition 4.11: changes to a price w_f affect the first player in a permutation that has feature f in their feature set. Formally, fix parameters w, M and a permutation $\pi = (1, \dots, n)$ of N

and calculate a core allocation x for that scenario using Theorem 2.15. Now consider a change in parameters $\bar{w} = w + \alpha\chi_f$ for some $f \in F, \alpha > 0$ (the case of price reductions is done similarly). Then a new core allocation $\bar{x}$ is obtained by

$$\bar{x}_i = \begin{cases} x_i + \alpha & \text{if } i = \min\{j \mid f \in F_j\} \\ x_i & \text{otherwise} \end{cases}$$

for all $1 \leq i \leq n$.

Changes to a feature set of a player k can be $\bar{F}_k = F_k \cup \{f\}$ if $f \notin F_k$ or $\bar{F}_k = F_k \setminus \{f\}$ if $f \in F_k$ for some $f \in F$. In terms of the aforementioned matrix, this corresponds to $\bar{M}_{kf} = M_{kf} \pm 1$. We focus on the case $M_{kf} = 1$ (and so $\bar{M}_{kf} = M_{kf} - 1 = 0$) and note that the opposite case can be dealt with similarly. This change affects player k if they are the first in a permutation to have f in their feature set and - provided k is affected - another player who is second to have f in their feature set. Formally, for all $1 \leq i \leq n$,

$$\bar{x}_i = \begin{cases} x_i - w_f & \text{if } i = k \text{ and } k = \min\{j \mid f \in F_j\} \\ x_i + w_f & \text{if } i > k \text{ and } k = \min\{j \mid f \in F_j\} \text{ and } i = \min\{j > k \mid f \in F_j\} \\ x_i & \text{otherwise} \end{cases}$$

The cost function of the coverage game is somewhat different from the functions we saw before: it can not be separated by parameters per player, because a parameter $M_{if} = 1$ or $M_{if} = 0$ may or may not influence the cost for a coalition depending on whether there is another player in the coalition that has f in their feature set. The modular functions g_d and the coverage game cost functions are similar, however, in that they contain coalition dependent sums over parameters. This observation may be useful to prove a characterization of all functions that Definition 4.11 applies to, though we have not found such a characterization yet.

There are, of course, also convex games that do not allow for recomputing core allocations through consecutive changes. We conclude this section by showing that the cost function of the joint replenishment game does not possess the property from Definition 4.11.

Example 4.16
Consider a joint replenishment scenario involving two retailers, $N = \{1, 2\}$, that have demands $d_1 = 1, d_2 = 2$ and holding costs $h_1 = 4, h_2 = 5$, therefore $H_1 = 2$ and $H_2 = 5$. The function of joint setup costs p is given by $p(\{1\}) = 4, p(\{2\}) = 5$ and $p(\{1, 2\}) = 8$.

Now fix the permutation $\pi = (1, 2)$ of N. We have for the costs per coalition

$$\begin{aligned} c(\{1\}) &= \min_{T_1 \in \mathcal{T}^{\mathscr{L}}} \max_{0 \leq x_1 \leq 4} \frac{x_1}{T_1} + 2T_1 = 6, \\ c(\{1,2\}) &= \min_{(T_1,T_2) \in \mathcal{T}^{\mathscr{L}}_{\{1,2\}}} \max_{\substack{0 \leq x_1 \leq 4, \\ 0 \leq x_2 \leq 5, \\ 0 \leq x_1 + x_2 \leq 8}} \frac{x_1}{T_1} + 2T_1 + \frac{x_2}{T_2} + 5T_2 = 15, \end{aligned}$$

where for the first optimization problem, the optimal solution is obtained with $x_1 = 4, T_1 = 1$ and for the second: $x_1 = 4, T_1 = 1, x_2 = 4, T_2 = 1$. Therefore, a core allocation y for this scenario can be computed as $y_1 = c(\{1\}) - 0 = 6$ and $y_2 = c(\{1, 2\}) - c(\{1\}) = 9$.

Now consider a decrease of the holding costs for player 1*, so* $\bar{h}_1 = h_1 - 2 = 2$ *and* $\bar{H}_1 = H_1 - 1 = 1$*. The new costs per coalition are*

$$\bar{c}(\{1\}) = \min_{T_1 \in \mathcal{T}^{\mathscr{L}}} \max_{0 \leq x_1 \leq 4} \frac{x_1}{T_1} + 1T_1 = 4,$$

$$\bar{c}(\{1,2\}) = \min_{(T_1,T_2) \in \mathcal{T}^{\mathscr{L}}_{\{1,2\}}} \max_{\substack{0 \leq x_1 \leq 4, \\ 0 \leq x_2 \leq 5, \\ 0 \leq x_1 + x_2 \leq 8}} \frac{x_1}{T_1} + 1T_1 + \frac{x_2}{T_2} + 5T_2 = 13.5,$$

where the optimal solution of the first problem is obtained through $x_1 = 4, T_1 = 2$ *and for the second:* $x_1 = 3, T_1 = 2, x_2 = 5, T_2 = 1$*.*

Note that $\bar{c}(\{1\}) = c(\{1\}) - 2$ *and* $\bar{c}(\{1,2\}) = c(\{1,2\}) - 1.5$*. From this we obtain a new core allocation* $\bar{y}$ *with* $\bar{y}_1 = 4$ *and* $\bar{y}_2 = 9.5$*. The function values for the consecutive coalitions* $\{1\}$ *and* $\{1,2\}$ *have changed by different amounts. We conclude that there are parameter changes for the joint replenishment game that do not cause consecutive changes and therefore Definition 4.11 does not apply to that game's cost function.*

Chapter 5

Conclusion

This thesis has dealt with polymatroid optimization and its applications to game theory. We have introduced sub- and supermodular functions and looked at their importance in matroid and polymatroid theory. Furthermore, we studied linear and convex separable optimization problems over these structures. Additionally, we looked at applications of (poly-)matroid properties to cooperative and non-cooperative games. In doing so, we placed an emphasis on applying results from sensitivity analysis for polymatroid optimization. Finally, we showed how the polymatroid structure of certain cooperative games can be exploited to recompute core vectors with a constant number of changes to their components.

We have, of course, surveyed only a fraction of the available literature on the topics covered. In particular, there is a large body of research into the joint replenishment problem and its variations (for an overview, see [14]). A generalization of joint replenishment called the "one warehouse multiple retailers game" may be of special interest, as it is also shown by He et al. to be convex (see [13]), thus allowing efficient computation of core allocations.

Braess paradox and Wardrop equilibria in flow networks constitute another widely studied research area. A summary of results is given, for instance, in [10] and [19].

There are also open questions regarding the problem covered in section 4.4. Firstly, one could look for other functions that comply with Definition 4.11. All examples covered in section 4.4 allow for recomputing allocations regardless of the permutation that was used to calculate the initial core vector. One could try to find a game where only the core vector obtained from a specific permutation has the consecutive changes property. Secondly, it would be interesting to find a characterization of functions that fulfill the constraints of Definition 4.11. The examples from section 4.4 suggest that the characterization will likely be similar to modular functions.

In conclusion, we hope to have given an accessible introduction into an active research area. The problems presented in this thesis are actively studied, as is evident by the number of recent publications (e.g. [13],[10],[12]) and more findings can be expected.

Acknowledgements

First and foremost, I want to express my deepest gratitude to Prof. Dr. Britta Peis for supervising this thesis. She not only drew my interest to the topic but also encouraged me to work on open questions and took a lot of time to discuss these with me. Without her kind support this thesis would not have been realized. I could not have asked for a better advisor.

I would also like to thank Prof. Dr. Gerhard Woeginger for being second examiner on this thesis. Additionally, I thank Studienstiftung des Deutschen Volkes for their generous support during my studies. Furthermore, I am grateful to Laura Vargas Koch for her very helpful comments on an early version of the thesis.

Finally, I am indebted to my friends and family for proofreading and for believing in me.

Bibliography

[1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *J. ACM*, 55(6):25:1–25:22, 2008.

[2] M. Beckmann, C. McGuire, and C. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[4] I. J. Curiel, M. Maschler, and S. H. Tijs. Bankruptcy games. *Zeitschrift für Operations Research*, 31(5):A143–A159, 1987.

[5] Shantayanan Devarajan. A note of network equilibrium and noncooperative games. *Transportation Research Part B: Methodological*, 15(6):421 – 426, 1981.

[6] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612. ACM, 2004.

[7] A. Federgruen, M. Queyranne, and Yu-Sheng Zheng. Simple power-of-two policies are close to optimal in a general class of production/distribution networks with general joint setup costs. *Mathematics of Operations Research*, 17(4):951–963, 1992.

[8] András Frank. *Connections in Combinatorial Optimization*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2011.

[9] Satoru Fujishige. *Submodular Functions and Optimization*. Number 58 in Annals of Discrete Mathematics. Elsevier, 2nd edition, 2005.

[10] Satoru Fujishige, Michel X. Goemans, Tobias Harks, Britta Peis, and Rico Zenklusen. Matroids are immune to braess paradox. arXiv:1504.07545 [cs.GT], 2015.

[11] H. Groenevelt. Two algorithms for maximizing a separable concave function over a polymatroid feasible region. *European Journal of Operational Research*, 54(2):227 – 236, 1991.

[12] Tobias Harks, Max Klimm, and Britta Peis. Sensitivity analysis for convex seperable optimization over integral polymatroids. arXiv:1611.05372 [cs.DM], 2016.

[13] Simai He, Jiawei Zhang, and Shuzhong Zhang. Polymatroid optimization, submodularity, and joint replenishment games. *Operations Research*, 60(1):128–137, 2012.

[14] Moutaz Khouja and Suresh Goyal. A review of the joint replenishment problem literature: 1989–2005. *European Journal of Operational Research*, 186(1):1 – 16, 2008.

[15] Andreas Krause and Daniel Golovin. Submodular function maximization. In Lucas Bordeaux, Youssef Hamadi, and Pushmeet Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, chapter 3, pages 71–104. Cambridge University Press, 2014.

[16] Kazuo Murota. *Discrete Convex Analysis*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2003.

[17] Kazuo Murota and Akiyoshi Shioura. Extension of m-convexity and l-convexity to polyhedral convex functions. *Advances in Applied Mathematics*, 25(4):352 – 427, 2000.

[18] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[19] Lars Olbrich. *Aspects of Wardrop Equilibria*. PhD thesis, RWTH Aachen, 2010.

[20] Britta Peis. Algorithmic game theory, lecture notes. 2016.

[21] R.T. Rockafellar. *Convex Analysis*. Princeton landmarks in mathematics and physics. Princeton University Press, 1970.

[22] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Number 24 in Algorithms and Combinatorics. Springer, 2002.

[23] Jiawei Zhang. Cost allocation for joint replenishment models. *Operations Research*, 57(1):146–156, 2009.