Algorithmic Symplectic Packing

Greta Fischer\(^a\), Jean Gutt\(^b\), and Michael Jünger\(^a\)

\(^a\)Department of Mathematics and Computer Science, University of Cologne, Cologne, Germany; \(^b\)Institute of Mathematics, University of Toulouse, Toulouse, and INU Champollion, Albi, France

**ABSTRACT**

In this article we explore a symplectic packing problem where the targets and domains are \(2n\)-dimensional symplectic manifolds. We work in the context where the manifolds have first homology group equal to \(\mathbb{Z}^2\), and we require the embeddings to induce isomorphisms between first homology groups. In this case, Miller Maley, Mastrangeli, and Traynor showed that the problem can be reduced to a combinatorial optimization problem, namely packing certain allowable simplices into a given standard simplex. They designed a computer program and presented computational results. In particular, they determined the simplex packing widths in dimension four for up to \(k = 12\) simplices, along with lower bounds for higher values of \(k\). We present a modified algorithmic approach that allows us to determine the \(k\)-simplex packing widths for up to \(k = 13\) simplices in dimension four and up to \(k = 8\) simplices in dimension six. Moreover, our approach determines all simplex-multisets that allow for optimal packings.

**KEYWORDS**

Symplectic packings; symplectic capacities; combinatorial optimization; mixed integer linear programming; semidefinite programming

1. Introduction and motivation

Symplectic geometry originated in a mathematical formulation of the classical mechanics of dynamical systems with finitely many degrees of freedom. The objects studied are smooth manifolds with an additional structure. Formally, a symplectic manifold \((M, \omega)\) is a pair consisting of a smooth manifold \(M\) and a symplectic form \(\omega\) on \(M\). A symplectic form \(\omega\) on a manifold \(M\) is a closed nondegenerate 2-form on \(M\). A map \(\varphi : (M_1, \omega_1) \to (M_2, \omega_2)\) between two symplectic manifolds is called symplectic if it satisfies \(\varphi^* \omega_2 = \omega_1\). As Riemannian geometry is the study of transformations preserving the inner product, symplectic geometry is the study of transformations preserving the symplectic form. An important theorem in symplectic geometry, which goes back to Darboux in 1882, is that locally all symplectic forms are the same. This result implies that there are no local invariants in symplectic geometry. This is in great contrast to Riemannian geometry where curvature is an invariant that can be determined locally. So the natural question that arises is: Can we find globally defined symplectic invariants?

One of the first striking results in this direction, which lies at the root of symplectic geometry, is due to Gromov. He asked for the greatest radius of a ball that can be symplectically embedded into a given symplectic manifold. To illustrate the nontriviality of this question, he stated the famous Nonsqueezing Theorem in 1985 \([3]\). Let

\[
\begin{align*}
B^{2n}(r) &= \left\{ (x, y) \in \mathbb{R}^{2n} \mid |x|^2 + |y|^2 < \frac{r}{\pi} \right\}, \\
Z^{2n}(s) &= \left\{ (x, y) \in \mathbb{R}^{2n} \mid x_1^2 + y_1^2 < \frac{s}{\pi} \right\}
\end{align*}
\]

denote the \(2n\)-dimensional open ball of radius \(\sqrt{r/\pi}\) and the \(2n\)-dimensional open cylinder of radius \(\sqrt{s/\pi}\), respectively. Gromov’s Nonsqueezing theorem states that one cannot symplectically embed \(B^{2n}(r)\) into \(Z^{2n}(s)\) unless the radius \(r\) of the ball is less than or equal to the radius \(s\) of the cylinder. The Nonsqueezing Theorem indicates the rigidity of symplectic embeddings as compared to the flexibility of volume preserving diffeomorphisms.

The invariant found by Gromov is called the ball packing width. Let the expression \(\varphi : (M_1, \omega_1) \xrightarrow{\varphi} (M_2, \omega_2)\) denote that the map \(\varphi\) is a symplectic embedding. Then the ball packing width is formally defined as follows.

**Definition 1.1.** The ball packing width of a \(2n\)-dimensional symplectic manifold \((M, \omega)\) is

\[
g(M, \omega) = \sup \left\{ r \mid \exists \varphi : (B^{2n}(r), \omega_0) \xrightarrow{\varphi} (M, \omega) \right\}.
\]
Instead of studying only one symplectic embedding of a ball of maximum radius, one can also study $k$ symplectic embeddings of a ball with maximum radius such that the embeddings have pairwise disjoint images. The corresponding invariant is called the $k$-ball packing width.

**Definition 1.2.** The $k$-ball packing width of a $2n$-dimensional symplectic manifold $(M, \omega)$ is

$$g_k(M, \omega) = \sup \left\{ r \mid \exists \varphi_1, \ldots, \varphi_k : (B^{2n}(r), \omega_0) \rightarrow (M, \omega) \text{ with } \varphi_i(B^{2n}(r)) \cap \varphi_j(B^{2n}(r)) = \emptyset \quad \forall 1 \leq i < j \leq k \right\}.$$  

The ball packing width and the $k$-ball packing width are symplectic invariants which have been studied in [1, 9–12, 14]. They give information that can be used to distinguish symplectic manifolds. Key properties of these and other symplectic invariants were first axiomatized in 1994 by Ekeland and Hofer who introduced the notion of symplectic capacity [5]. They define a symplectic capacity as a map $c$ from the set of all symplectic manifolds to the interval $[0, \infty]$ that satisfies the following three properties:

1. Monotonicity: $(M_1, \omega_1) \hookrightarrow (M_2, \omega_2) \Rightarrow c(M_1, \omega_1) \leq c(M_2, \omega_2)$.
2. Conformality: $\forall \alpha \in \mathbb{R} \setminus \{0\} : c(M, \alpha \omega) = |\alpha| c(M, \omega)$.
3. Nontriviality: $c(B^{2n}(1), \omega_0) > 0$ and $c(Z^{2n}(1), \omega_0) < \infty$.

The nontriviality condition guarantees that, in dimension greater than two, volume is not a capacity. The search for symplectic capacities and techniques to calculate them are major areas of research in symplectic geometry. Although these invariants are quite easy to define, they are extremely difficult to calculate. For a computational approach, we replace the ball by an open prism.

We endow $P^{2n}(r)$ with the symplectic form $\omega_0$ coming from the standard symplectic form on $\mathbb{R}^{2n}$. Mastrangeli [7] has shown that, for every $\varepsilon > 0$, there exist symplectic embeddings

$$\left( (B^{2n}(r - \varepsilon), \omega_0) \right) \hookrightarrow \left( (P^{2n}(r), \omega_0) \right) \hookrightarrow \left( (B^{2n}(r), \omega_0) \right).$$

Therefore, embeddings of the ball into a symplectic manifold give rise to embeddings of the prism and vice versa. For this reason, there is no quantitative difference between looking at symplectic packings of the ball or the prism. When we restrict to symplectic manifolds that have first homology equal to $\mathbb{Z}^n$, it is possible to add the condition that the symplectic embeddings induce isomorphisms on the level of first homology. We call these maps 1-isomorphic. With this additional property we can define a slight modification of the $k$-ball packing width that allows for a computational approach.

**Definition 1.3.** The $k$-simplex packing width of a $2n$-dimensional symplectic manifold $(M, \omega)$ is

$$s_k(M, \omega) = \sup \left\{ r \mid \exists \varphi_1, \ldots, \varphi_k : (P^{2n}(r), \omega_0) \hookrightarrow (M, \omega) \text{ with } \varphi_i(P^{2n}(r)) \cap \varphi_j(P^{2n}(r)) = \emptyset \quad \forall 1 \leq i < j \leq k \right\}.$$  

The $k$-simplex packing width has been introduced by Traynor [12]. It is a symplectic invariant that satisfies a set of axioms analogous to the capacity axioms. Since we reduce the set of symplectic embeddings to those that are 1-isomorphic, the $k$-simplex packing width gives a lower bound on the $k$-ball packing width: $s_k \leq g_k$.

Miller Maley et al. calculated the $k$-simplex packing width of $P^4(1)$ by using a computer program. Their results are shown in Table 1. The $k$-ball packing width can be computed using pseudoholomorphic curves [10]. For $k > 12$ the values of $s_k(P^4(1), \omega_0)$ are known lower bounds that are conjectured to be optimal. The approach of Miller Maley et al. builds up on work of Traynor who obtained maximal packings of $B^4(1)$ by constructing maximal packings of $P^1(1)$ by hand [12]. Similar techniques were applied by Kruglikov [6]. Other explicit symplectic packings were constructed by Schlenk [11] and Wieck [14].

The implementation of the algorithmic approach that we are going to present confirms the results of [9] for $k$ up to 12 and confirms the optimality of the lower bound for the simplex packing width for $k = 13$.

Moreover, our approach allows for the determination of the simplex packing widths in dimension six for all $k < 8$.

In addition, our approach determines all $k$-cardinality simplex-multisets that allow for optimal packings. For each such multiset, an explicit optimal packing is generated. We present and discuss the number of optimal multisets as a function of $k$. 

We denote the 2-dimensional open prism, which is the Cartesian product of the $n$-dimensional torus $\mathbb{T}^n = \mathbb{R}^n/\mathbb{Z}^n$ and the $n$-dimensional open standard simplex of side length $r$

$$\Delta^n(r) = \left\{ x \in \mathbb{R}^n \mid x_i > 0 \quad \forall i \in \{ 1, \ldots, n \} \text{ and } \sum_{i=1}^n x_i < r \right\}.$$  

We endow $P^{2n}(r)$ with the symplectic form $\omega_0$ coming from the standard symplectic form on $\mathbb{R}^{2n}$. Mastrangeli [7] has shown that, for every $\varepsilon > 0$, there exist symplectic embeddings

$$\left( (B^{2n}(r - \varepsilon), \omega_0) \right) \hookrightarrow \left( (P^{2n}(r), \omega_0) \right) \hookrightarrow \left( (B^{2n}(r), \omega_0) \right).$$

Therefore, embeddings of the ball into a symplectic manifold give rise to embeddings of the prism and vice versa. For this reason, there is no quantitative difference between looking at symplectic packings of the ball or the prism. When we restrict to symplectic manifolds that have first homology equal to $\mathbb{Z}^n$, it is possible to add the condition that the symplectic embeddings induce isomorphisms on the level of first homology. We call these maps 1-isomorphic. With this additional property we can define a slight modification of the $k$-ball packing width that allows for a computational approach.

**Definition 1.3.** The $k$-simplex packing width of a $2n$-dimensional symplectic manifold $(M, \omega)$ is

$$s_k(M, \omega) = \sup \left\{ r \mid \exists \varphi_1, \ldots, \varphi_k : (P^{2n}(r), \omega_0) \hookrightarrow (M, \omega) \text{ with } \varphi_i(P^{2n}(r)) \cap \varphi_j(P^{2n}(r)) = \emptyset \quad \forall 1 \leq i < j \leq k \right\}.$$  

The $k$-simplex packing width has been introduced by Traynor [12]. It is a symplectic invariant that satisfies a set of axioms analogous to the capacity axioms. Since we reduce the set of symplectic embeddings to those that are 1-isomorphic, the $k$-simplex packing width gives a lower bound on the $k$-ball packing width: $s_k \leq g_k$.

Miller Maley et al. calculated the $k$-simplex packing width of $P^4(1)$ by using a computer program. Their results are shown in Table 1. The $k$-ball packing width can be computed using pseudoholomorphic curves [10]. For $k > 12$ the values of $s_k(P^4(1), \omega_0)$ are known lower bounds that are conjectured to be optimal. The approach of Miller Maley et al. builds up on work of Traynor who obtained maximal packings of $B^4(1)$ by constructing maximal packings of $P^1(1)$ by hand [12]. Similar techniques were applied by Kruglikov [6]. Other explicit symplectic packings were constructed by Schlenk [11] and Wieck [14].

The implementation of the algorithmic approach that we are going to present confirms the results of [9] for $k$ up to 12 and confirms the optimality of the lower bound for the simplex packing width for $k = 13$.

Moreover, our approach allows for the determination of the simplex packing widths in dimension six for all $k < 8$.

In addition, our approach determines all $k$-cardinality simplex-multisets that allow for optimal packings. For each such multiset, an explicit optimal packing is generated. We present and discuss the number of optimal multisets as a function of $k$. 
In Section 2 we describe how we model the determination of the simplex packing width $s_k(P^4(1),\omega_0)$ as a mixed integer linear program (MILP), along with the presentation and discussion of the computer implementation and the computational results. In Section 3 we present an alternative modeling as a quadratically constrained quadratic program (QCQP) which is then relaxed to a semidefinite program (SDP). In Section 4 we extend the MILP approach to the next higher dimension and compute the $k$-simplex packing width of the six-dimensional open prism.

2. MILP for $k$-simplex packing width of $P^4(1)$

2.1. Modelling

 Whereas algebraic geometry is a crucial tool to calculate the ball packing widths, the main tool to calculate the simplex packing widths is the following theorem.

**Theorem 2.1 (Packing Theorem [9])**. Let $V$ be an open, connected subset of $\mathbb{R}^n$ with $H_1(V,\mathbb{Z}) = 0$. Then

$$s_k(\mathbb{T}^n \times V, \omega_0) = \sup \left\{ r \mid \exists A_1, \ldots, A_k \in \text{GL}_n(\mathbb{Z}) \exists t_1, \ldots, t_k \in \mathbb{R}^n : \right.$$ 

$$A_i(\mathbb{T}^n(r)) + t_i \subseteq V \forall i \in \{1, \ldots, k\}$$ 

$$(A_i(\mathbb{T}^n(r)) + t_i) \cap (A_j(\mathbb{T}^n(r)) + t_j) = \emptyset$$

$$\forall 1 \leq i < j \leq k \right\}.$$ 

Here, $\text{GL}_n(\mathbb{Z})$ denotes the set of all matrices in $\mathbb{Z}^{n \times n}$ that are invertible over $\mathbb{Z}$, together with matrix multiplication as the group operation. The meaning of Theorem 2.1 is that for symplectic manifolds of the form $\mathbb{T}^n \times V$, where $V$ is an open connected subset of $\mathbb{R}^n$ with first homology equal to zero, we can compute the $k$-simplex packing width $s_k(\mathbb{T}^n \times V, \omega_0)$ by computing an optimal packing of $V$ by copies of $\mathbb{T}^n(r)$ under integral affine transformations while maximizing the sidelength $r$. This not only reduces the dimension of the problem space from $2n$ to $n$ but also converts the calculation of the $k$-simplex packing width into a classical combinatorial packing problem. By applying Theorem 2.1 to $P^4(1) = \mathbb{T}^2 \times \mathbb{R}^2(1)$, we get the following formulation.

**Problem 2.2 (Outer Optimization Problem—Combinatorial Formulation)**. Given $k \in \mathbb{N}$, determine the minimum side length $s$ such that there exist matrices $A_1, \ldots, A_k \in \text{GL}_2(\mathbb{Z})$ and vectors $t_1, \ldots, t_k \in \mathbb{R}^2$ satisfying the containment condition

$$A_i(\mathbb{T}^2(1)) + t_i \subseteq \mathbb{T}^2(s) \quad \text{for all } 1 \leq i \leq k$$

and the disjointness condition

$$(A_i(\mathbb{T}^2(1)) + t_i) \cap (A_j(\mathbb{T}^2(1)) + t_j) = \emptyset \quad \text{for all } 1 \leq i < j \leq k.$$ 

Notice that previously we were maximizing the size of the packing objects but now we are minimizing the size of the packing container. The reason for this is that the latter is easier to model. Let $g_k$ denote the minimum side length $s$ from Problem 2.2. Then we have the relation

$$s_k(P^4(1), \omega_0) = \frac{1}{g_k}.$$ 

By taking the reciprocal value of $g_k(P^4(1), \omega_0)$ in Table 1, we obtain an upper bound $\overline{g_k}$ on $g_k$ for $k = 1, \ldots, 20$. For $k \leq 12$ and $k = 16$ we even have equality $\overline{g_k} = g_k$. We only need to consider triangles whose shapes are admissible for that upper bound. For a given $k$, we call the list of admissible triangles the shapelist $S_k$. Now we can determine $\overline{s_k}$ by computing an optimal packing for every $k$-cardinality multisubset of the shapelist. The number of $k$-cardinality multisubsets from the shapelist $\binom{S_k}{k}$ according to $k$ is shown in Table 2.

| $k$ | $g_k$ | $\overline{g_k}$ |
|-----|------|------------------|
| 1   | $\sqrt{2}$ | $\sqrt{2}$        |
| 2   | $\sqrt{3}$ | $\sqrt{3}$        |
| 3   | $\sqrt{5}$ | $\sqrt{5}$        |
| 4   | $\sqrt{7}$ | $\sqrt{7}$        |
| 5   | $\sqrt{9}$ | $\sqrt{9}$        |
| 6   | $\sqrt{11}$ | $\sqrt{11}$      |
| 7   | $\sqrt{13}$ | $\sqrt{13}$      |
| 8   | $\sqrt{15}$ | $\sqrt{15}$      |
| 9   | $\sqrt{17}$ | $\sqrt{17}$      |
| 10  | $\sqrt{19}$ | $\sqrt{19}$      |
| 11  | $\sqrt{21}$ | $\sqrt{21}$      |
| 12  | $\sqrt{23}$ | $\sqrt{23}$      |
| 13  | $\sqrt{25}$ | $\sqrt{25}$      |
| 14  | $\sqrt{27}$ | $\sqrt{27}$      |
| 15  | $\sqrt{29}$ | $\sqrt{29}$      |
| 16  | $\sqrt{31}$ | $\sqrt{31}$      |
| 17  | $\sqrt{33}$ | $\sqrt{33}$      |
| 18  | $\sqrt{35}$ | $\sqrt{35}$      |
| 19  | $\sqrt{37}$ | $\sqrt{37}$      |
| 20  | $\sqrt{39}$ | $\sqrt{39}$      |
Table 2. Number of $k$-cardinality multisubsets of the shapelists.

| $k$ | $|\Delta_k|$ | $\left(\binom{|\Delta_k|}{k}\right)$ |
|-----|--------------|---------------------------------|
| 1   | 1            | 1                               |
| 2   | 8            | 36                              |
| 3   | 8            | 120                             |
| 4   | 8            | 330                             |
| 5   | 8            | 792                             |
| 6   | 8            | 1,716                           |
| 7   | 20           | 657,800                         |
| 8   | 20           | 2,220,075                       |
| 9   | 20           | 6,906,900                       |
| 10  | 20           | 20,030,010                      |
| 11  | 20           | 54,627,300                      |
| 12  | 20           | 141,120,525                     |
| 13  | 20           | 347,373,600                     |
| 14  | 20           | 818,809,200                     |
| 15  | 32           | 511,738,760,544                 |
| 16  | 32           | 1,503,232,609,098               |
| 17  | 32           | 4,244,421,484,512               |
| 18  | 32           | 11,554,258,485,616              |
| 19  | 32           | 30,405,943,383,200              |
| 20  | 32           | 77,535,155,627,160              |

Figure 1. Branch-and-bound search tree for the 3-triangle packing.

For $k = 10$ one already needs to solve more than twenty million subproblems. This clearly indicates that complete enumeration is out of the question, especially in view of the fact that only exponential time algorithms for the computation of the optimal packings are known. Rather, like [9] we implement a branch-and-bound approach. Each node of the branch-and-bound search tree represents a multisubset. The level of the node is equal to the cardinality of the multisubset. For example, the root on level zero corresponds to the empty multisubset and the children on level one correspond to all subsets of cardinality one.

The (incomplete) search tree for $k = 3$ is shown in Figure 1.

We start the search with the global upper bound obtained from Table 1. Whenever the computation for a node produces an optimum packing that exceeds the global upper bound, we can discard the subtree rooted at this node from further investigation: the node (together with its subtree) is fathomed. Whenever the computation for a node at level $k$ produces an optimum packing that improves the global upper bound, we update the global upper bound and memorize the packing as the incumbent solution. At termination the incumbent solution is an optimum packing with value of the upper bound.

Now we address how to compute an optimal packing at each node. We call this the inner optimization problem.

Problem 2.3 (Inner Optimization Problem). Given $T_1, \ldots, T_m \in \Delta_k$, determine the minimum side length $s$ such that there exist vectors $t_1, \ldots, t_m \in \mathbb{R}^2$ satisfying

$$T_i + t_i \subseteq \Delta^2(s) \quad \text{for all } i \in \{1, \ldots, m\} \quad \text{(containment condition)},$$
$$\left(T_i + t_i\right) \cap \left(T_j + t_j\right) = \emptyset \quad \text{for all } 1 \leq i < j \leq m \quad \text{(disjointness condition)}.$$

We formulate Problem 2.3 as a MILP. Every triangle from the shapelist is given in the form

$$T_i = \text{conv}\left(\left(\begin{array}{c} 0 \\ 0 \end{array}\right), \left(\begin{array}{c} a_i \\ b_i \end{array}\right), \left(\begin{array}{c} c_i \\ d_i \end{array}\right)\right) \cup \ldots \cup \left(\begin{array}{c} x_i \\ y_i \end{array}\right),$$

where $a_i, b_i, c_i, d_i$ are integer constants. Instead of working with the open sets $T_i$ and $\Delta^2(s)$, we will consider their closures. This does not make any difference for the containment condition but is easier to model. Let $t_i = \left(\begin{array}{c} x_i \\ y_i \end{array}\right)$ denote the translation vector, then the closure of the translated triangle $T_i + t_i$ is given by

$$T_i + t_i = \text{conv}\left(\left(\begin{array}{c} x_i \\ y_i \end{array}\right), \left(\begin{array}{c} x_i + a_i \\ y_i + b_i \end{array}\right), \left(\begin{array}{c} x_i + c_i \\ y_i + d_i \end{array}\right)\right).$$
and the closed standard simplex of side length \( s \) is given by

\[
\Delta^2(s) = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \mid x \geq 0, \ y \geq 0, \ x + y \leq s \right\}.
\]

Because of convexity, it suffices to check the containment condition for the three vertices only. Thus, we obtain a total of nine inequalities for every \( i \in \{1, \ldots, m\} \):

\[
\begin{align*}
    x_i & \geq 0, & \quad y_i & \geq 0, & \quad x_i + y_i & \leq s, \\
    x_i + a_i & \geq 0, & \quad y_i + b_i & \geq 0, & \quad (x_i + a_i) + (y_i + b_i) & \leq s, \\
    x_i + c_i & \geq 0, & \quad y_i + d_i & \geq 0, & \quad (x_i + c_i) + (y_i + d_i) & \leq s.
\end{align*}
\]

By putting the constants to the right hand sides and taking extreme in every column, we can reduce these nine inequalities to only three for every \( i \in \{1, \ldots, m\} \):

\[
\begin{align*}
    x_i & \geq \max \{0, -a_i, -c_i\} =: K_i, \\
    y_i & \geq \max \{0, -b_i, -d_i\} =: K_i', \\
    x_i + y_i - s & \leq \min \{0, -a_i - b_i, -c_i - d_i\} =: K_i''.
\end{align*}
\]

Thus, we get a total of \( 3m \) inequalities that we call the containment constraints.

Next, we model the disjointness condition. We observe that the equation \( (T_i + t_i) \cap (T_j + t_j) = \emptyset \) is equivalent to \( t_j - t_i \notin (T_i \ominus T_j) \). Here, \( T_i \ominus T_j \) denotes the Minkowski difference of \( T_i \) and \( T_j \) that is defined as

\[
T_i \ominus T_j = \left\{ v_i - v_j \mid v_i \in T_i, v_j \in T_j \right\}.
\]

In our setting, the Minkowski difference of two triangles \( T_i \) and \( T_j \) is of the form

\[
T_i \ominus T_j = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \mid \alpha_f x + \beta_f y < \gamma_f \quad \forall f \in \{1, \ldots, 6\} \right\},
\]

where \( \alpha_f, \beta_f, \gamma_f \in \mathbb{Z} \) are integer constants. Working with this representation, the difference vector \( t_j - t_i \) is not contained in the Minkowski difference \( T_i \ominus T_j \) if and only if at least one of the six inequalities \( \alpha_f x + \beta_f y < \gamma_f \) is violated. To model this condition, we introduce a binary variable \( z_{ij} \in \{0, 1\} \) for every \( f \in \{1, \ldots, 6\} \) with the following meaning:

\[
z_{ij} = 1 \Rightarrow \alpha_f (x_i - x_j) + \beta_f (y_i - y_j) \geq \gamma_f.
\]

Now one could by brute force enumerate all \( O\left(6^{(5)}\right) \) possible 0/1-assignments of the binary variables \( z_{ij} \) and solve a linear program for each. This strategy was pursued by Miller Maley et al. [9]. Instead, we model the implication using a Big-M-formulation, where the parameter \( M \) has to be chosen sufficiently large:

\[
\alpha_f (x_i - x_j) + \beta_f (y_i - y_j) \geq \gamma_f - M \left(1 - z_{ij}\right).
\]

In our setting we can choose \( M \) as

\[
M = \left( |\alpha_f| + |\beta_f| \right) \hat{s} + \gamma_f,
\]

where \( \hat{s} \) is the current global upper bound on \( s \) in the branch-and-bound algorithm when encountering the inner optimization problem.

We want at least one of the six binary variables \( z_{ij} \) to take the value one. Equivalently we can say that their sum should be greater or equal to one. Altogether, the disjointness condition is equivalent to

\[
\begin{align*}
    z_{ij} & \in \{0, 1\}, \\
    z_1 + \cdots + z_6 & \geq 1, \\
    \alpha_f (x_i - x_j) + \beta_f (y_i - y_j) & \geq \gamma_f - M \left(1 - z_{ij}\right)
\end{align*}
\]

for all \( f \in \{1, \ldots, 6\} \). For every pair of triangles \( T_i, T_j \) with \( 1 \leq i < j \leq m \), we have these seven inequalities, so in total there are \( 7\binom{m}{2} \) inequalities that we call the disjointness constraints. By merging the containment constraints and the disjointness constraints, we obtain an MILP with \( 1 + 2m \) continuous variables, \( 6\binom{m}{2} \) (binary) integer variables and \( 3m + 7\binom{m}{2} \) constraints.
Problem 2.4 (MILP Formulation of the Inner Optimization Problem 2.3).

\[
\begin{align*}
\min \quad & s \\
\text{s.t.} \quad & x_i \geq K_i \quad \forall i \in \{1, \ldots, m\} \\
& y_i \geq K_i' \quad \forall i \in \{1, \ldots, m\} \\
& x_i + y_i - s \leq K''_i \\
& z_{ij}^1 + \cdots + z_{ij}^k \geq 1 \\
& \alpha_{ij}^1(x_j - x_i) + \beta_{ij}^1(y_j - y_i) \geq \gamma_{ij} - M\left(1 - z_{ij}^k\right) \\
& s \in \mathbb{R}, \quad x, y \in \mathbb{R}^m, \quad z \in \{0,1\}^{6\binom{m}{2}}
\end{align*}
\]

2.2. Implementation

We keep a data base of multisets with known lower bounds or optimal solutions, initialized with the empty set. Then we run our program with input \( k = 1, 2, \ldots \) Via the data base the run for \( k \) can profit from data base queries for given multisets of cardinalities up to \( k - 1 \) rather than solving the inner optimization problem.

The program maintains a list of blocking multisets, i.e., multisets for which a lower bound exceeding the global upper bound is known.

The backbone of the program is a queue of branch-and-bound nodes. Each of these nodes represents an inner optimization instance. As long as the queue is not empty, we extract its front element. If the lower bound of the inner optimization instance exceeds the global upper bound, the node is fathomed. Otherwise we check the inner optimization instance for blocking multisets. If we detect a blocking subset, the node is fathomed. Otherwise, we check for previously known bounds. If a lower bound is found that exceeds the global upper bound, the inner optimization instance is recorded as a new blocking structure and the node is fathomed. In any other case, we run the inner optimization solver and update the database of known solutions and lower bounds.

If the solution computed by the inner optimization solver exceeds the global upper bound, the inner optimization instance is recorded as a new blocking structure and the node is fathomed. If the optimal value of the solution is smaller or equal to the global upper bound, we distinguish between two cases.

In the first case, the number of simplices in the inner optimization instance is equal to the number \( k \) of simplices in the outer optimization problem, which means the node is a leaf of the branch-and-bound tree. If the optimal value of the solution is equal to the current global upper bound, we draw a picture of the solution and append it to a \( \text{MIP}\)-file. This \( \text{MIP}\)-file already contains pictures of all previous solutions with the same optimal value. If the optimal value of the solution is strictly smaller than the global upper bound, we update the latter and replace the old \( \text{MIP}\)-file by a new \( \text{MIP}\)-file containing a picture of the solution.

In the second case, the number of simplices in the inner optimization instance is smaller than the number \( k \) of simplices in the outer optimization problem, which means the node is not a leaf. If so, we add all feasible extensions of the inner optimization instance to the queue.

All described steps are summarized in Figure 2. The computer code consists of 1841 lines written in the programming language C, 406 of which (\( \approx 22\% \)) concern the branch-and-bound framework. The code for the inner optimization procedure has 471 lines (\( \approx 26\% \)). Reading and rewriting the bounds file and the \( \text{MIP}\)-file takes 204 lines (\( \approx 11\% \)) and 162 lines (\( \approx 9\% \)), respectively.

2.3. Results

The computational experiments were carried out under the Debian 10 operating system on two Intel Xeon E5-2690v2 CPUs with 3.00 GHz and 10 cores each. For solving the inner optimization problem, we call the GUROBI Optimization Software Version 9.0.3. All computation times are wall clock times.

Before we discuss the timing statistics of the algorithm, we take a look at some of the computed packings. Figure 3 and 4 show one exemplary optimal packing for \( k = 1, \ldots, 13 \). For \( k = 1, \ldots, 12 \) we confirm the values of \( s_{k,0}^n \) found by Miller Maley et al. that were presented in Table 1. For \( k = 13 \) we can verify their conjecture that the upper bound they found is indeed optimal.

Our program does not only compute one optimal packing for a given \( k \) but is able to detect all multisubsets that allow for an optimal packing. We denote such multisubsets as optimal. In Table 3 we report the number of optimal multisubsets for \( k = 1, \ldots, 13 \). For some multisubsets there exist more than one optimal packing.

An inspection of Table 3 raises the natural question why the number of optimal multisubsets increases or decreases significantly for certain \((k, k+1)\) pairs. In order to shine some light on this phenomenon, let us consider sequences for which the optimal container size \( s_{k,0}^n \) does not change, e.g., \( k = 2, 3, 4 \) or \( k = 7, 8, 9 \). Assume \( k \) and \( k - 1 \) are in the considered sequence.

Henceforth, we abbreviate “multiset of cardinality \( k \)” to “\( k \)-multiset.” From every optimal \( k \)-multiset with \( s \) distinct shapes we obtain \( s \) optimal \((k - 1)\)-multisets (removing one of the first shape or one of the second shape or ... or one of the \( s \)-th shape).

For \( k = 4 \) we have the 4 optimal multisets shown in Figure 5. If we apply the procedure above, the first gives rise to 2 optimal 3-multisets, the other three to 3 optimal 3-multisets, in total exactly 11 multisets, the same that we had calculated before for \( k = 3 \).
Now let us consider the pair $k = 2, 3$. If we apply the procedure to the 11 optimal multisets for $s_k = 2$, we obtain a total of 24 multisets including duplicates: One of them appears eight times, six appear twice, four appear once. After removal of the duplicates we obtain the 11 optimal multisets for $k = 2$.

While the above examination of the $k = 2, 3, 4$ sequence has been experimental mathematics with pencil and paper, we needed computer assistance for an analogous examination of the $k = 7, 8, 9$ sequence.

For $k = 9$ we have 47 optimal multisets. Application of the procedure gives exactly the 261 multisets that we had calculated before for $k = 8$, again with no duplicates. If we apply the procedure to these 261 8-multisets, we obtain 1361 7-multisets including duplicates. After duplicate removal 646 remain. Beyond these 646, the list of the 668 optimal 7-multisets contains 22 multisets that have no extensions to optimal 8-multisets. Since we have a picture of an optimal packing for each optimal multiset we have computed in this study, we could inspect these 22 multisets, one of which is shown in Figure 6 along with an optimal packing.\footnote{At this point we can’t refrain from telling that a colored version of the picture in Figure 6 is the logo of the SFB/TRR 191 “Symplectic Structures in Geometry, Algebra and Dynamics” of the German Science Foundation that has supported our research. When it was chosen in 2016, we had no idea of its rare feature, we just found it visually pleasing.}

All 22 multisets share the distinct feature that the free spaces in the optimal packing consist of three triangles rather than two for the other 646 packings whose extendability to 8-packings is visually obvious.

While it is tempting to conjecture that the procedure applied to a full packing (here $k = 4$ and $k = 9$) gives us the full list of optimum multisets for the next lower $k$ (here 3 and 8), we shall see in Section 4 that this is not true in general.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{flowchart.png}
\caption{Flowchart of our program.}
\end{figure}
Figure 3. Optimal $k$-triangle packings for $k = 1, \ldots, 9$.

Table 3. Number of multisubsets that allow for an optimal $k$-triangle packing for $k = 1, \ldots, 13$.

| $k$ | #Optimal multisubsets |
|-----|------------------------|
| 1   | 1                      |
| 2   | 11                     |
| 3   | 11                     |
| 4   | 4                      |
| 5   | 18                     |
| 6   | 21                     |
| 7   | 668                    |
| 8   | 261                    |
| 9   | 47                     |
| 10  | 198                    |
| 11  | 142                    |
| 12  | 78                     |
| 13  | 166                    |

Table 4 shows the timing statistics of our algorithm for $k = 1, \ldots, 13$. The column labels are:

- $k$: the number of triangles to be packed,
- $\binom{|S|}{k}$: the number of $k$-cardinality multisubsets of the shapelist,
- #I-Calls: the number of calls to the inner optimization procedure,
- Avg I-Time: the average cpu time spent in an inner optimization procedure,
- Max I-Time: the maximum cpu time spent in an inner optimization procedure,
- Total Time: the total cpu time.

The number of calls to the inner optimization procedure is significantly smaller than the number of $k$-cardinality multisubsets of the shapelist that would have been considered using complete enumeration. Due to the database we built up from previous runs for smaller $k$, many calls of the inner optimization procedure have been replaced by simple database queries.
Figure 4. Optimal $k$-triangle packings for $k = 10, \ldots, 13$.

One can see that the inner optimization procedure is quite fast on average but a few instances with larger values of $k$ can take very long, especially for $k = 13$. We have explored several improvement strategies to the algorithmic approach. The first improvement strategy consists of reducing the occurrence of symmetric solutions for the inner optimization problem. For example, consider the inner optimization instance $\{T_1, T_1, T_1, T_4\}$. As can be seen in Figure 7, we obtain six symmetric optimal solutions for this multisubset.

In general, for each shape with multiplicity $\mu$ we obtain $\mu!$ symmetric solutions. Our approach to reduce the occurrence of symmetric solutions is to add artificial symmetry breaking inequalities to the MILP version of the inner optimization problem. We have examined three different types of symmetry breaking inequalities. All of them try to put an order on the displacement variables $(x_i, y_i)$ belonging to triangles of the same shape. To compare the three symmetry breaking inequality types, we calculated all optimal $k$-triangle packings for $k = 1, \ldots, 12$. Table 5 shows the computation time of our algorithm combined with the corresponding symmetry breaking constraint. The inequality type numbers are given in the following order:

- Type 0: No symmetry breaking inequality applied.
- Type 1: $x_i \leq x_{i+1}$.
- Type 2: $x_i + y_i \leq x_{i+1} + y_{i+1}$.
- Type 3: $(x_i \leq x_{i+1}) \land (x_i = x_{i+1} \Rightarrow y_i \leq y_{i+1})$.

For $k \geq 10$, one can see that applying symmetry breaking constraints is an enormous improvement to the run-time of the algorithm. For small values of $k$, there is no significant difference between the three symmetry breaking constraints, but for $k = 12$ the clear winner is the symmetry breaking type 2 of the form $x_i + y_i \leq x_{i+1} + y_{i+1}$. For the run producing the computational results presented in Table 4, we also chose this symmetry breaking type.

The second improvement strategy consist in applying a time limit to the inner optimization procedure and changing the GUROBI parameter settings for instances exceeding the given time limit. The GUROBI Optimizer provides a wide variety of parameters that allow to control the operation of the optimization engines. To find parameter values that improve the performance on our model, we both manually tested different parameter settings and used the built-in automated parameter tuning tool. We found that there are
four parameters that speed up the computation time for most instances. Table 6 gives an overview of the four parameters together with their default value and the modified value.

For the $k = 13$ computation, we reapplied the inner optimization procedure under the modified parameter setting on the instances that exceeded a time limit of 30 seconds. This decreased the total computation time by approximately 38% compared to applying no time limit and using the default parameter setting. The drawback is that we only detect 165 instead of all 166 optimal multisets. In contrast, when applying the modified parameter setting to all inner optimization instances, the computation time increases by approximately 26%. The results are summarized in Table 7.

The third improvement strategy is to reformulate the inner optimization problem as a QCQP. Once this formulation is achieved, we can consider the semidefinite relaxation of this program. The hope is that the semidefinite relaxation gives us strong lower bounds
Table 4. Timing statistics for the $k$-triangle packing given in the format “hh:mm:ss” for $k = 1, \ldots, 13$.

| $k$ | $(\binom{k}{2})$ | #I-Calls | Avg I-Time | Max I-Time | Total Time |
|-----|------------------|----------|------------|------------|------------|
| 1   | 1                | 1        | 0:00:00.00 | 0:00:00.00 | 0:00:00.00 |
| 2   | 36               | 43       | 0:00:00.00 | 0:00:00.01 | 0:00:00.02 |
| 3   | 120              | 11       | 0:00:00.00 | 0:00:00.02 | 0:00:00.03 |
| 4   | 330              | 11       | 0:00:00.00 | 0:00:00.09 | 0:00:00.14 |
| 5   | 792              | 433      | 0:00:00.01 | 0:00:00.27 | 0:00:03.11 |
| 6   | 1716             | 185      | 0:00:00.04 | 0:00:00.33 | 0:00:06.72 |
| 7   | 657 800          | 255 158  | 0:00:10.00 | 0:00:13.30 | 0:18:36.90 |
| 8   | 2 220 075        | 263      | 0:00:00.14 | 0:00:04.46 | 0:12:29.10 |
| 9   | 6 906 900        | 47       | 0:00:00.09 | 0:00:01.64 | 0:11:59.33 |
| 10  | 20 030 010       | 34 029   | 0:00:00.52 | 0:02:27.23 | 4:56:28.02 |
| 11  | 54 627 300       | 43 187   | 0:00:07.67 | 0:46:18.30 | 92:05:28.83 |
| 12  | 141 120 525      | 129 630  | 0:00:09.39 | 3:26:34.38 | 338:19:31.65 |
| 13  | 347 373 600      | 196 735  | 0:00:37.88 | 46:59:29.37 | 2070:08:23.43 |

Table 5. Computation time of the $k$-triangle packing involving symmetry breaking constraints given in the format “hh:mm:ss” for $k = 1, \ldots, 12$.

| $k$ | Type 0 | Type 1 | Type 2 | Type 3 |
|-----|--------|--------|--------|--------|
| 1   | 0:00:00.00 | 0:00:00.00 | 0:00:00.00 | 0:00:00.00 |
| 2   | 0:00:00.04 | 0:00:00.04 | 0:00:00.02 | 0:00:00.03 |
| 3   | 0:00:00.05 | 0:00:00.03 | 0:00:00.03 | 0:00:00.01 |
| 4   | 0:00:00.36 | 0:00:00.14 | 0:00:00.14 | 0:00:00.02 |
| 5   | 0:00:05.67 | 0:00:03.16 | 0:00:03.11 | 0:00:02.88 |
| 6   | 0:00:12.71 | 0:00:09.37 | 0:00:06.72 | 0:00:08.62 |
| 7   | 0:28:47.35 | 0:17:42.80 | 0:18:56.90 | 0:14:33.28 |
| 8   | 0:12:47.89 | 0:12:25.75 | 0:12:29.10 | 0:12:30.34 |
| 9   | 0:11:57.62 | 0:11:57.32 | 0:11:59.33 | 0:11:59.01 |
| 10  | 16:04:10.11 | 5:33:28.68 | 4:56:28.02 | 5:29:36.59 |
| 11  | 11:03:55.85 | 11:21:41.18 | 9:20:28.83 | 9:30:24.37 |
| 12  | 40:06:41.10 | 50:63:59.92 | 33:19:31.65 | 53:05:44.44 |

Table 6. Modified GUROBI parameter setting.

| Parameter      | Description                 | Default value | Modified value |
|----------------|-----------------------------|---------------|----------------|
| Presolve       | Controls the presolve level | −1            | 2              |
| PreDual        | Controls presolve model dualization | −1 | 1 |
| MIPFocus       | MIP solver focus            | 0             | 2              |
| Heuristics     | Time spent in feasibility heuristics | 0.05 | 0 |

Table 7. Timing statistics for the 13-triangle packing under time limit to the inner optimization procedure combined with modified GUROBI parameter setting given in the format “hh:mm:ss.”

| Time limit | Parameter setting | Avg I-time | Total time | #Optimal multisubsets | #Open problems |
|-----------|------------------|------------|------------|-----------------------|----------------|
| $\infty$  | default          | 0:00:37.88 | 2070:08:23.43 | 166                   | 0              |
| $\infty$  | modified         | 0:00:47.74 | 2608:06:35.65 | 166                   | 0              |
| 30s       | default          | 0:00:18.49 | 1001:34:32.45 | 160                   | 1650           |
| 30s       | modified         | 0:00:23.31 | 1274:41:01.53 | 165                   | 1486           |

on $s$ such that the exact formulation coming from the MILP only needs to be applied if the bounds are not strong enough. This could in turn speed up the branch-and-bound process of the outer optimization problem. We will describe this approach in the next section.
3. SDP for k-simplex packing width of $P^4(1)$

3.1. Modelling

We have three different approaches to model the inner optimization problem as a QCQP. We will successively describe each approach and conclude with a comparison of the results.

In the first approach we model the containment condition just like we did in the MILP formulation. Instead of expressing the disjointness condition by a Big-$M$-inequality, we will use the quadratic inequality

$$z^\|_{ij}^\| \left( \alpha^\|_{ij}(x_j - x_i) + \beta^\|_{ij}(y_j - y_i) - r^\|_{ij} \right) \geq 0.$$  

We formulate the binary constraint $z^\|_{ij} \in \{0, 1\}$ by the equation $z^\|_{ij}(1 - z^\|_{ij}) = 0$. Since we want at least one of the variables $z^\|_{ij}$ to take the value one, we require $z^\|_{ij} + \cdots + z^\|_{6} \geq 1$. In order to obtain a semidefinite relaxation, we desire the QCQP to be in homogeneous form. To eliminate the linear terms, we introduce a variable $t \in \mathbb{R}$ with $t^2 = 1$ and multiply all linear terms by it. Thus, we obtain the following QCQP formulation of the inner optimization problem.

**Problem 3.1 (Inner Optimization Problem—QCQP 1).**

$$\begin{align*}
\min & \quad ts \\
\text{s.t.} & \quad tx_i \geq K_i, & \forall i \in \{1, \ldots, m\} \\
& \quad ty_i \geq K'_j, & \forall i \in \{1, \ldots, m\} \\
& \quad tx_i + ty_i - ts \leq K''_j, & \forall i \in \{1, \ldots, m\} \\
& \quad tz^\|_{ij} + \cdots + tz^\|_{6} \geq 1, & \forall 1 \leq i < j \leq m \\
& \quad z^\|_{ij} \left( \alpha^\|_{ij}(x_j - x_i) + \beta^\|_{ij}(y_j - y_i) - t\gamma^\|_{ij} \right) \geq 0, & \forall 1 \leq i < j \leq m; \forall f \in \{1, \ldots, 6\} \\
& \quad z^\|_{ij} \left( t - z^\|_{ij} \right) = 0, & \forall 1 \leq i < j \leq m; \forall f \in \{1, \ldots, 6\} \\
& \quad t^2 = 1 \\
& \quad s \in \mathbb{R}, \quad x, y \in \mathbb{R}^m, \quad z \in \mathbb{R}^6(\mathbb{R}), \quad t \in \mathbb{R}.
\end{align*}$$

The second approach uses the separating hyperplane theorem.

**Theorem 3.2 (Separating Hyperplane Theorem [13]).** Let $C, D \subseteq \mathbb{R}^n$ be nonempty disjoint convex sets. Then there exist a nonzero vector $a \in \mathbb{R}^n$ and a real number $b$ such that $a^Tx \leq b$ for all $x \in C$ and $a^Tx \geq b$ for all $x \in D$.

We can directly apply Theorem 3.2, illustrated in Figure 8, to the disjointness condition

$$T_i + T_i \cap T_j = \emptyset.$$

By Theorem 3.2, the two sets are disjoint if there exist a nonzero vector $\left( \begin{array}{c} \alpha^\| \\ \beta^\| \end{array} \right)$ and a real number $\gamma^\|$ such that

$$\begin{align*}
\alpha^\|x_i + \beta^\|y_i \leq \gamma^\|, \\
\alpha^\|(x_i + a_i) + \beta^\|(y_i + b_i) \leq \gamma^\|, \\
\alpha^\|(x_i + c_i) + \beta^\|(y_i + d_i) \leq \gamma^\|. \\
\end{align*}$$

Because of convexity, the inequalities hold for the entire set $T_i + T_i$ and $T_j + T_j$ if they are satisfied by their vertices. To ensure that $\left( \begin{array}{c} \alpha^\| \\ \beta^\| \end{array} \right)$ is nonzero, we impose the additional constraint $\left( \alpha^\| \right)^2 + \left( \beta^\| \right)^2 = 1$. Together with the three containment constraints we obtain the following QCQP.
Problem 3.3 (Inner Optimization Problem—QCQP 2).

\[
\begin{align*}
\min & \quad s \\
\text{subject to} & \quad tx_i \geq K_i \quad \forall i \in \{1, \ldots, m\} \\
& \quad ty_i \geq K'_i \quad \forall i \in \{1, \ldots, m\} \\
& \quad tx_i + ty_i - ts \leq K''_i \quad \forall i \in \{1, \ldots, m\} \\
& \quad (\alpha^i)^2 + (\beta^i)^2 = 1 \quad \forall 1 \leq i < j \leq m \\
& \quad \alpha^i(x_i + t\alpha_i) + \beta^i(y_i + t\beta_i) \leq ty^j \quad \forall 1 \leq i < j \leq m \\
& \quad \alpha^i(x_i + t\alpha_i) + \beta^i(y_i + t\beta_i) \leq ty^j \quad \forall 1 \leq i < j \leq m \\
& \quad \alpha^i(x_i + t\alpha_i) + \beta^i(y_i + t\beta_i) \geq ty^j \quad \forall 1 \leq i < j \leq m \\
& \quad \alpha^i(x_i + t\alpha_i) + \beta^i(y_i + t\beta_i) \geq ty^j \quad \forall 1 \leq i < j \leq m \\
& \quad t^2 = 1 \\
& \quad s \in \mathbb{R}, \quad x, y \in \mathbb{R}^m, \quad \alpha, \beta, \gamma \in \mathbb{R}^m, \quad t \in \mathbb{R} \\
\end{align*}
\]

The third approach utilizes Farkas Lemma.

**Theorem 3.4 (Farkas Lemma [2]).** Let \( A \in \mathbb{R}^{m \times n} \) be a matrix and \( b \in \mathbb{R}^m \) be a vector. There exists a vector \( x \in \mathbb{R}^n \) such that \( x \geq 0 \) and \( Ax = b \) if and only if there does not exist a vector \( y \in \mathbb{R}^m \) such that \( y^T A \geq 0 \) and \( y^T b = -1 \).

As seen before the disjointness constraint \( T_i \cap T_j = \emptyset \) is equivalent to \( t_j - t_i \notin T_i \cup T_j \). Previously, we described the Minkowski difference as an intersection of halfspaces. This time we will describe it as the convex hull of its vertices:

\[
T_i \cup T_j = \text{int} \left( \text{conv} \left( \left\{ \left( \begin{array}{c} 0 \\ a_i \\ c_i \\ \ldots \\ b_i \\ \ldots \\ d_i \\ \ldots \\ \gamma \end{array} \right) : \left( \begin{array}{cccccccc} 0 & a_i & c_i & \ldots & b_i & \ldots & d_i & \gamma \\ 0 & b_i & -b_j & \ldots & b_j & \ldots & d_j & \gamma \\ 1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 1 \end{array} \right) \right\} \right) ight).
\]

We know that the difference vector \( t_j - t_i \) is not an element of the Minkowski difference \( T_i \cup T_j \) if and only if it cannot be written as a convex combination of its vertices. More formally, there do not exist \( \lambda^i_1, \ldots, \lambda^i_n \geq 0 \) with \( \lambda^i_1 + \cdots + \lambda^i_n = 1 \) such that

\[
\left( \begin{array}{c} x_j - x_i \\ y_j - y_i \end{array} \right) = \lambda^i_1 \left( \begin{array}{c} 0 \\ a_i \\ c_i \\ \ldots \\ b_i \\ \ldots \\ d_i \\ \gamma \end{array} \right) + \cdots + \lambda^i_n \left( \begin{array}{c} 0 \\ a_i \\ c_i \\ \ldots \\ b_i \\ \ldots \\ d_i \\ \gamma \end{array} \right) + \lambda^i_n \left( \begin{array}{c} 0 \\ a_i \\ c_i \\ \ldots \\ b_i \\ \ldots \\ d_i \\ \gamma \end{array} \right).
\]

Written in matrix form, this means that there does not exist a vector \( \lambda^i \in \mathbb{R}^9 \) such that \( \lambda^i \geq 0 \) and

\[
\begin{bmatrix}
0 & a_i & c_i & \ldots & b_i & \ldots & d_i & \gamma \\
0 & b_i & -b_j & \ldots & b_j & \ldots & d_j & \gamma \\
1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 1 
\end{bmatrix}
\begin{bmatrix}
\lambda^i_1 \\
\vdots \\
\lambda^i_n \\
\end{bmatrix}
= \begin{bmatrix}
x_j - x_i \\
y_j - y_i \\
1 
\end{bmatrix}.
\]

Then, by Farkas Lemma, there must exist a vector \( \mu^i \in \mathbb{R}^3 \) such that

\[
\begin{bmatrix}
0 & a_i & c_i & \ldots & b_i & \ldots & d_i & \gamma \\
0 & b_i & -b_j & \ldots & b_j & \ldots & d_j & \gamma \\
1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 1 
\end{bmatrix}
\begin{bmatrix}
\mu^i_1 \\
\vdots \\
\mu^i_n \\
\end{bmatrix}
\geq 0.
\]
and

\[
(\mu^g)^T \begin{bmatrix}
    x_j - x_i \\
    y_j - y_i \\
    1
\end{bmatrix} = -1.
\]

From the second constraint we derive \(\mu^g_j = -1 - \mu^g_1 (x_j - x_i) - \mu^g_2 (y_j - y_i)\). By plugging this expression into the first constraint and renaming \(\mu^g_1\) to \(\alpha^g\) and \(\mu^g_2\) to \(\beta^g\), we obtain the following QCQP.

**Problem 3.5 (Inner Optimization Problem—QCQP 3).**

\[
\begin{align*}
\min \quad & s \\
\text{subject to} \quad & tx_i \geq K_i \quad \forall i \in \{1, \ldots, m\} \\
& ty_i \geq K'_i \quad \forall i \in \{1, \ldots, m\} \\
& t x_i + t y_i - ts \leq K''_i \\
& -\alpha^g (x_j - x_i) - \beta^g (y_j - y_i) \geq 1 \\
& \alpha^g (t a_i - x_i) + \beta^g (t b_i - y_i) \geq 1 \\
& \alpha^g (t c_i - x_i) + \beta^g (t d_i - y_i) \geq 1 \\
& \alpha^g (-t a_j - x_j) + \beta^g (-t b_j - y_j) \geq 1 \\
& \alpha^g (t (a_i - a_j) - x_j + x_i) + \beta^g (t (b_i - b_j) - y_j + y_i) \geq 1 \\
& \alpha^g (t (c_i - c_j) - x_j + x_i) + \beta^g (t (d_i - d_j) - y_j + y_i) \geq 1 \\
& \alpha^g (t (c_i - c_j) - x_j + x_i) + \beta^g (t (d_i - d_j) - y_j + y_i) \geq 1 \\
& t^2 = 1
\end{align*}
\]

\[\begin{align*}
s \in \mathbb{R}, \quad & x, y \in \mathbb{R}^m, \quad \alpha, \beta \in \mathbb{R}^{(m)}_+, \quad t \in \mathbb{R}
\end{align*}\]

### 3.2. Implementation and Results

Table 8 shows the optimal value of the MILP and the three SDPs and Table 9 shows the corresponding computation times of the four optimization problems. The instances shown in the first column were randomly chosen. For solving the SDPs, we call the MOSEK Optimization Software Version 8.1.

For the first two and the last inner optimization instances, the optimal values of the three semidefinite relaxations differ. For the remaining inner optimization instances the optimal values coincide. The computation time of the first semidefinite relaxation is slightly larger than the computation time of the other two relaxations but much faster than the computation time of the MILP. However, the gap between the exact value of the inner optimization problem obtained by the MILP and the best lower bound obtained by the semidefinite relaxations is too large for a successful application in our branch and bound framework.

### 4. MILP for \(k\)-simplex packing width of \(P^6(1)\)

In this section we will extend the MILP approach to the next higher dimension and compute the \(k\)-simplex packing width of the six-dimensional open prism \(P^6(1) = \triangle^3(1) \times \mathbb{T}^3\). The outer and inner optimization problem are defined analogously.

| Random instance | MILP | SDP 1 | SDP 2 | SDP 3 |
|-----------------|------|-------|-------|-------|
| 1               | 3.89 | 3.00  | 1.00  | 1.00  |
| 2               | 3.92 | 3.00  | 2.00  | 2.00  |
| 3               | 3.94 | 2.00  | 2.00  | 2.00  |
| 4               | 3.83 | 2.00  | 2.00  | 2.00  |
| 5               | 3.81 | 2.00  | 2.00  | 2.00  |
| 6               | 3.80 | 2.00  | 2.00  | 2.00  |
| 7               | 3.80 | 2.00  | 2.00  | 2.00  |
| 8               | 3.83 | 2.00  | 2.00  | 2.00  |
| 9               | 4.00 | 3.00  | 3.00  | 3.00  |
| 10              | 3.92 | 3.00  | 2.00  | 2.00  |
Table 9. Computation time of the MILP and the three SDPs for different inner optimization instances given in the format “hh:mm:ss”.

| Computation time | MILP       | SDP 1  | SDP 2  | SDP 3  |
|------------------|------------|--------|--------|--------|
| 0:13:26.6        | 00:00:03.7 | 00:00:00.4 | 00:00:00.3 |
| 0:07:06.2        | 00:00:04.2 | 00:00:00.7 | 00:00:00.4 |
| 1:13:16.2        | 00:00:07.1 | 00:00:00.7 | 00:00:00.5 |
| 1:08:18.9        | 00:00:04.2 | 00:00:00.6 | 00:00:00.4 |
| 0:56:57.6        | 00:00:06.1 | 00:00:00.6 | 00:00:00.5 |
| 0:46:36.9        | 00:00:01.4 | 00:00:00.6 | 00:00:00.4 |
| 0:45:26.9        | 00:00:08.0 | 00:00:00.6 | 00:00:00.5 |
| 0:39:21.5        | 00:00:09.9 | 00:00:00.6 | 00:00:00.5 |
| 1:17:29.6        | 00:00:12.4 | 00:00:00.6 | 00:00:00.5 |
| 0:19:24.2        | 00:00:19.7 | 00:00:00.6 | 00:00:00.5 |

The instances are in the same order as in Table 8.

Table 10. Number of $k$-cardinality multisubsets of the shapelist for $k = 1, \ldots, 8$.

| $k$ | $|S_k^\triangle|$ | $\left(\left|S_k^\triangle\right|/k\right)$ |
|-----|-----------------|-----------------|
| 1   | 1               | 1               |
| 2   | 73              | 2701            |
| 3   | 73              | 67525           |
| 4   | 73              | 1282975         |
| 5   | 73              | 19757815        |
| 6   | 73              | 256851595       |
| 7   | 73              | 2898753715      |
| 8   | 73              | 28987537150     |

The inner optimization procedure is very fast on all instances. For computing $k$-tetrahedron packings for $k \geq 9$, the difficulty rather consists in the high number of feasible multisubsets. For $k = 9$ the cardinality of the shapelist increases to 854 and the number of multisubsets thereof increases to $694,392,240,786,929,755,070 \approx 7 \times 10^{20}$.

Therefore, computing $s_9^\triangle$ seems out of reach. Nevertheless, instead of working with the shapelist $S_9^\triangle$, one might consider the smaller shapelist $S_8^\triangle$ to compute upper bounds on $s_9^\triangle$.

5. Discussion

Concerning the semidefinite relaxation described in Section 3 there are still two open questions. First, for all investigated instances the optimal values of the second semidefinite program and third semidefinite program coincide. So the natural question that arises is...
Figure 9. Optimal $k$-tetrahedron packings for $k = 1, \ldots, 8$. 
Table 11. Number of multisubsets that allow for an optimal $k$-tetrahedron packing for $k = 1, \ldots, 8$.

| $k$ | #Optimal multisubsets |
|-----|------------------------|
| 1   | 1                      |
| 2   | 1123                   |
| 3   | 4871                   |
| 4   | 9914                   |
| 5   | 11709                  |
| 6   | 8075                   |
| 7   | 2899                   |
| 8   | 408                    |

Table 12. Statistics on all derived versus all optimal multisets.

| $k \rightarrow (k - 1)$ | Number of derived multisets | Number of including duplicates | Number of without duplicates | Number of optimal multisubsets | Number of nonextendable multisubsets |
|-------------------------|-----------------------------|--------------------------------|-------------------------------|-------------------------------|-----------------------------------|
| 8 → 7                   | 2835                        | 2835                           | 2899                          | 64                            |
| 7 → 6                   | 17755                       | 8071                           | 8075                          | 4                             |
| 6 → 5                   | 43353                       | 11709                          | 11709                         | 0                             |
| 5 → 4                   | 54429                       | 9914                           | 9914                          | 0                             |
| 4 → 3                   | 38459                       | 4871                           | 4871                          | 0                             |
| 3 → 2                   | 14539                       | 1123                           | 1123                          | 0                             |

Table 13. Timing statistics for the $k$-tetrahedron packing given in the format "hh:mm:ss" for $k = 1, \ldots, 8$.

| $k$ | $(\left| \mathcal{S}_k \right|)$ | #I-calls | Avg I-time | Max I-time | Totaltime |
|-----|---------------------------------|-----------|------------|------------|-----------|
| 1   | 1                               | 1         | 0:00:00.00 | 0:00:00.00 | 0:00:00.00 |
| 2   | 2701                            | 2773      | 0:00:00.00 | 0:00:00.07 | 0:00:10.90 |
| 3   | 67525                           | 4871      | 0:00:00.04 | 0:00:06.59 | 0:00:06.59 |
| 4   | 1282975                         | 9914      | 0:00:00.14 | 0:00:25.83 | 0:00:30.67 |
| 5   | 19757815                        | 13118     | 0:00:10.15 | 0:00:06.31 | 0:00:16.46 |
| 6   | 256851595                       | 8075      | 0:00:00.09 | 0:01:01.06 | 0:01:01.06 |
| 7   | 8496753715                      | 2899      | 0:00:00.35 | 0:00:39.66 | 0:00:40.01 |
| 8   | 84967537150                     | 408       | 0:00:00.08 | 0:00:12.77 | 0:00:12.77 |

**Question 5.1.** Are the semidefinite relaxations of Problems 3.3 and 3.5 equivalent?

Second, for all investigated instances the optimal values of the three semidefinite programs are integer which raises the question

**Question 5.2.** Can the semidefinite relaxations of Problems 3.1, 3.3, and 3.5 be reduced to integer programs?

In view of [4], a natural question is

**Question 5.3.** How many nonequivalent maximal symplectic packings of the ball by equal balls exist?

Note that, at the time of writing, the only answers to question 5.3 are in dimension 4 [8]. The “natural” definition of equivalence is: Two packings $\varphi, \psi : \mathcal{L}_k B(r) \leftrightarrow B(1)$ are equivalent if there exist a symplectomorphism $\phi : B(1) \rightarrow B(1)$ such that $\phi \circ \psi (\mathcal{L}_k B(r)) = \psi (\mathcal{L}_k B(r))$. For this notion, the answer to question 5.3 is 1. Indeed,

**Theorem 5.4** [8]. All symplectic embeddings $\varphi, \psi : \mathcal{L}_k B^4(r) \leftrightarrow B^4(1)$ are isotopic through symplectic embeddings.

One can think of other notions of equivalence (for instance affine transformations mapping a packing onto another one or isometries,...) but since none seems more natural than the others we won’t do it here.

**Acknowledgments**

We gratefully acknowledge the advice of Hansjörg Geiges and Felix Schlenk.
Funding

This research is part of a project in the SFB/TRR 191 “Symplectic Structures in Geometry, Algebra and Dynamics,” funded by the DFG.

References

[1] Biran, P. (1996). Symplectic packing in dimension 4. Geom. Funct. Anal. 7: 420–437.
[2] Farkas, J. (1902). Theorie der einfachen ungleichungen. J. Reine Angew. Math. 124: 1–27.
[3] Gromov, M. (1985). Pseudo holomorphic curves in symplectic manifolds. Invent. Math. 82: 307–348.
[4] Gutt, J., Usher, M. (2019). Symplectically knotted codimension-zero embeddings of domains in $\mathbb{R}^4$. Duke Math. J. 168: 2299–2363.
[5] Hofer, H. and Zehnder, E. (2012). Symplectic Invariants and Hamiltonian Dynamics. Basel: Birkhäuser.
[6] Kruglikov, B. S. (1996). A remark on symplectic packings. Dokl. Akad. Nauk. 350: 730–734.
[7] Mastrangeli, J. B. (1997). Symplectic packings in cotangent bundles of tori. PhD thesis, Bryn Mawr College.
[8] McDuff, D. (1998). From symplectic deformation to isotopy. Topics in symplectic 4-manifolds (Irvine, CA, 1996), First Int. Press Lect. Ser., I, Int. Press, Cambridge, MA, 85–99.
[9] Miller Maley, F., Mastrangeli, J. B., Traynor, L. (2000). Symplectic packings in cotangent bundles of tori. Exp. Math. 9: 435–455.
[10] McDuff, D., Polterovich, L. (1994). Symplectic packings and algebraic geometry. Invent. Math. 115: 405–430.
[11] Schlenk, F. (2005). Packing symplectic manifolds by hand. J. Symplectic. Geom. 3: 313–340.
[12] Traynor, L. (1995). Symplectic packing constructions. J. Differ. Geom. 42: 09.
[13] Vandenberghe, L. Boyd, S. (1996), Semidefinite programming. SIAM Rev. 38: 49–95, 1996.
[14] Wieck, I. A. (2009). Explicit Symplectic Packings - Symplectic Tunnelling and New Maximal Constructions. Aachen: Shaker.