New Algorithms for k-Center and Extensions

René Brandenberg and Lucia Roth*

Zentrum Mathematik, Technische Universität München,
Boltzmannstr. 3, D–85747 Garching b. München, Germany
{brandenb,roth}@ma.tum.de

Abstract. The problem of interest is covering a given point set with homothetic copies of several convex containers $C_1, \ldots, C_k$, while the objective is to minimize the maximum over the dilatation factors. Such k-containment problems arise in various applications, e.g. in facility location, shape fitting, data classification or clustering. So far most attention has been paid to the special case of the Euclidean k-center problem, where all containers C_i are Euclidean unit balls. New developments based on so-called core-sets enable not only better theoretical bounds in the running time of approximation algorithms but also improvements in practically solvable input sizes. Here, we present some new geometric inequalities and a Mixed-Integer-Convex-Programming formulation. Both are used in a very effective branch-and-bound routine which not only improves on best known running times in the Euclidean case but also handles general and even different containers among the C_i.

Keywords: approximation algorithms, branch-and-bound, computational geometry, geometric inequalities, containment, core-sets, k-center, diameter partition, SOCP, 2-SAT.

1 Introduction

The issue of the following is the k-containment problem, that is covering a given point set with homothetic copies of several convex containers $C_1, \ldots, C_k$, while the objective is to minimize the maximum over the dilatation factors used in the covering. k-Containment problems arise in various applications, for instance in facility location, shape fitting, data classification or clustering (see [2], [18], [31], and [35] for several examples).

The k-center problem (the k-containment problem with identical containers) is known to be $\mathbb{NP}$-complete in general dimensions even when $k \geq 2$ and all containers are Euclidean unit balls (the *Euclidean k-center problem*) or $k \geq 3$ and all containers are l_∞ unit cubes [29]. Many approximation algorithms have been suggested for solving k-center problems (see [1] and the surveys [2], [31]). In many papers, the aim is improving complexity bounds and the presented algorithms are mostly of theoretical value. For practical purposes many purely

* Supported by the "Deutsche Forschungsgemeinschaft" through the graduate program "Angewandte Algorithmische Mathematik", Technische Universität München.

B. Yang, D.-Z. Du, and C.A. Wang (Eds.): COCOA 2008, LNCS 5165, pp. 64–78, 2008.

heuristic approaches exist (see e.g. [3], [19], [23], or [35]). Although they behave well for many inputs, they fail to provide provable guarantees.

So far most attention has been paid to the Euclidean k-center problem. Until recently it was believed that even in this case there is little hope to solve bigger instances, i.e. $n \geq 3$ or $k \geq 3$ (see e.g. [31]). Therefore, the planar Euclidean 2-center problem has been studied separately, for instance in [13], [14], [21], [24], and [32]. Recent progress is due to so-called core-sets [11], which gain a polynomial time approximation scheme (PTAS) for Euclidean k-center. However, the proposed full enumeration of all partitions of possible core-sets quickly causes non-computability in practice, even for moderate approximation errors. A first simple branch-and-bound (B&B) algorithm was suggested in [27].

Nevertheless, non-Euclidean containers are of practical interest, too. For instance in data analysis, the goal is in finding "similar" data points. Usually, there is no inherent reason why the 2-norm should be the better choice to express relations between data points than e.g. the 1- or ∞-norm. This is noteworthy as the polytopal norms often simplify calculations, e.g. in pattern recognition [28]. Special cases of rectilinear k-center problems have been addressed in [6] and [22]. In facility location (see e.g. Sect. 6) and shape fitting (see e.g. [9]) even non-symmetric and/or different container shapes may occur. Our algorithms allow both, general shapes and different C_i's within one instance (see Fig. 2 and 3 for examples).

Sections 2 and 3 address the basic definitions and a fundamental B&B procedure with good practical performance. In Sect. 4, a Mixed Integer Convex Programming formulation is given and its relaxation is used for further performance improvements. Especially if $k = 2$, further progress is achieved by diameter partitioning algorithms. These are described in Sect. 5, also including a couple of new geometric inequalities guaranteeing good bounds and a 2-SAT formulation used for the 2-containment problem with different containers. Both Sects. 2 and 3 are enhanced by some examples and experiments.[1] We stress that the new methods apply to a wider class of problems, therefore state them in full generality and provide an example in Sect. 6 indicating the use of the extension.

2 Problem Formulation

A container C is a full dimensional, convex, and compact subset of $\mathbb{R}^n$ with $0 \in \text{int}(C)$. For any container C and any $x \in \mathbb{R}^n$ let $\|x\|_C := \min_{\rho \geq 0}\{x \in \rho C\}$. Furthermore, for any point set $P \subset \mathbb{R}^n$ let $R(P, C) := \min_{c \in \mathbb{R}^n} \max_{p \in P} \|p - c\|_C$ and $d(P, C) := \max_{p,q \in P} R(\{p, q\}, C)$.

For any container $C \subset \mathbb{R}^n$, let s_C denote the *Minkowski symmetry* of C, that is the maximal dilatation factor ρ such that some translate of $-\rho C$ is contained in C, or for short $s_C = 1/R(-C, C)$. Obviously, $s_C \leq 1$, and we say that C is symmetric if and only if $s_C = 1$. In the latter case C can be translated such

[1] The experiments are restricted to exemplary tests with balls and cubes as containers, in order to allow valuation of the running times. Note that our results may be more important for other container shapes, where no specialized methods such as the core-set results apply.

that $C = -C$, i.e. C is 0-symmetric. Furthermore, $s_C \geq 1/n$ follows from John's theorem [25] (and can easily be shown directly).[2]

If C is 0-symmetric, $\|\cdot\|_C : \mathbb{R}^n \to \mathbb{R}$ denotes the Minkowski norm with unit ball C. In this case, $R(P, C)$ and $d(P, C)$ denote the outer radius and half-diameter of P with respect to $\|\cdot\|_C$. Furthermore, if P *and* C are symmetric, $R(P, C) = d(P, C)$ [16]. However, be aware that $\|x\|_C \neq \| - x\|_C$ for some $x \in \mathbb{R}^n$ if $C \neq -C$. Now, the problem of interest can be stated. Let $k \in \mathbb{N}$, and for $1 \leq i \leq k$, let $\mathcal{C}_i^n$ be families of n-dimensional containers and $\mathcal{P}^n$ the family of finite point sets in $\mathbb{R}^n$.

MINIMAL k-CONTAINMENT PROBLEM UNDER HOMOTHETICS(k-$\mathbf{MCP}_{Hom}^{\mathcal{P}}$)
Input: $n \in \mathbb{N}$, $m \in \mathbb{N}$, $P = \{p_1, \ldots, p_m\} \subset \mathcal{P}^n$, $C_1 \in \mathcal{C}_1^n, \ldots, C_k \in \mathcal{C}_k^n$.
Task: min ρ, s. th. $P = \{p_1, \ldots, p_m\} \subset \bigcup_{1 \leq i \leq k}(c_i + \rho C_i)$, $c_1, \ldots, c_k \in \mathbb{R}^n$.

The optimal value ρ is denoted by $R(P, C_1, \ldots, C_k)$. If $k = 1$, we get the minimal 1-containment problem under homothetics, which indeed computes the outer radius $R(P, C)$ of P with respect to the (non-symmetric) norm $\|\cdot\|_C$. When solving k-containment problems for general containers C, many and therefore fast computations of $R(P, C)$ and especially $R(\{p, q\}, C)$ with $\{p, q\} \subset P$ are needed. An overview on good solution or approximation techniques for different representations of the container C is given in [10]. k-$\text{MCP}_{Hom}^{\mathcal{P}}$ becomes the well known k-center problem when $C = C_1 = \ldots = C_k$:

k-CENTER PROBLEM
Input: $n \in \mathbb{N}$, $m \in \mathbb{N}$, $P = \{p_1, \ldots, p_m\} \in \mathcal{P}^n$, $C \in \mathcal{C}^n$.
Task: min ρ, s. th. $\forall j \in \{1, \ldots, m\}\ \exists i \in \{1, \ldots, k\} : \|p_j - c_i\|_C \leq \rho$, $c_1, \ldots, c_k \in \mathbb{R}^n$.

In this case the optimal radius ρ is denoted by $R^k(P, C)$.

3 A Core-Set Based Branch-and-Bound Scheme

In this section, we describe a basic core-set based B&B algorithm for k-$\text{MCP}_{Hom}^{\mathcal{P}}$.

3.1 Core-Sets

Let $S \subset P$ such that all points of S are assigned consistently with an optimal solution of the full k-$\text{MCP}_{Hom}^{\mathcal{P}}$ instance. For each of the k parts $S_i \subset S$, let c_i denote a center in an optimal solution of the corresponding 1-containment problem. Let $\rho = \max_i R(S_i, C_i)$. If for all $p \in P$ an index i exists such that $p \in c_i + (1 + \varepsilon)\rho C_i$, we have

$$\rho \leq R(P, C_1, \ldots, C_k) \leq (1 + \varepsilon)\rho$$

[2] Note that s_C of vertex- or facet-presented polytopes C can be computed via linear programming [17].

implying an ε-approximate solution of k-MCP$^{\mathcal{P}}_{Hom}$. Any such S is called an *ε-core-set* of P (with respect to $C_1, \ldots, C_k$).

In [11] it was shown that if all C_i are Euclidean, the sizes of the core-sets depend only on ε and neither on n nor m. Helly's theorem [20] implies the existence of core-sets whose size is independent of the number of points in P for all container shapes. However, dimension independence does not hold true for general (non-symmetric) containers [10].[3] Furthermore, one should note that in l_∞-spaces every diametrical pair of points is a 0-core-set (see Sect. 5.1), but that the algorithm as proposed in [11] may construct a core-set of size depending on n [10].

3.2 Branch-and-Bound Scheme

At each node in the B&B tree, we regard a core-set $S \subset P$ already partitioned into clusters S_i which have to be covered by homothetic copies of the containers C_i. For the branching, a point $p^* \in P \setminus S$ not (yet) covered is chosen and added to each of the sets S_i consecutively. We choose the point p maximizing $\min_i \|p - c_i\|_{C_i}$[4], or, in case the maximum is too expensive to compute, any point p with $\|p - c_i\|_{C_i}$ bigger than the current $(1+\varepsilon)\max_i \rho_i$. The remaining points play no further role in this step of the basic B&B procedure. (This will be improved in Sect. 4.)

For the branching, the clusters are sorted according to the distances $\|p - c_i\|_{C_i}$ and then p^* is assigned to the nearest cluster first. With this greedy-like strategy, good upper bounds are computed at an early stage of the algorithm, resulting in fast truncation of many branches and shorter overall running time. Solving the 1-center instances for each C_i and its assigned core-set points generates first lower bounds on the optimal value for the subtree below the current node.[5]

The algorithm returns an ε-core-set $S \subset P$ consisting of the points chosen at the nodes of an optimal branch, partitioned into k subsets $S_1, \ldots, S_k$, corresponding to the assignment of the points to the containers $C_1, \ldots, C_k$.

Algorithm 1[6]

initialize: set $S_i = \emptyset$, $\rho_i = 0$, c_i *arbitrarily for all* i,
  and $\bar{\rho}$ *to an upper bound for* $R(P, C_1, \ldots, C_k)$
k-containment(S_i, ρ_i, c_i):
  update the global upper bound $\bar{\rho}$
  compute $\delta = \max_{p \in P \setminus \bigcup S_i} \min_i(\|p - c_i\|_{C_i})$
  let p^* *the point where the maximum is attained*

[3] In case of general symmetric containers the existence of dimension independent ε-core-sets is open.

[4] In [27] p^* maximizes $\min_i(\|p - c_i\|_{C_i} - \rho_i)$, but our choice yields better results.

[5] It is recommendable to compute the $\|.\|_{C_i}$-distances between the new point and the points already assigned to S_i first to prevent unnecessary radius computations.

[6] The algorithm is written down recursively for better readability. However, to gain good running times, recursion in implementations should be avoided.

if $(1+\varepsilon)\max_i \rho_i \geq \delta$*: return*
else: sort cluster indices descending according to $\|p^* - c_i\|_{C_i}$
for $j = i_1, \ldots, i_k$*:*
recompute c_j *and* ρ_j *for* $S_j = S_j \cup p^*$
if $\max_i \rho_i \leq \bar{\rho}(1+\varepsilon)$*:*
k-containment(S_i, ρ_i, c_i)
return the best S_i, ρ_i, *and* c_i *found*

Testing the $(1+\varepsilon)$-containment condition at each node of the tree yields an approximation algorithm with a running time of $O(k^h nm)$, where h is the size of a maximal core-set constructed during the algorithm. It follows from [11] that for Euclidean k-center this B&B algorithm is a PTAS as $h = O(k/\varepsilon^2)$ in this case.

If an upper bound for the optimal radius is known, $\bar{\rho}$ can be initialized accordingly. Since the first k steps of Algorithm 1 (i.e. when each cluster contains exactly one point), match the first k steps of the greedy algorithm in [15] (assuming that the distance to an empty cluster is set to zero), in the *symmetric case* an approximation factor of at least 2 can be guaranteed at that stage.

According to [27], the implementation reported there is the first to practically solve huge k-center instances. The experiments in that paper show that the B&B algorithm used performs much better on practical data sets than the predicted worst case running times suggest. It is concluded that in dimensions 2 and 3, Euclidean k-center is practical for $\varepsilon \geq 0.01$ and $k \leq 4$, whereas computations in 3-space are significantly more expensive than in 2-space. The latter is caused in the fact, that though the upper bounds on core-set sizes are dimension independent, in practical computations the core-set sizes in lower dimensions are far from the upper bounds and grow noticeably with the dimension (and so do the running times of the B&B procedure). However, it is also reported that "some of the data sets [...] solved in 3D [...], ran for almost a week on an Intel Itanium system". Our implementation allows solving Euclidean k-center instances with bigger input sizes even in higher dimensions and for greater k values within some hours (at most) on an Intel Core 2 system[7]. Our realization of Algorithm 1 already substancially improves the running times as reported in [27] and further improvements are obtained by the methods presented in the following. In addition to that, our methods apply to *general* k-containment problems.

4 Convex Relaxation

In this section, a version of k-MCP$^{\mathcal{P}}_{Hom}$ with additional information is considered. It is assumed that the correct clusters are known for *some* of the points in P. This is a natural hypothesis in the context of a B&B scheme and enhances the chances to compute good upper and lower bounds for the optimal solution.

Especially good lower bounds are crucial for the performance of a B&B procedure. Whereas Algorithm 1 computes local lower bounds by determining the

[7] Both implementations use Matlab and comparable SOCP solvers.

radii of the current clusters, we now propose lower bounds taking both, assigned and unassigned points, into account. The new bounds are at least as good as the old ones, but usually much better.

4.1 A Mixed-Integer-Convex Program

Recall that the core-set $S = S_1 \cup \ldots \cup S_k$ denotes the assigned subset of P, i.e. $S_i \subset c_i + \rho C_i$ for some $c_i \in \mathbb{R}^n$ and $\rho > 0$, $i = 1, \ldots, k$. Now, let $S_0 \subset P \setminus S$ denote some of the unassigned points. Then the k-MCP$^{\mathcal{P}}_{Hom}$ with assigned points in $S_1, \ldots, S_k \neq \emptyset$ can be formulated as a mixed integer convex program with variables ρ, c_i and $\lambda_{ij} \in \{0, 1\}$. For this purpose for each $p_j \in S_0$ and each possible cluster S_i, a reference point $q_{ij} \in \text{conv}(S_i)$ is fixed (see Sect. 4.2 for strategies for choosing these points).

$$\begin{aligned} \min\ & \rho \\ \|p_j - c_i\|_{C_i} &\leq \rho \qquad \forall p_j \in S_i,\ i = 1, \ldots, k \\ \|\lambda_{ij} p_j - c_i + (1 - \lambda_{ij}) q_{ij}\|_{C_i} &\leq \rho \qquad \forall p_j \in S_0,\ \forall i = 1, \ldots, k \\ \sum_{i=1}^{k} \lambda_{ij} &= 1 \qquad \forall p_j \in S_0 \\ \lambda_{ij} &\in \{0, 1\}\ \forall p_j \in S_0,\ \forall i = 1, \ldots, k \end{aligned} \tag{1}$$

So, whenever $\lambda_{ij} = 0$ only the reference point q_{ij} has to be covered, a redundant condition. In contrast, p_j actually has to be contained in the homothetic copy of C_i if $\lambda_{ij} = 1$.

4.2 Relaxation

Relaxing the $\{0, 1\}$-condition on the multipliers λ_{ij} yields a convex program, providing a lower bound for $R(P, C_1, \ldots, C_k)$. A possible interpretation of the relaxation is including not the point p_j itself but a point on the line section between p_j and q_{ij} for all i, whereas the constraint $\sum_{i=1}^{k} \lambda_{ij} = 1$ enforces that not all of these points can be close to the reference points (see Fig. 1).

Picking q_{ij} such that the distance between q_{ij} and p_j is small gives the best bounds. However, the projection of p_j onto the convex hull of S_i causes elongation of overall computing time. Balancing between fast computations and a good choice of q_{ij}, the most successful strategy seems choosing q_{ij} as the point in S_i closest to p_j.

For polytopal C_i, the relaxation of (1) is a Linear Program; for Euclidean containers, we get a Second-Order-Cone Program (SOCP). Many other cases can be cast as SOCPs, too, for instance when the containers are intersections or Minkowski sums of Euclidean balls and polytopes (compare [10]).

Obviously, the more points from $P \setminus S$ belong to S_0, the better the lower bound on $R(P, C_1, \ldots, C_k)$ will be. However, as each $p \in S_0$ results in at least $k - 1$ additional variables and constraints, the relaxation of (1) is practical only when S_0 is not too big. Experiments show that even very small sets S_0 usually provide enough potential to reduce the number of nodes in the B&B tree significantly

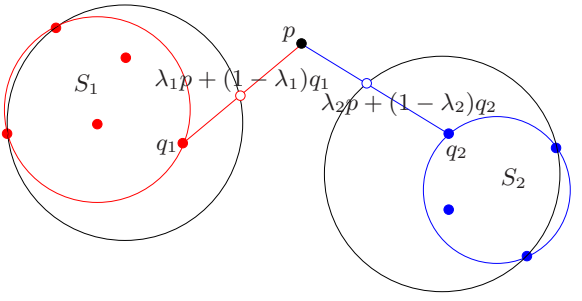

Fig. 1. The geometric meaning of the relaxed program. Optimal cluster radii with (black) and without (red resp. blue) considering the unassigned point p.

(compare Table 1, where 5 points have been chosen). There are different possible strategies to select points for S_0, e.g. randomly, maximizing the minimal distance to a current cluster, maximizing the distance to the latest core-set point, or maximizing the minimal distance to the unassigned points already chosen (which is what we do in Table 1). The solution of the convex program provides not only lower bounds. Upper bounds can easily be obtained by assigning the points $p_j \in P \setminus S$ to the clusters, e.g by $\min_i \|p_j - c_i\|_{C_i}$ or $\max_i \lambda_{ij}$ if $p \in S_0$.

The test results in Table 1 show that the MISOCP-relaxation significantly reduces the size of the B&B scheme for Euclidean k-center. Since solving the convex program at each node of the B&B tree is expensive, the improvement in the running time is still considerable but not as big as in the number of nodes. Further speedup should be possible by advanced strategies for the MISOCP-relaxation. In particular, we expect that improvements can be achieved through more elaborate techniques for determining the nodes at which to solve the convex program, the accuracy to which it should be solved, and the points in $|S_0|$. Moreover, practical solutions may be accelerated significantly by replacing the pure B&B algorithm by some kind of branch-and-cut routine.

5 Diameter Partitioning

Another possibility to improve the performance of Algorithm 1 is to consider $R(\{p,q\}, C_i)$ for all pairs of points $\{p,q\} \subset P$, and all $i = 1, \ldots, k$. The distances between point pairs provide information about optimal partitionings which can be used to compute bounds for $R(P, C_1, \ldots, C_k)$. The approach is useful especially when $k = 2$ and $R(\{p,q\}, C_i)$ can easily be computed. Surely, computing all pairwise distances is quadratic in the number of input points, so the approach is practical mainly for moderate point sets P.

5.1 Identical Containers

The information about the pairwise distances is captured in the ρ-distance graph:

Definition 1. *For every $\rho > 0$ we call the graph $G(\rho) = (P, E)$ with edges for every pair $\{p,q\}$ with $R(\{p,q\}, C) > \rho$ the* ρ-distance graph *of (P, C).*

Table 1. The B&B algorithm with and without SOCP bounds for Euclidean k-center and an approximation error of 0.01. The 3D geometric model data sets are comparable to the ones used in [27]. The 5D "rand. box" data sets refer to equally distributed points within boxes with randomly scaled axes. (We assume that this is more appropriate for k-center problems than, e.g., equally distributed points within the unit cube.) Sizes of the B&B tree and running times (in seconds) are listed – in case of the random data sets, the mean over samples of 20. We use a 2.0 GHz Intel Core 2 system running Matlab R2006B and SeDuMi [30], [33]. The code for Euclidean distance computations is provided by [12].

				pure B&B			B&B with relaxed MISOCP		
data set	m	n	k	nodes	leaves	time	nodes	leaves	time
cat	352	3	4	10353	2138	505.6	2380	144	207.7
shark	1744	3	4	649	126	26.1	225	27	13.6
seashell	18033	3	4	12718	2365	925.6	3266	479	371.0
dragon	437645	3	3	341	96	154.9	161	43	89.2
rand. box	1000	5	3	889.3	57.8	44.6	623.9	35.7	45.6
rand. box	1000	5	4	20919.9	3249.8	1272.6	6544.0	238.4	611.2
rand. box	10000	5	3	2595.1	167.6	166.6	1577.7	84.3	139.0
rand. box	10000	5	4	32611.9	3021.6	2273.7	13768.3	808.1	1459.4

The next algorithm computes the maximal ρ such that $G(\rho)$ is k-colorable. Finding a k-coloring of $G(\rho)$ corresponds to partitioning the point set P into k subsets, where no pair of points with $R(\{p,q\},C) > \rho$ lies within one set.

Algorithm 2

for all l pairs $\{p,q\}$ of points in P:
  compute $\rho_j = R(\{p,q\},C)$, $1 \le j \le l$
label such that $\rho_1 \ge \ldots \ge \rho_l$
for $j = 1,\ldots,l$:
  if $G(\rho_j)$ is not k-colorable
    break
  set $\rho = \rho_j$
return ρ

Deciding whether a graph is k-colorable is itself a hard problem if $k \ge 3$ and Algorithm 2 may not be polynomial. Still, good bounds may be obtained from heuristic coloring algorithms.

If $k = 2$, Algorithm 2 can be implemented by maintaining a 2-coloring of $G(\rho)$ while successively inserting new edges. One should note that since $G(\rho)$ may be not connected, more than two labels (or colors) may be necessary. When an edge is inserted which is not connected to the subgraph already built, a new pair of labels is created. When an edge joins two previously disconnected components, the relevant labels are merged.

Depending on the shape of the container, different approximation qualities for the underlying k-center problem can be guaranteed.

Parallelotopes. If (and only if) C is a parallelotope (e.g. a unit cube, if $\|\cdot\|_C = \|\cdot\|_\infty$) the Helly dimension of C is 1; that is, $R(P,C) = d(P,C)$ for all P [8, 14.3]. This implies that P can be packed into k translates of ρC if and only if $G(\rho)$ is k-colorable [4], [29]. Hence, Algorithm 2 solves the k-center problem for parallelotopes[8] exactly.

Note that solving the 2-center problem in l_∞ via diameter partitioning is not optimal. A faster algorithm is proposed in [5]. It computes a minimal axis-parallel enclosing box for P and determines the position of the two cubes in this box by maximizing consecutively in the directions of the n coordinate axes. However, Algorithm 2 has the advantage of being adaptable to *general* 2-containment problems, whereas the algorithm in [5] is limited to two *identical* parallelotopal containers.

Euclidean Containers. We get the following for Euclidean containers:

Lemma 1. *Algorithm 2 computes a* $\sqrt{\frac{2n}{n+1}}$*-approximation of* $R^k(P,C)$ *for any point set* P *and any ellipsoid* C*.*

Proof. Surely, $d(P_i,C) \le R^k(P,C)$ for all P_i when $P_1,\dots,P_k$ is a partition of P such that every two points joint by an edge in the final distance-graph of Algorithm 2 are in different P_i. If $P_1^*,\dots,P_k^*$ is an optimal partition, $\max_i R(P_i,C) \ge \max_i R(P_i^*,C) = R^k(P,C)$. Hence, by Jung's inequality [26],

$$\max_i d(P_i,C) \le R^k(P,C) \le \max_i R(P_i,C) \le \max_i \sqrt{\frac{2n}{n+1}}\, d(P_i,C).$$

In computations, an incomplete partition can be extended in a greedy manner upon all points in P. Besides the lower bound output ρ of Algorithm 2, an upper bound $\bar\rho = \max_i R(P_i,C)$ is obtained. Surely, this upper bound is often much smaller than $\sqrt{2n/(n+1)}\rho$ in practice (compare Table 2).

General, Identical Containers. For general containers C, the bounds are weaker, but only slightly when C is (almost) symmetric.

Lemma 2. *Algorithm 2 computes an* $\frac{n}{n+1}(1+\frac{1}{s_C})$*-approximation of the optimal radius* $R^k(P,C)$ *for any point set* $P \subset \mathbb{R}^n$ *and any container* $C \subset \mathbb{R}^n$*.*

Proof. Following the proof of Lemma 1 it suffices to show that $R(P,C) \le \frac{n}{n+1}(1+\frac{1}{s_C})d(P,C)$ for any point set P. Suppose $d(P,C) = 1$, i.e. every two points in P can be covered by a translate of C. It easily follows that every two points of $P-P$ can be covered by $C-C$, and since both $P-P$ and $C-C$ are symmetric $R(P-P,C-C) = d(P-P,C-C) = 1$ [16]. Since $(1+s_P)\,P$ can be covered by a translate of $P-P$ and $C-C$ by a translate of $(1+\frac{1}{s_C})C$, we conclude with $s_P \ge \frac{1}{n}$ that P is contained in a translate of $\frac{n}{n+1}(1+\frac{1}{s_C})C$.

[8] When the parallelotope is given in $\mathcal{H}$-representation $C = \bigcap_i \{x : a_i^T x \le 1\}$, and especially for l_∞-containment, $R(\{p,q\},C) = \max_i a_i^T(p-q)$ can easily be computed.

Remark 1. a) If C is symmetric, a well known inequality about the ratio between the outer radius and the diameter of convex sets (or point sets) in arbitrary Minkowski spaces [7] can be obtained as a corollary of Lemma 2:

$$\frac{R(P,C)}{d(P,C)} \leq \frac{2n}{n+1}.$$

b) If $\mathcal{C}^n$ is a small subset of the set of convex bodies in $\mathbb{R}^n$, like the parallelotopes or ellipsoids (or – maybe even non-symmetric – sets close to these shapes) in Sects. 5.1 and 5.1, the approximation error may be much better than predicted by Lemma 2.

c) If a better guarantee on lower bounds on the Minkowski symmetry of the input point set P can be given, the bounds in Lemma 2 can be improved.

5.2 Different Containers

Regarding the general k-MCP$^{\mathcal{P}}_{Hom}$, two points p and q which are far apart in the (non-symmetric) norm induced by one container may be close in the norm induced by another. Definition 1 has to be adapted.

Definition 2. *Let $G = (V, E_1, \ldots, E_k)$ be an (edge-colored) multigraph with vertex set V and edge sets $E_1, \ldots, E_k$. A* generalized k-coloring *of G is a partition $V_1, \ldots, V_k$ of the vertices V such that for any $\{v, w\} \in E_i$ it follows $\{v, w\} \not\subset V_i$, $i = 1, \ldots, k$.*

Again, we can define the ρ-distance graph:

Definition 3. *For every $\rho > 0$ the* ρ-distance graph *of $(P, C_1, \ldots, C_k)$ is the edge-colored multigraph $G(\rho) = (P, E_1, \ldots, E_k)$ with edges in E_i for every pair $\{p, q\}$ with $R(\{p, q\}, C_i) > \rho$.*

Now a solution of the generalized k-coloring problem for the ρ-distance graph $G(\rho)$ implies again that ρ is a lower bound for $R(P, C_1, \ldots, C_k)$.

Algorithm 3

for all l combinations of pairs $\{p, q\}$ of points in P and $i \in \{1, \ldots, k\}$:
  compute $\rho_j = R(\{p, q\}, C_i)$, $1 \leq j \leq l$
label such that $\rho_1 \geq \ldots \geq \rho_l$
for $j = 1, \ldots, l$:
  if $G(\rho_j)$ has no valid generalized k-coloring
    break
  set $\rho = \rho_j$
return ρ

Respecting the edge colors seems to make generalized k-coloring more difficult than usual coloring. Yet, if $k = 2$, the problem can still be solved efficiently:

Lemma 3. *The generalized 2-coloring problem can be reduced to 2-*SAT.

Proof. By assigning boolean variables z_i, where $z_i \Leftrightarrow (v_i \in V_i)$, the generalized 2-coloring instance (V, E_1, E_2) is equivalent to the following instance of 2-SAT:

$$\bigwedge_{\substack{(p_i,p_j)\\ E_1\text{-edges}}} (\neg z_i \vee \neg z_j) \quad \wedge \quad \bigwedge_{\substack{(p_i,p_j)\\ E_2\text{-edges}}} (z_i \vee z_j).$$

A valid assignment of a 2-SAT instance (or evidence that no valid assignment exists) can be found in linear time (in the size of $G(\rho)$), see, e.g., [34].

Any valid assignment of the variables z_i in the corresponding 2-SAT formula yields a partition into two sets P_1 and P_2 with the following property: $R(\{p, q\}, C_i) \leq \rho$, $i = 1, 2$ for any pair of points $p, q \in S_i$.

Lemma 4. *Algorithm 3 computes*

a) an $\frac{n}{n+1}(\max_{1\leq i\leq k}(\frac{1}{s_{C_i}}) + 1)$*-approximation for the general* k*-MCP*$^{\mathcal{P}}_{Hom}$*.*
b) a $\frac{2n}{n+1}$*-approximation for* k*-MCP*$^{\mathcal{P}}_{Hom}$ *if all containers are 0-symmetric.*
c) a $\sqrt{\frac{2n}{n+1}}$*-approximation for* k*-MCP*$^{\mathcal{P}}_{Hom}$ *if all containers are ellipsoids or parallelotopes.*
d) an exact solution of k*-MCP*$^{\mathcal{P}}_{Hom}$ *if all containers are parallelotopes (compare Fig. 2).*

Proof. This follows directly from Lemma 3 and Sect. 5.1.

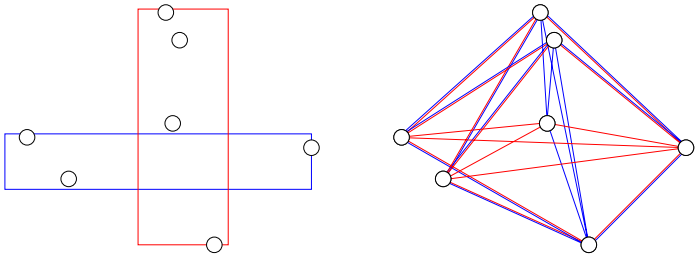

Fig. 2. An example of an optimal containment with two boxes as containers and the corresponding edges in the final ρ-distance graph. For parallelotopal containers, Algorithm 3 computes the exact solution.

5.3 Partitioning Procedures

Algorithms 2 and 3 approximate the 2-containment problem within the bounds given in Lemmas 1, 2, and 4.[9] For better approximations, we rely on the B&B procedure. If $|P|$ is not too big, the super-quadratic[10] running time of the diameter partitioning is not too expensive and it is even possible to combine Algorithms 1 and 2 (resp. 3) to compute an (almost) exact solution of the underlying 2-center problem (compare Table 2).

[9] E.g., the error is at most 0.225 if both C_i are parallelotopes or ellipsoids and $d \leq 3$.
[10] Since the edges have to be sorted.

Table 2. Test results for diameter partitioning where the containers are either two Euclidean balls or two arbitrarily, independently rotated unit cubes. The "rand. box" data sets refer to 100 equally distributed points within boxes with randomly scaled axes. The "norm. dist." data sets refer to 100 $(0, 1)$-normally distributed points. Due to the page limit, we report only the mean running times (in seconds) and the approximation quality after the diameter partitioning step (DP) over samples of 20 here. The accuracy is $\varepsilon = 0.01$ for all tests. See Table 1 for details on the environment used.

			pure B&B	B&B with diameter partitioning			
containers	data set	n	time	DP error	DP time	B&B time	overall time
Euclidean	rand. box	10	7.2	0.06	0.5	2.6	3.1
Euclidean	rand. box	20	26.2	0.11	0.5	14.1	14.6
Euclidean	rand. box	30	88.3	0.15	0.8	67.3	68.2
Euclidean	norm. dist.	10	10.9	0.12	0.5	7.1	7.6
Euclidean	norm. dist.	20	56.1	0.15	0.6	32.5	33.1
Euclidean	norm. dist.	30	135.5	0.17	0.8	89.0	89.9
rot. cubes	rand. box	10	12.8	$< \varepsilon$	2.1	-	2.1
rot. cubes	rand. box	20	103.2	$< \varepsilon$	2.4	-	2.4
rot. cubes	rand. box	30	639.9	$< \varepsilon$	3.1	-	3.1
rot. cubes	norm. dist.	10	21.5	$< \varepsilon$	1.1	-	1.1
rot. cubes	norm. dist.	20	210.9	$< \varepsilon$	2.1	-	2.1
rot. cubes	norm. dist.	30	1153.1	$< \varepsilon$	2.7	-	2.7

Combining the two algorithms is accomplished as follows. First, consider identical containers. A good upper bound obtained by Algorithm 2 decreases the running time as many branches need not be considered. Secondly, it provides a-priori information about point pairs not fitting in the same container: If $R(\{p, q\}, C) > \bar{\rho}$ for two points in P, assigning one of them to P_1 forces the other one into P_2. Since all pairwise distances have been computed and sorted, all such pairs of points can easily be identified and assigned to different partition sets. This is equivalent to building the distance graph $G(\bar{\rho})$ and 2-coloring it. As all possible 2-colorings have to be considered and the resulting bipartite graph is not connected in general, this leads to (usually several) disjoint subset pairs of P. During the B&B routine, each of those subset pairs can be considered as a whole and requires only one node in the B&B tree. For instance, when we choose such a point as first core-set point, we can assign all the points from the corresponding pair of colored subsets to the right cluster – even before the branching has started. The same can be done for 2-containment problems with different containers using Algorithm 3. Yet, here, each color yields a distinct set of subset pairs which has to be taken into account in the B&B procedure.

As one can conclude from the experiments in Table 2, Euclidean 2-center problems can be approximated to a good level of accuracy even in higher dimensions. One should especially recognize the quality of the bounds computed by the diameter partitioning before starting the B&B. Of course, the approximation quality achieved by the diameter partitioning is even better for some classes of non-Euclidean containers where nothing is known about the existence of small core-sets or even fast algorithms to compute those. This becomes clear when

looking at the results in Table 2 for applying an implementation of Algorithm 3 on 2-containment problems with rotated cubes.

Algorithms 2 and 3 get slow on large point sets. However, it is not necessary to abandon their advantages. We restrict the algorithms to small subsets (e.g. random samples) of the input data, perform the diameter partitioning, compute an upper bound $\bar{\rho}$ (for the complete point set) and apply the pre-partitioning only to the sample. Any time the B&B algorithm picks a new core-set point, we test whether this point supplies additional information and if applicable add more points from the sample to the core-set. Surely, there is no guarantee for the quality of the computed bounds. But even this simple strategy improves the running times in experiments. Further reductions should be possible by more advanced strategies to avoid the evaluation of the complete graph over P.

6 An Application of Non-euclidean Container Shapes

In the case of non-Euclidean containers, a typical setting for instance in facility location is covering a 2D (point) set with several objects. However, different from the problems addressed before, rotations of the containers in addition to homothetics are of interest.

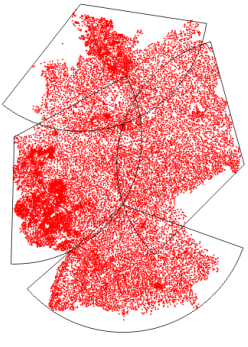

Fig. 3. Solution of a 4-containment problem with 18512 data points allowing rotations of the containers (accuracy 2%)

Figure 3 depicts the solution of such a 4-containment problem with identical 2-dimensional containers being conical sections of circles. These 'pie slice' shapes arise in applications when points should be within the sight of cameras, in the transmission range of oriented senders, or reachable by robot arms with joint limits [18]. A discretization of the possible space of rotations is considered, and included in the B&B algorithm. Note that the computational effort increases severely since the rotations of the four containers have to be addressed independently of each other. Still, the full computation takes less than 5 hours.

References

1. Agarwal, P.K., Procopiuc, C.M.: Exact and approximation algorithms for clustering. In: Proc. 9th ACM-SIAM Symp. Discrete Alg., pp. 658–667 (1998)
2. Agarwal, P.K., Sharir, M.: Efficient algorithms for geometric optimization. ACM Comput. Surv. 30(4), 412–458 (1998)
3. Anderberg, M.R.: Cluster analysis for applications. Probability and mathematical statistics. Academic Press, London (1973)
4. Avis, D.: Diameter partitioning. Discrete Comput. Geom. 1, 265–276 (1986)
5. Bespamyatnikh, S., Kirkpatrick, D.: Rectilinear 2-center problems. In: Proc. 11th Canad. Conf. Comp. Geom., pp. 68–71 (1999)
6. Bespamyatnikh, S., Segal, M.: Covering a set of points by two axis-parallel boxes. Inf. Process. Lett. 75(3), 95–100 (2000)
7. Bohnenblust, H.F.: Convex regions and projections in Minkowski spaces. Ann. Math. 39, 301–308 (1938)
8. Boltyanski, V., Martini, H., Soltan, P.S.: Excursions into Combinatorial Geometry. Springer, Heidelberg (1997)
9. Brandenberg, R., Gerken, T., Gritzmann, P., Roth, L.: Modeling and optimization of correction measures for human extremities. In: Jäger, W., Krebs, H.-J. (eds.) Mathematics – Key Technology for the Future. Joint Projects between Universities and Industry 2004-2007, pp. 131–148. Springer, Heidelberg (2008)
10. Brandenberg, R., Roth, L.: Optimal containment under homothetics, a practical approach (submitted, 2007)
11. Bădoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: Proc. 34th Annu. ACM Symp. Theor. Comput., pp. 250–257 (2002)
12. Bunschoten, R.: A fully vectorized function that computes the Euclidean distance matrix between two sets of vectors (1999), `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=71`
13. Chan, T.M.: More planar two-center algorithms. Comp. Geom. Theor. Appl. 13(3), 189–198 (1999)
14. Eppstein, D.: Faster construction of planar two-centers. In: Proc. 8th ACM-SIAM Symp. Discrete Alg., pp. 131–138 (1997)
15. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theor. Comput. Sci. 38, 293–306 (1985)
16. Gritzmann, P., Klee, V.: Inner and outer j-radii of convex bodies in finite-dimensional normed spaces. Discrete Comput. Geom. 7, 255–280 (1992)
17. Gritzmann, P., Klee, V.: On the complexity of some basic problems in computational convexity I: Containment problems. Discrete Math. 136, 129–174 (1994)
18. Halperin, D., Sharir, M., Goldberg, K.: The 2-center problem with obstacles. J. Alg. 42(1), 109–134 (2002)
19. Hartigan, J.A.: Clustering algorithms. Wiley series in probability and mathematical statistics. John Wiley and Sons, New York (1975)
20. Helly, E.: Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. Jahresbericht Deutsch. Math. Verein 32, 175–176 (1923)
21. Hershberger, J.: A faster algorithm for the two-center decision problem. Inf. Process. Lett. 47(1), 23–29 (1993)
22. Hoffmann, M.: A simple linear algorithm for computing rectilinear 3-centers. Comput. Geom. Theor. Appl. 31(3), 150–165 (2005)
23. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice Hall, Englewood Cliffs (1988)

24. Jaromczyk, J.W., Kowaluk, M.: An efficient algorithm for the Euclidean two-center problem. In: Symp. Comp. Geom, pp. 303–311 (1994)
25. John, F.: Extremum problems with inequalities as subsidiary conditions. In: Courant Anniversary Volume, pp. 187–204. Interscience (1948)
26. Jung, H.W.E.: Über die kleinste Kugel, die eine räumliche Figur einschließt. J. Reine Angew. Math. 123, 241–257 (1901)
27. Kumar, P.: Clustering and reconstructing large data sets. PhD thesis, Department of Computer Science, Stony Brook University (2004)
28. Mangasarian, O.L., Setiono, R., Wolberg, W.H.: Pattern recognition via linear programming: theory and application to medical diagnosis. In: Coleman, T.F., Li, Y. (eds.) Large-Scale Numerical Optimization, pp. 22–31. SIAM, Philadelphia (1990); Computer Sciences TR 878 (1989)
29. Megiddo, N.: On the complexity of some geometric problems in unbounded dimension. J. Symb. Comput. 10(3/4), 327–334 (1990)
30. Pólik, I.: Addendum to the sedumi user guide version 1.1. Technical report, Advanced Optimization Laboratory, McMaster University (2005)
31. Procopiuc, C.M.: Clustering problems and their applications: A survey. Department of Computer Science, Duke University (1997)
32. Sharir, M.: A near-linear algorithm for the planar 2-center problem. In: Proc. Symp. Comp. Geom., pp. 106–112 (1996)
33. Sturm, J.F.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optim. Method. Softw. 11-12, 625–653 (1999)
34. del Val, A.: On 2-SAT and renamable Horn. In: Proc. 17th Nat. Conf. on Artif. Intel. AAAI / MIT Press (2000)
35. Wei, H., Murray, A.T., Xiao, N.: Solving the continuous space p-centre problem: planning application issues. IMA J. Management Math. 17, 413–425 (2006)