

---

# Mathematical programming formulations for the bottleneck Hyperplane Clustering Problem

Kanika Dhyani<sup>1,2</sup> and Leo Liberti<sup>1</sup>

<sup>1</sup> LIX, École Polytechnique, 91128 Palaiseau, France  
{dhyani, liberti}@lix.polytechnique.fr

<sup>2</sup> DEI, Politecnico Di Milano, P.zza L. Da Vinci 32, 20133 Milano, Italy  
dhyani@elet.polimi.it

**Summary.** We discuss a mixed-integer nonlinear programming formulation for the problem of covering a set of points with a given number of slabs of minimum width, known as the bottleneck variant of the hyperplane clustering problem. We derive several linear approximations, which we solve using a standard mixed-integer linear programming solver. A computational comparison of the performance of the different linearizations is provided.

**Key words:** MINLP,  $k$ -line center problem, reformulation, linearization.

## 1 Introduction

We investigate some mathematical programming formulations for the following optimization problem.

**BOTTLENECK HYPERPLANE CLUSTERING PROBLEM (bHCP).** Given integers  $n, m, d > 0$  and a set  $N = \{\mathbf{p}_i \in \mathbb{R}^d \mid i \leq n\}$ , find a set  $M = \{(\mathbf{w}_j, w_j^0) \in \mathbb{R}^d \times \mathbb{R} \mid j \leq m\}$  and an assignment  $x : N \times M \rightarrow \{0, 1\}$  of points to hyperplanes such that  $\max_{\substack{i \leq n, j \leq m \\ x_{ij}=1}} \frac{|\mathbf{w}_j \cdot \mathbf{p}_i - w_j^0|}{\|\mathbf{w}_j\|_2}$  is minimum.

In other words, we want to partition  $N$  into  $m$  clusters whose points are projected onto a  $(d - 1)$ -dimensional subspace in such a way that the maximum Euclidean distance between a point and its projection is minimized. Our problem is a special case of a projective clustering problem in which all the subspaces are of the same dimension. It is also known as the **HYPERPLANE COVER PROBLEM** [9], the  **$m$ -HYPERPLANE CENTER PROBLEM** [15] and the **SLAB WIDTH PROBLEM** [7] in literature.

If we fix  $\max_{\substack{i \leq n, j \leq m \\ x_{ij}=1}} \frac{|\mathbf{w}_j \cdot \mathbf{p}_i - w_j^0|}{\|\mathbf{w}_j\|_2}$  to some maximum tolerance, geometrically our problem is that of finding slabs of minimum width that cover all the points —

thus the name bottleneck. In the case when the slabs are of zero width, the problem at hand is known as the  $k$ -line center problem in which lines are used instead of slabs. The  $k$ -line center problem has been studied extensively in literature: [15] reports a table with summarized complexities of the developed algorithms. Most of the past work studied the problem from a computational geometry point of view; most results are theoretical in nature. To the best of our knowledge, mathematical programming based solution approaches for the bHCP have not been extensively studied yet.

### 1.1 Previous work

Clustering techniques are widely studied in areas ranging from data mining to information retrieval and pattern recognition to name a few. They also arise in the context of shape fitting in geometric covering problems where given a set of shapes the goal is to choose the one that covers the data points w.r.t. some objective. Deciding whether a set of  $n$  points in  $\mathbb{R}^2$  can be covered with  $m$  lines was shown to be **NP**-complete in [14]; trying to approximate the width of the minimum slab that covers the point is also **NP**-complete.

A sublinear time randomized algorithm in which all but  $(1 - \gamma)n$  of the points are covered is presented in [15]. They prove that the  $m$ - $q$ -dimensional hyperplane center problem (where  $q$  is the dimension of the subspace) can be solved in  $\tilde{O}(d \frac{mq^{q+1}}{\gamma})$ . The described algorithm finds a collection of  $\mathcal{O}(m \log \frac{mdq}{\gamma})$  slabs of width at most  $2^q$  times the optimum.

For points in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , [2] present randomized algorithms which compute  $\mathcal{O}(m \log m)$  strips of bounded width that cover the data points. Their algorithms have run times of  $\mathcal{O}(nm^2 \log^4 n)$  if  $m^2 \log m \leq n$  and  $\mathcal{O}(n^{2/3} m^{8/3} \log^4 n)$  for larger  $m$  when  $d = 2$  and  $\mathcal{O}(n^{3/2} m^{9/4} \text{polylog}(n))$  for  $d = 3$ .

A coresets framework based approach was proposed in [7]. It was shown that no coresets exist for the problem of covering a set of points with 2 slabs in  $\mathbb{R}^3$ ; however, a  $(1 + \gamma)$  approximation algorithm for the same problem was also presented.

In [9], some fixed-parameter tractable algorithms which are based on general techniques of parameterized complexity are presented. The main argument of this work rests on the fact that certain parameters (such as the dimension) can be used to limit the complexity of these problems.

Many variants of the bHCP have been proposed in the literature. [5] adapted the  $k$ -means algorithm to the case of hyperplanes and treated the problem of minimizing the sum of the Euclidean distances of points to the assigned hyperplanes (for fixed  $m$ ). Another variant minimizes the total number of hyperslabs of a fixed width used to cover a set of points [3].

The rest of this paper is organized as follows. The mathematical programming formulation of the bHCP is given in Sect. 2. Some exact reformulations are given in Sect. 3. Three model-based approximating linearizations are proposed in Sect. 4. Our computational results are discussed in Sect. 5. Sect. 6 concludes the paper.

## 2 Problem formulation

Given a set of  $n$  points  $\mathbf{p}_i \in \mathbb{R}^d$  ( $i \leq n$ ) we seek a set of  $m$  hyperplanes  $(\mathbf{w}_j, w_j^0) \in \mathbb{R}^{d+1}$  ( $j \leq m$ ) and an assignment  $\mathbf{x} \in \{0, 1\}^{nm}$  (such that  $x_{ij} = 1$  iff  $\mathbf{p}_i$  is assigned to the hyperplane  $(\mathbf{w}_j, w_j^0)$  for all  $i \leq n, j \leq m$ ) that minimizes the maximum of the Euclidean distances between the hyperplanes and their assigned points. The following Mixed-Integer Nonlinear Programming (MINLP) formulation correctly describes the problem:

$$\left. \begin{aligned} \min \max_{\substack{j \leq m \\ i \leq n}} \frac{|\mathbf{w}_j \mathbf{p}_i - w_j^0|}{\|\mathbf{w}_j\|_2} x_{ij} \\ \text{s.t. } \forall i \leq n \quad \sum_{j \leq m} x_{ij} = 1 \\ \mathbf{w} \in \mathbb{R}^{md} \\ \mathbf{x} \in \{0, 1\}^{nm}. \end{aligned} \right\} \quad (1)$$

Computationally, the direct solution of (1) is problematic because (1) is non-differentiable and has a singularity at  $\mathbf{w} = 0$ .

## 3 Reformulations

Reformulations are symbolic transformations that are applied to the problem formulation and yield modified formulations with different mathematical properties [11]. Within this paper, we shall make use of two types of reformulations: opt-reformulations and approximations. *Opt-reformulations* guarantee that there is an optimum of the reformulated problem corresponding to each optimum of the original problem [12]. The precise definition of an approximation is given in [12]; for our purposes, it suffices to know that an approximating reformulation yields a sequence of reformulated problems dependent on a numerical parameter, which “tends” to the original problem when the parameter tends to infinity. An approximation is simply a problem in the sequence.

We first provide opt-reformulations that yield a differentiable MINLP. We remark that if we require all vectors  $\mathbf{w}$  to be normalized to 1, there is no need for dividing the objective function terms through by  $\|\mathbf{w}\|_2$ : the objective thus becomes

$$\min \max_{\substack{j \leq m \\ i \leq n}} |\mathbf{w}_j \mathbf{p}_i - w_j^0| x_{ij}$$

subject to added constraints

$$\forall j \leq m \quad \|\mathbf{w}_j\|_2^2 = 1. \quad (2)$$

We reformulate the maximum operator by introducing an added nonnegative continuous variable  $\varepsilon \geq 0$ : the objective becomes

$$\min \varepsilon$$

subject to added constraints

$$\forall i \leq n, j \leq m \quad \varepsilon \geq |\mathbf{w}_j \mathbf{p}_i - w_j^0| x_{ij}.$$

Secondly, we reformulate the absolute values by introducing added nonnegative continuous variables  $t_{ij}^+, t_{ij}^- \geq 0$ , which yield reformulated constraints

$$\forall i \leq n, j \leq m \quad \varepsilon \geq (t_{ij}^+ + t_{ij}^-) x_{ij}, \quad (3)$$

subject to added constraints

$$\forall i \leq n, j \leq m \quad \mathbf{w}_j \mathbf{p}_i - w_j^0 = t_{ij}^+ - t_{ij}^-.$$

This reformulation is exact as long as a complementarity constraint  $\sum_{i,j} t_{ij}^+ t_{ij}^- = 0$  is enforced; in this particular case, however, it is not necessary because of the minimization direction of the objective function. Lastly, since the products  $t_{ij}^+ x_{ij}$  and  $t_{ij}^- x_{ij}$  involve a binary variable, they can be linearized exactly by replacing them with added nonnegative continuous variables  $y_{ij}^+, y_{ij}^-$  whilst adding the following (linear) constraints:

$$\begin{aligned} \forall i \leq n, j \leq m \quad & y_{ij}^+ \leq \min(Mx_{ij}, t_{ij}^+) \\ \forall i \leq n, j \leq m \quad & y_{ij}^+ \geq t_{ij}^+ - M(1 - x_{ij}) \\ \forall i \leq n, j \leq m \quad & y_{ij}^- \leq \min(Mx_{ij}, t_{ij}^-) \\ \forall i \leq n, j \leq m \quad & y_{ij}^- \geq t_{ij}^- - M(1 - x_{ij}), \end{aligned}$$

where  $M$  is a large enough constant. We also remark that the intuitive meaning of (3) is that  $\varepsilon$  should be bounded below by  $t_{ij}^+ + t_{ij}^-$  if and only if  $x_{ij} = 1$ . This can be written formally as:

$$\forall i \leq n, j \leq m \quad y_{ij} \geq t_{ij}^+ + t_{ij}^-,$$

where  $y_{ij}$  is an added nonnegative continuous variable constrained to take value  $\varepsilon$  if and only if  $x_{ij} = 1$  (otherwise, it is free):

$$\begin{aligned} \forall i \leq n, j \leq m \quad & y_{ij} \leq \varepsilon + M(1 - x_{ij}) \\ \forall i \leq n, j \leq m \quad & y_{ij} \geq \varepsilon - M(1 - x_{ij}). \end{aligned}$$

The latter approach provides an alternative linearization of the products.

The reformulations above therefore provide two different twice-differentiable MINLP formulations for the bHCP:

$$\left. \begin{array}{l} \min \\ \text{s.t. } \forall i \leq n, j \leq m \\ \quad \forall j \leq m \\ \forall i \leq n, j \leq m \\ \forall i \leq n, j \leq m \\ \forall i \leq n, j \leq m \\ \forall i \leq n, j \leq m \\ \forall i \leq n, j \leq m \\ \quad \forall i \leq n \end{array} \right\} \left. \begin{array}{l} \varepsilon \\ \varepsilon \geq y_{ij}^+ + y_{ij}^- \\ \|\mathbf{w}_j\|_2^2 = 1 \\ \mathbf{w}_j \mathbf{p}_i - w_j^0 = t_{ij}^+ - t_{ij}^- \\ y_{ij}^+ \leq \min(Mx_{ij}, t_{ij}^+) \\ y_{ij}^+ \geq t_{ij}^+ - M(1 - x_{ij}) \\ y_{ij}^- \leq \min(Mx_{ij}, t_{ij}^-) \\ y_{ij}^- \geq t_{ij}^- - M(1 - x_{ij}) \\ \sum_{j \leq m} x_{ij} = 1 \\ \mathbf{w} \in \mathbb{R}^{md} \\ \mathbf{x} \in \{0, 1\}^{nm} \\ \mathbf{y}^+, \mathbf{y}^-, \mathbf{t}^+, \mathbf{t}^- \in [0, M]^{nm} \end{array} \right\} \quad (4)$$

$$\left. \begin{array}{l}
\min \quad \varepsilon \\
\text{s.t. } \forall i \leq n, j \leq m \quad y_{ij} \geq t_{ij}^+ + t_{ij}^- \\
\quad \forall j \leq m \quad \|\mathbf{w}_j\|_2^2 = 1 \\
\quad \forall i \leq n, j \leq m \quad \mathbf{w}_j \mathbf{p}_i - w_j^0 = t_{ij}^+ - t_{ij}^- \\
\quad \forall i \leq n, j \leq m \quad y_{ij} \leq \varepsilon + M(1 - x_{ij}) \\
\quad \forall i \leq n, j \leq m \quad y_{ij} \geq \varepsilon - M(1 - x_{ij}) \\
\quad \quad \sum_{j \leq m} x_{ij} = 1 \\
\quad \quad \mathbf{w} \in \mathbb{R}^{md} \\
\quad \quad \mathbf{x} \in \{0, 1\}^{nm} \\
\quad \quad \mathbf{y}, \mathbf{t}^+, \mathbf{t}^- \in [0, M]^{nm}
\end{array} \right\} \quad (5)$$

**Proposition 1.** *If  $(\mathbf{w}^*, \mathbf{x}^*)$  is a global optimum of (4) (resp. (5)) then it is also a global optimum of (1).*

*Proof.* This follows by Defn. 2.3.10 and Lemma 2.3.11 in [11], because all the reformulations involved are opt-reformulations.

Both (4) and (5) are extremely difficult problems to solve, due to the high number of binary variables and the nonconvexity of (2). Exact solutions of such MINLPs can be obtained via the spatial Branch-and-Bound (sBB) algorithm [1, 16, 17, 10] only for very small instances ( $\leq 10$  points,  $\leq 3$  hyperplanes,  $\leq 2$  dimensions). MINLP heuristics such as VNS [13] fare slightly better but are far from being able to tackle realistically-sized instances.

## 4 Approximations

In this section we propose three different Mixed-Integer Linear Programming (MILP) approximations for the problematic nonconvex constraints (2) in terms of the  $\ell_1$  and  $\ell_\infty$  norm, which can both be linearized exactly. We first remark the following inclusion relationships:

$$\begin{aligned}
U_1 &= \{\mathbf{w} \mid \|\mathbf{w}\|_1 \leq 1\} \subseteq \{\mathbf{w} \mid \|\mathbf{w}\|_2 \leq 1\} = U_2 \\
U_2 &= \{\mathbf{w} \mid \|\mathbf{w}\|_2 \leq 1\} \subseteq \{\mathbf{w} \mid \|\mathbf{w}\|_\infty \leq 1\} = U_\infty.
\end{aligned}$$

We shall exploit these inclusions to derive exactly linearizable approximations for  $U_2$ .

In the rest of this section, we shall discuss the exact linearization of both  $\ell_1$  and  $\ell_\infty$  unit constraints. We shall then propose three different approximations of (2): the  $\ell_\infty$  approximation, the  $\ell_1/\ell_\infty$  “sandwiching” approximation, and the  $\ell_1/\ell_\infty$  “alternating” approximation (shown graphically in Fig. 1).

### 4.1 Linearization of $\ell_1$ unit constraint

The linearization of the  $\ell_1$  unit constraint:

$$\forall j \leq m \quad \|\mathbf{w}_j\|_1 = 1, \quad (6)$$

which can also be written as  $\sum_{k \leq d} |w_{jk}| = 1$  for  $j \leq m$ , proceeds by repeated application of the ABSDIFF opt-reformulation [11] to each absolute value term  $|w_{jk}|$ : let  $w_{jk}^+, w_{jk}^-$  be added nonnegative continuous variables, replace (6) with:

$$\forall j \leq m \quad \sum_{k \leq d} (w_{jk}^+ + w_{jk}^-) = 1,$$

add the constraints:

$$\forall j \leq m, k \leq d \quad w_{jk} = w_{jk}^+ - w_{jk}^-,$$

and add the following exact reformulation of the linear complementarity conditions  $w_{jk}^+ w_{jk}^- = 0$  (for  $j \leq m, k \leq d$ ):

$$\begin{aligned} \forall j \leq m, k \leq d \quad w_{jk}^+ &\leq M \mu_{jk} \\ \forall j \leq m, k \leq d \quad w_{jk}^- &\leq M(1 - \mu_{jk}), \end{aligned}$$

where for  $j \leq m, k \leq d$   $\mu_{jk}$  are added binary variables that are 1 if  $w_{jk}^+$  has nonzero value.

## 4.2 Linearization of $\ell_\infty$ unit constraint

The linearization of the  $\ell_\infty$  unit constraint:

$$\forall j \leq m \quad \|\mathbf{w}_j\|_\infty = 1, \tag{7}$$

which can also be written as  $\max_{k \leq d} |w_{jk}| = 1$  for  $j \leq m$ , is a reformulation of the *narrowing* type [12] denoted by INFNORM, and was proposed in [6]. In full generality it works as follows. Consider a mathematical programming formulation  $P$  with a  $d$ -vector of variables  $x = (x_1, \dots, x_d) \in \mathbb{R}^d$  bounded in  $[-\alpha, \alpha]$  (for some  $\alpha > 0$ ) with the property that if  $x^*$  is a feasible solution of  $P$  then  $-x^*$  is also a feasible solution of  $P$  with the same objective function cost; and a constraint  $\|x\|_\infty = \alpha$ . The INFNORM reformulation is as follows:

- for all  $k \leq d$ , add a binary decision variable  $u_k$  to  $P$ ;
- delete the constraint  $\|x\|_\infty = \alpha$ ;
- add the following constraints:

$$\forall k \leq d \quad x_k \geq \alpha(1 - 2(1 - u_k)) \tag{8}$$

$$\sum_{k \leq d} u_k = 1. \tag{9}$$

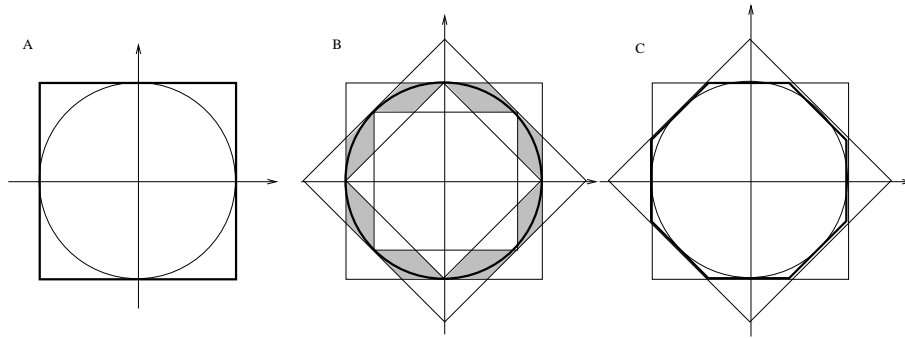
This reformulation being a narrowing, it guarantees that at least one optimum of the original problem is mapped into an optimum of the reformulation.

**Proposition 2.** *There exist an optimum of  $P$  which is also an optimum of INFNORM( $P$ ).*

*Proof.* Constraint (9) ensures that there is at least an index  $k \leq d$  such that, by (8) and the fact that the upper bound for  $x_k$  is  $\alpha$ , the value of  $x_k$  is exactly  $\alpha$ : this forces  $\|x\|_\infty$  to be precisely  $\alpha$ . Suppose now there is a feasible values of  $x$  with  $\|x\|_\infty = \alpha$  such that  $x_k \neq \alpha$  for all  $k$ . Since  $\|x\|_\infty = \alpha$ , this implies there is at least an index  $k \leq d$  for which  $x_k = -\alpha$ . By the symmetry assumption,  $-x$  is feasible and has the same objective function value as  $x$ .

### 4.3 Pure $\ell_\infty$ approximation

This approximation is based on simply replacing (2) by (7) and applying the INF NORM narrowing. Geometrically, we replace a hyperspherical feasible region with its hypercubical overapproximation, as shown graphically in Fig. 1 (A).



**Fig. 1.** Three types of approximation: pure (A), sandwich (B), and alternating (C).

An equivalent pure  $\ell_1$  approximation (where (2) was replaced by (6) and subsequently linearized) was tested but found to yield markedly inferior performances, and thence discarded.

### 4.4 Sandwiching $\ell_1/\ell_\infty$ approximation

This approximation, depicted graphically in Fig. 1 (B), consists of replacing (2) by the following constraints:

$$\begin{aligned} \forall j \leq m \quad 1 \leq \|\mathbf{w}_j\|_1 \leq \sqrt{d} \\ \forall j \leq m \quad \frac{1}{\sqrt{d}} \leq \|\mathbf{w}_j\|_\infty \leq 1. \end{aligned}$$

The above constraints can be linearized exactly by applying the reformulations of Sections 4.1 and 4.2.

#### 4.5 Alternating $\ell_1/\ell_\infty$ approximation

This approximation, depicted graphically in Fig. 1 (C), consists of replacing (2) by the following disjunction:

$$\forall j \leq m \quad (\|\mathbf{w}_j\|_\infty = 1 \quad \vee \quad \|\mathbf{w}_j\|_1 = \sqrt{d}).$$

This is modelled by introducing added binary variables  $\mu_j \in \{0, 1\}$  for  $j \leq m$  which have value 1 if the constraint  $\|\mathbf{w}_j\|_\infty = 1$  is active and 0 otherwise, and serve the purpose of alternating between  $\ell_1$  and  $\ell_\infty$  unit norm constraints.

### 5 Computational experiments

We considered a set of 8 instances whose statistics are given in Table 1. The cancer instance is taken from the Wisconsin Prognostic Breast Cancer (WPBC) Database [4]. The other instances are randomly generated. All instances were solved using 6 different MILP formulations: (4) and (5) with the “pure  $\ell_\infty$ ” (Sect. 4.3), “sandwich” (Sect. 4.4) and “alternating” (Sect. 4.5) approximations. All results were obtained on one core of a quad-CPU Intel Xeon 2.4GHz with 8GB RAM using ILOG CPLEX 11.0 [8].

<b>Instance</b>	<i>n</i>	<i>m</i>	<i>d</i>
cancer	194	3	2
cluster3	8	2	2
esr_150_8_10	150	8	10
esr_210_8_10	210	8	10
hsynt_1500_8_3	1500	8	2
synt_35_2_3	35	3	2
synt_60_2_3	60	3	2
synt_70_2_3	70	3	2

**Table 1.** Instance statistics.

The results are given in Table 2. Each group of 5 columns describes the results obtained by solving all instances in the test set with a particular formulation. Within each group, we recorded: the value of objective function of the original problem (1)  $\varepsilon = \max_{\substack{j \leq m \\ i \leq n}} \frac{|\mathbf{w}_j \mathbf{p}_i - w_j^0|}{\|\mathbf{w}\|_2} x_{ij}$  computed using the solution

given by the approximation; the CPU time (limited to 1800 seconds of user time); the Branch-and-Bound (BB) node index  $F$  where the recorded optimum was found; the total number  $N$  of BB nodes explored within the allowed time; and the final optimality gap reported by CPLEX (opt=0%). For each approximation type, boldface is used to emphasize the original problem ((4) or (5)) yielding better results for a particular measure (for  $F, N$  the comparative measure is  $F/N$ ). Underlined figures emphasize draws. Values marked by \* indicate a winning method for a particular instance: this is chosen by looking at (in order of priority):  $\varepsilon$ , CPU, gap,  $F/N$ .



Instance	(4)+Pure $\ell_\infty$					(5)+Pure $\ell_\infty$				
	$\varepsilon$	CPU	$F$	$N$	gap	$\varepsilon$	CPU	$F$	$N$	gap
cancer	<b>0.73*</b>	1800.21	677574	760654	<b>12.33%</b>	0.915	1800.11	<b>7671</b>	209071	67.65%
cluster3	<u>0.3046</u>	0.02	50	91	opt	<u>0.3046*</u>	0.02	<b>10</b>	90	opt
esr_150.8.10	<b>0.0744*</b>	1800.21	13879	13980	100.00%	0.0896	1800.26	<b>10164</b>	11264	100.00%
esr_210.8.10	0.1704	1800.32	<b>7874</b>	8075	100.00%	<b>0.1503*</b>	1800.34	5371	5472	100.00%
hsynt_1500.8.3	0.4497	1800.89	<b>511</b>	812	100.00%	<b>0.4368</b>	1800.46	1911	2129	100.00%
synt_35.2.3	0.0561	<b>6.33</b>	5103	7500	opt	<b>0.0556</b>	31.53	<b>10211</b>	41062	opt
synt_60.2.3	<b>0.0614</b>	<b>125.81</b>	<b>27001</b>	52389	opt	0.071	275.4	84071	142184	opt
synt_70.2.3	0.0749	<b>47.52</b>	8661	10238	opt	<b>0.0741</b>	163.43	<b>24223</b>	62325	opt

Instance	(4)+Sandwich					(5)+Sandwich				
	$\varepsilon$	CPU	$F$	$N$	gap	$\varepsilon$	CPU	$F$	$N$	gap
cancer	<b>0.8922</b>	1800.07	104385	119285	<b>52.53%</b>	0.9286	1800.12	<b>105540</b>	243340	53.52%
cluster3	<u>0.3046</u>	0.04	<b>44</b>	86	opt	<u>0.3046</u>	<b>0.03</b>	90	91	opt
esr_150.8.10	0.1121	1800.28	18411	18612	100.00%	<b>0.1006</b>	1800.2	8841	8942	100.00%
esr_210.8.10	0.1593	1800.56	11158	11175	100.00%	<b>0.157</b>	1800.43	<b>5964</b>	6165	100.00%
hsynt_1500.8.3	0.5103	1801.88	<b>860</b>	1962	100.00%	<b>0.416*</b>	1800.6	1701	2002	100.00%
synt_35.2.3	<b>0.0473</b>	<b>6.7</b>	891	3314	opt	0.0482	127.63	<b>28025</b>	117132	0.27%
synt_60.2.3	<b>0.0639</b>	<b>88.06</b>	19320	24685	2.37%	0.0647	223.15	<b>36521</b>	105342	<b>opt</b>
synt_70.2.3	<b>0.0656</b>	<b>199.78</b>	84489	84490	opt	0.0664	648.44	<b>71625</b>	334408	opt

Instance	(4)+Alternating					(5)+Alternating				
	$\varepsilon$	CPU	$F$	$N$	gap	$\varepsilon$	CPU	$F$	$N$	gap
cancer	<b>0.73</b>	1800.11	131488	163688	51.47%	0.82	1800.11	<b>168988</b>	269689	<b>50.82%</b>
cluster3	0.32	0.25	<b>38</b>	682	opt	<b>0.318</b>	<b>0.2</b>	120	774	opt
esr_150.8.10	0.1511	1800.2	<b>15374</b>	15875	100.00%	<b>0.1076</b>	1800.26	11060	11161	100.00%
esr_210.8.10	0.2184	1800.29	14284	14685	100.00%	<b>0.1871</b>	1800.29	7374	7575	100.00%
hsynt_1500.8.3	0.4741	1800.75	478	879	100.00%	<b>0.4713</b>	1800.65	<b>501</b>	2204	100.00%
synt_35.2.3	<u>0.0462</u>	155.08	86293	90612	opt	<u>0.0462*</u>	<b>85.31</b>	<b>12201</b>	50181	opt
synt_60.2.3	<u>0.0579</u>	165.72	<b>20568</b>	40212	opt	<u>0.0579*</u>	<b>113.27</b>	45208	50551	opt
synt_70.2.3	<b>0.0584*</b>	206.55	<b>14715</b>	45222	opt	0.0596	259.43	50931	73956	opt

Table 2. Computational results.

## 6 Conclusion

We presented several approximate MILP formulations for the bHCP. In particular, we discussed some techniques for linearizing a unit  $\ell_2$  norm constraint approximately. We evaluated the performance of the linearizations on an instance test set. Although there was no clear winner, and all linearization variants yielded good results on at least some instance, approximations derived from (5) seemed to be marginally better than those derived from (4).

## References

1. C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method,  $\alpha$ BB, for general twice-differentiable constrained NLPs: II. Implementation and computational results. *Computers & Chemical Engineering*, 22(9):1159–1179, 1998.
2. P. K. Agarwal and C. M. Procopiuc. Approximation algorithms for projective clustering. *Journal of Algorithms*, 46:115–139, 2003.
3. E. Amaldi, A. Ceselli, and K. Dhyani. Column generation for the min-hyperplane clustering problem. *7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, 2008.

4. A. Asuncion and D.J. Newman. UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, <http://www.ics.uci.edu/~mllearn/MLRepository.html>, 2007.
5. P. S. Bradley and O. L. Mangasarian.  $k$ -Plane clustering. *J. of Global Optimization*, 16(1):23–32, 2000.
6. S. Coniglio and F. Italiano. Hyperplane clustering and piecewise linear model fitting. Master’s thesis, DEI, Politecnico di Milano, 2007.
7. S. Har-Peled. No coresets, no cry. In K. Lodaya and M. Mahajan (Eds.), *FSTTCS*, LNCS 3328, p.324–335, 2004.
8. ILOG. *ILOG CPLEX 11.0 User’s Manual*. ILOG S.A., Gentilly, France, 2008.
9. Stefan Langerman and Pat Morin. Covering things with things. *Discrete Computational Geometry*, 33(4):717–729, 2005.
10. L. Liberti. Writing global optimization software. In L. Liberti and N. Maculan, editors, *Global Optimization: from Theory to Implementation*, pages 211–262. Springer, Berlin, 2006.
11. L. Liberti. Reformulation techniques in mathematical programming, November 2007. Thèse d’Habilitation à Diriger des Recherches.
12. L. Liberti. Reformulations in mathematical programming: Definitions. In R. Aringhieri, R. Cordone, and G. Righini, editors, *Proceedings of the 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, pages 66–70, Crema, 2008. Università Statale di Milano.
13. L. Liberti, N. Mladenović, and G. Nannicini. A good recipe for solving MINLPs. In V. Maniezzo, editor, *Proceedings of Metaheuristics 2008 Conference*, Bertinoro, 2008 (accepted).
14. N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Operations Research Letters*, 1:194–197, 1982.
15. N. Mishra, R. Motwani, and S. Vassilvitskii. Sublinear projective clustering with outliers. *15th Annual Fall Workshop on Computational Geometry and Visualization*, 2005.
16. E.M.B. Smith and C.C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 23:457–478, 1999.
17. M. Tawarmalani and N.V. Sahinidis. Semidefinite relaxations of fractional programming via novel techniques for constructing convex envelopes of nonlinear functions. *Journal of Global Optimization*, 20(2):137–158, 2001.