# Minimum Enclosing Ball for Anomaly Detection on Biological Data using Frank-Wolfe based algorithms

José Chacón, Inês Jesus, Alejandra Cruces and Mario Tapia

University of Padua

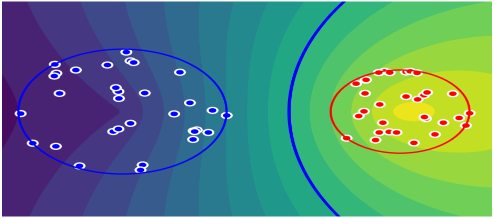
# Introduction

- The **Minimum Enclosing Ball** (MEB) is a problem with a wide range of applications, of which anomaly detection is a prominent one.
- **Objective**: To understand and implement the MEB problem through three different variants of the Frank-Wolfe algorithm: the Pairwise Frank-Wolfe, the Blended Pairwise Conditional Gradient and a variant of the Away-Steps.
- **Application**: To find anomalies in the context of four different biological problems.

# Minimum Enclosing Ball Problem

The problem is to find the smallest n-sphere that contains a given set of points in a Euclidean space. This sphere is called the minimum enclosing ball (MEB) of the point set.

- **Applications** in clustering, data classification, machine-learning and facility location, anomaly detection, among others.
- **Advantages**: Robustness, geometric interpretability and high-dimensionality



# Minimum Enclosing Ball Problem

## The Dual MEB

In high-dimensional spaces, transforming the MEB problem into its dual form can help reduce dimensionality and simplify the optimization process.

$$(\mathcal{D}1) \quad \max_{\mu} \qquad \phi(\mu) := \sum_{j=1}^{m} \mu_j (a^j)^T a^j - \left( \sum_{j=1}^{m} \mu_j a^j \right)^T \sum_{j=1}^{m} \mu_j a^j$$

$$\text{subject to}$$

$$\sum_{j=1}^{m} \mu_j = 1$$

$$\mu_i \geq 0, \quad i = 1, ..., m \quad ,$$

where vector $\mu$ contains the Lagrangian multipliers respective to the constraints of problem $(\mathcal{P})$.

# Minimum Enclosing Ball Problem

## The Dual MEB: Convex Formulation

We can express the dual formulation of the MEB ($\mathcal{D}1$) as a simplified convex version:

$$(\mathcal{D}2) \quad \min_{\mu \in \triangle^{m-1}} -\phi(\mu) \quad ,$$

where the Unit Simplex

$$\triangle^{m-1} = \left[ \mu \in \mathbb{R}^m \middle| \sum_{i=1}^{m} \mu_i = 1 \quad \wedge \quad \mu_j \geq 0, \quad j = 1, ..., m \right]$$

satisfies the previous constraint of non-negativity and sum-to-one of the lagrangian multipliers.

# Anomaly Detection Problem

Anomaly detection is the task of identifying data points that deviate significantly from the normal behavior of the data.

## MEB Approach

To find the smallest hypersphere that contains all the data points used for training, i.e., points that are known to not be anomalies *a priori*. From there, new data points that lie outside of the hypersphere obtained in the training stage are considered to be anomalies.

# Algorithms

In this work we present three algorithms:

- Pairwise Frank Wolfe (Lacoste-Julien and Jaggi (2015) and Mitchell, Dem'yanov, and Malozemov (1974))
- Blended Pairwise Conditional Gradients (BPCG) (Tsuji, Tanaka, and Pokutta (2022))
- $(1+\epsilon)$-approximation to the MEB($\mathcal{A}$) algorithm (Yildirim (2008))

# Algorithm 1: Pairwise Frank Wolfe

- The direction chosen in every step is based on a weight change from the away atom $a_t$ to the FW atom $w_t$, while keeping all the other weights unchanged.
- Cons: Swap steps.

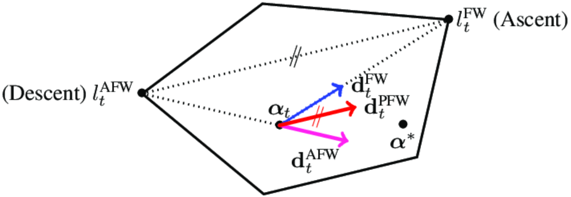



Figure 1: Illustration of Frank-Wolfe towards, away and pairwise directions

# Algorithm 2: Blended Pairwise Conditional Gradients (BPCG)

- A blending criterion is added that favors local steps made over the convex hull of the current active vertex set, offering a sparser solution without swap steps.
- The only reason for new atoms being added to $S_t$ is a sufficient decrease in the local pairwise gap.

# Algorithms 1 and 2: MEB adaptation of PFW and BPCG

## Gradient

Finding the step direction in algorithms PFW and BPCG, requires the computation of the gradient of $-\phi$ to compute:

$$\nabla_i(-\phi(\mu)) = 2\left(a^i\right)^T \sum_{j=1}^{m} a^j \mu_j - (a^i)^T a^i.$$

# Algorithms 1 and 2: MEB adaptation of PFW and BPCG

In the case of Frank-Wolfe iterations on the PFW and BPCG algorithms, we compute $w_t$ as:

$$w_t = argmin_{w \in \triangle^{m-1}} w^T \nabla\big(-\phi(\mu_t)\big) \tag{1}$$

From the Fundamental Theorem of Linear Programming the solution should be a vertex of $\triangle^{m-1}$. Since the vertices of $\triangle^{m-1}$ correspond to the standard basis vectors $e_j \in \mathbb{R}^m$, $j = 1, ..., m$, equation (1) obtains the same result as

$$w_t = argmin_{e_j \in \mathbb{R}^m} e_j^T \nabla\big(-\phi(\mu_t)\big)$$

So, if we take $i = i_t^w := argmin_{j \in \{1,\ldots,m\}} \Big(\nabla\big(-\phi(\mu_t)\big)\Big)_j$ , we see that $w_t = e_i$.

# Algorithms 1 and 2: Step rule

As step rule we chose to use the short-step rule suggested by Tsuji (2022):

$$\lambda_t = \frac{\langle -\nabla \phi(\mu), d_t \rangle}{L \| d_t \|^2} \quad .$$

To determine $L$, we start by calculating the Hessian matrix, $H$, where each component is determined as:

$$h_{ij} = -\frac{\partial^2 \left( \phi(\mu) \right)}{\partial \mu_i \partial \mu_j} = 2 \left( a^j \right)^T a^i$$

We can then obtain the Lipschitz constant as

$$L = \lambda_{\max}(H),$$

where $\lambda_{\max}(H)$ is the largest eigenvalue of $H$ and it needs to be positive.

# Algorithm 1: Pairwise Frank Wolfe

**Algorithm 1** Pairwise Frank-Wolfe algorithm for the MEB problem: `PFW_MEB`$(x^{(0)}, \mathcal{A}, \epsilon)$

**Require:** point $\mu_0 \in \mathcal{A}$

$S_0 \leftarrow \{\mu_0\}$

**for** $t = 0$ to $T$ **do**

$i_t^w \leftarrow argmin_{j\in\{1,\dots m\}}(-\nabla\phi(\mu_t))_j$

$d_t^{FW} = e_{\mu_t} - e_{i_t^w}$ ▷ FW direction

$i_t^a \leftarrow argmax_{\substack{j\in\{1,\dots,m\}\\ e_j \in S_t}}(-\nabla\phi(\mu_t))_j$ ▷ Away Atom

$g_t^{FW} \leftarrow \langle \nabla - \phi(\mu_t), d_t^{FW}\rangle$

**if** $g_t^{FW} \leq \epsilon$ **then** ▷ FW gap is small enough

**return** $x^t$

**end if**

$d_t \leftarrow e_{i_t^w} - e_{i_t^a}$ ▷ PW direction

$\lambda_{max} \leftarrow \alpha_a^t$

$\lambda_t \leftarrow \max\left(0, \min\left(\lambda_{max}, \frac{\langle -\nabla\phi(\mu_t), d_t\rangle}{L||d_t||_2^2}\right)\right)$

$\mu_{t+1} \leftarrow \mu_t + \lambda_t d_t$ ▷ Update $\mu$

$\alpha_{i_t^a}^{(t+1)} = \alpha_{i_t^a}^{(t)} - \lambda_t$

$\alpha_{i_t^w}^{(t+1)} = \alpha_{i_t^w}^{(t)} + \lambda_t$ ▷ Updating weights

$S_{t+1} \leftarrow \{e_{i_t^w} \in A \text{ s.t. } \alpha_a^{t+1} > 0\}$ ▷ Update Active Set

**end for**

# Algorithm 2: Blended Pairwise Conditional Gradients (BPCG)

**Algorithm 2** Blended Pairwise Conditional Gradients (BPCG) for the MEB problem

**Require:** convex smooth function $f$, start vertex $\mu_0 \in \triangle^{m-1}$.
**Ensure:** Weights Initialization: $\alpha_t^v = 1$ for $v = \mu_0$, and 0 otherwise.

$\mu_0 \leftarrow \frac{1}{m} * 1$ ▷ Feasible initialization
$S_0 \leftarrow \{\mu_0\}$
**for** $t = 0$ to $T-1$ **do**
  $i_t^a \leftarrow argmax_{\substack{j \in \{1,\dots,m\} \\ e_j \in S_t}} (-\nabla\phi(\mu_t))_j$ ▷ Away vertex
  $i_t^s \leftarrow argmin_{\substack{j \in \{1,\dots,m\} \\ e_j \in S_t}} (-\nabla\phi(\mu_t))_j$ ▷ Local FW
  $i_t^w \leftarrow argmin_{j \in \{1,\dots m\}} (-\nabla\phi(\mu_t))_j$ ▷ Global FW

  **if** $\langle -\nabla\phi(\mu_t), e_{i_t^a} - e_{i_t^s} \rangle \geq \langle -\nabla\phi(\mu_t), \mu_t - e_{i_t^w} \rangle$ **then**
    $d_t = e_{i_t^a} - e_{i_t^s}$ ▷ PW step
    $\Lambda_t^* \leftarrow c[\mu_t](e_{i_t^a})$
    $\lambda_t \leftarrow \max\left(0, \min\left(\Lambda_t^*, \frac{\langle \nabla\phi(\mu_t), d_t \rangle}{L||d_t||_2^2}\right)\right)$

    **if** $\lambda_t < \Lambda_t^*$ **then**
      $S_{t+1} \leftarrow S_t$ ▷ Descent step
      $\alpha^{e_{i_t^a}} = \alpha^{e_{i_{t-1}^a}} - \lambda_t$, $\alpha^{e_{i_t^s}} = \alpha^{e_{i_{t-1}^s}} + \lambda_t$ ▷ Update
weights
    **else**
      $S_{t+1} \leftarrow S_t \setminus \{e_{i_t^a}\}$ ▷ Drop step
      $\alpha^{e_{i_t^a}} = 0$, $\alpha^{e_{i_t^s}} = \alpha^{e_{i_{t-1}^s}} + \lambda_t$ ▷ Update weights
    **end if**
  **else**
    $d_t = \mu_t - e_{i_t^w}$ ▷ FW step
    $\lambda_t \leftarrow \max\left(0, \min\left(1, \frac{\langle \nabla\phi(\mu_t), d_t \rangle}{L||d_t||_2^2}\right)\right)$
    $S_{t+1} \leftarrow S_t \cup \{w_t\}$ (or $S_{t+1} \leftarrow i_t^w$ if $\lambda_t = 1$)
    $\alpha_t = \alpha_{t-1}(1-\lambda_t)$, $\alpha^{e_{i_t^w}} = \alpha^{e_{i_t^w}} + \lambda_t$ ▷ Update weights
  **end if**
  $\mu_{t+1} \leftarrow \mu_t - \lambda_t d_t$ ▷ Update $\mu$
**end for**

# Algorithm 3: $(1+\epsilon)$-approximation to the MEB($\mathcal{A}$) algorithm

## Some notions

Given $\epsilon > 0$, a ball $\mathcal{B}_{\mathbf{c},\rho}$ is said to be a $(1+\epsilon)$-approximation to the MEB($\mathcal{A}$), $\mathcal{B}_{\mathbf{c}_{\mathcal{A}},\rho_{\mathcal{A}}}$, if

$$\mathcal{A} \subset \mathcal{B}_{\mathbf{c},\rho} \quad \wedge \quad \rho \leq (1+\epsilon)\rho_{\mathcal{A}}.$$

A subset $\mathcal{X} \subseteq \mathcal{A}$ is said to be an $\epsilon$-core set (or a core set) of $\mathcal{A}$ if

$$\rho_{\mathcal{X}} \leq \rho_{\mathcal{A}} \leq (1+\epsilon)\rho_{\mathcal{X}},$$

where $\mathcal{B}_{\mathbf{c}_{\mathcal{X}},\rho_{\mathcal{X}}} :=$ MEB($\mathcal{X}$).

# Algorithm 3: $(1+\epsilon)$-approximation to the MEB($\mathcal{A}$) algorithm

**Algorithm 3** $(1+\epsilon)$-approximation to MEB($\mathcal{A}$)

**Require:** Input set of points $\mathcal{A} = \{a^1, \dots, a^m\} \subset \mathbb{R}^n, \epsilon > 0$.

$\alpha \leftarrow argmax_{i=1,\dots,m} \|a^i - a^1\|^2$ ▷ $\alpha$ and $\beta$ for initial approximation

$\beta \leftarrow argmax_{i=1,\dots,m} \|a^i - a^\alpha\|^2$

$u^0_\alpha \leftarrow 1/2,\ u^0_\beta \leftarrow 1/2$ ▷ Feasible solution

$\mathcal{X}_0 \leftarrow \{a^\alpha, a^\beta\}$ ▷ Coreset initialization

$c^0 \leftarrow \sum_{i=1}^m u^0_i a^i$ ▷ Initial center

$\gamma^0 \leftarrow \Phi(u^0)$ ▷ Initial $\mathcal{D}$ value

$\kappa \leftarrow argmax_{i=1,\dots,m} \|a^i - c^0\|^2$ ▷ Farthest point from the center index

$\xi \leftarrow argmin_{i:a^i \in \mathcal{X}_0} \|a^i - c^0\|^2$ ▷ Closest $\mathcal{X}_0$ point from the center index

$\delta^+_0 \leftarrow (\|a^\kappa - c^0\|^2/\gamma^0) - 1$

$\delta^-_0 \leftarrow 1 - (\|a^\xi - c^0\|^2/\gamma^0)$

$\delta_0 \leftarrow max\{\delta^+_0, \delta^-_0\}$ ▷ Initial $\delta$ value

$k \leftarrow 0$

# Algorithm 3: $(1+\epsilon)$-approximation to the MEB($\mathcal{A}$) algorithm

**Algorithm 4** $(1+\epsilon)$-approximation to MEB($\mathcal{A}$)

**Require:** Input set of points $\mathcal{A} = \{a^1, \dots, a^m\} \subset \mathbb{R}^n, \epsilon > 0$.
**while** $\delta_k > (1+\epsilon)^2 - 1$ **do**
  **if** $\delta_k > \delta_k^-$ **then** ▷ Plus iteration
    $\lambda^k \leftarrow \delta_k / [2(1+\delta_k)]$
    $k \leftarrow k+1$
    $u^k \leftarrow (1-\lambda^{k-1})u^{k-1} + \lambda^{k-1}e^\kappa$ ▷ Feasible solution update
    $c^k \leftarrow (1-\lambda^{k-1})c^{k-1} + \lambda^{k-1}a^\kappa$ ▷ Center moved toward the vertex $\kappa$ of the unit simplex
    $\mathcal{X}^k \leftarrow \mathcal{X}^{k-1} \cup \{a^k\}$
  **else** ▷ Minus iteration
    $\lambda^k \leftarrow \min\left\{\frac{\delta_k^-}{2(1-\delta_k^-)}, \frac{u_\xi^k}{1-u_\xi^k}\right\}$
    **if** $\lambda^k = u_\xi^k/(1-u_\xi^k)$ **then**
      $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k \setminus \{a^\xi\}$ ▷ Drop step
    **else**
      $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k$ ▷ Coreset update
    **end if**
    $k \leftarrow k+1$
    $u^k \leftarrow (1+\lambda^{k-1})u^{k-1} - \lambda^{k-1}e^\xi$ ▷ Feasible solution update
    $c^k \leftarrow (1+\lambda^{k-1})c^{k-1} - \lambda^{k-1}a^\xi$ ▷ Center moved away from the vertex $\xi$ of the unit simplex
  **end if**
  $\gamma^0 \leftarrow \Phi(u^k)$ ▷ Parameters update
  $\kappa \leftarrow argmax_{i=1,\dots,m}\|a^i - c^0\|^2$
  $\xi \leftarrow argmin_{i:a^i\in\mathcal{X}_0}\|a^i - c^k\|^2$
  $\delta_k^+ \leftarrow (\|a^\kappa - c^k\|^2/\gamma^k) - 1$
  $\delta_k^- \leftarrow 1 - (\|a^\xi - c^k\|^2/\gamma^k)$
  $\delta_k \leftarrow \max\{\delta_k^+, \delta_k^-\}$
**end while**
**Output** $c^k, \mathcal{X}_k, u^k, \sqrt{(1+\delta_k)\gamma^k}$

# Convergence Analysis

- All of the algorithms presented enjoy linear convergence when applied to the MEB problem.
- The MEB problem is a case of the Non-Strongly Convex generalization of the results in Lacoste [1], where global linear convergence is still guaranteed.
- BPCG is expected to perform at least as good as the PFW, if not better due to the lack of swap steps.
- The $(1+\epsilon)$-approximation to the MEB algorithm converges in $O(1/\epsilon)$ iterations.

# Metrics

In the anomaly detection problem, we want to assess how well the model predicts the anomaly data in a new dataset. Therefore, we report **Recall**:

$$Recall = \frac{TP}{TP+FN}$$

- Recall measures the **fraction of true anomalies** detected by the model.

# Datasets

1. **Breast Cancer:** 569 samples of women with breast tumors and 30 features of the tumor. The objective is to detect malignant tumors (anomaly).
2. **Breast cancer gene expression:** 801 instances and 20,531 features. The focus is placed on breast cancer samples identified with the gen BRCA (Breast Invasive Carcinoma).
3. **Vertebral column pathology:** 310 instances of patients with 6 biomechanical features. The goal is to detect if a patient has vertebral column pathology (anomaly).
4. **Maternity risk:** 1013 instances of pregnant women with 5 features of biomedical indicators.The objective is to identify pregnant women with high-risk levels of maternal mortality during pregnancy (anomaly).

# Dataset Preprocessing (I)

- We divided the dataset into two sets: *training* and *test*, with a proportion of 70/30.
- Training set to get the radius and center of the non-anomalous points.
- Test set uses as input the radius and center to identify the anomalies. We relied on recall to assess goodness of fit.
- On the Breast cancer dataset, we apply the Standard Scaler technique that transforms the features of a dataset to have zero mean and unit variance.

# Dataset Preprocessing (II)

We reduce the number of features for each dataset, except for the Breast cancer gene expression, for which we wish to keep it large-scale for later comparison purposes.

| Dataset | Features | Optimal K |
|---|---|---|
| **Breast Cancer** | 30 | 4 |
| **Vertebral Column** | 6 | 2 |
| **Maternity Risk** | 6 | 3 |

Table 1: Optimal K search

# Results

- In all the experiments, we set a maximum number of iterations of $T = 100,000$, and a *tolerance* of $10^{-4}$.
- In the case of PFW and BPCG we use the same initial point and stopping condition.

# Results

| | **Time** (ms) | | | **Iterations** | | |
|---|---|---|---|---|---|---|
| **Dataset** | **PFW** | **BPCG** | **MEB(A)** | **PFW** | **BPCG** | **MEB(A)** |
| **1** | 305.42 | 637.14 | 53.59 | 1,484 | 2,998 | 44 |
| **2** | 253,589.34 | 255,090.94 | 2,659.64 | 100,000 | 100,000 | 44 |
| **3** | 1,886.47 | 4,911.60 | 48.71 | 20,411 | 52,194 | 105 |
| **4** | 44,240.58 | 42,329.92 | 65.24 | 100,000 | 100,000 | 28 |

Table 2: Computational results with short step rule for PFW and BPCG

| | **Time** (ms) | | **Iterations** | |
|---|---|---|---|---|
| **Dataset** | **PFW** | **BPCG** | **PFW** | **BPCG** |
| **1** | 65.40 | 31.50 | 396 | 163 |
| **2** | 332,794.27 | 334,992.27 | 100,000 | 100,000 |
| **3** | 18.76 | 155.47 | 158 | 2,155 |
| **4** | 62.19 | 70.12 | 168 | 292 |

Table 3: Computational results with diminishing step size

# Results

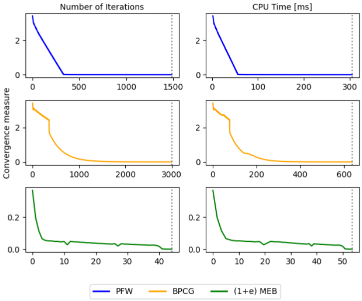



Figure 2: Convergence measure for Dataset 1

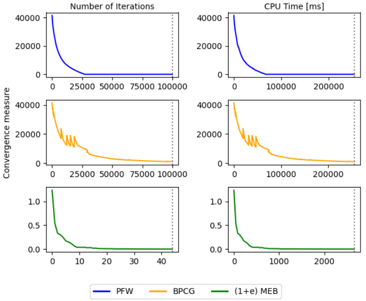
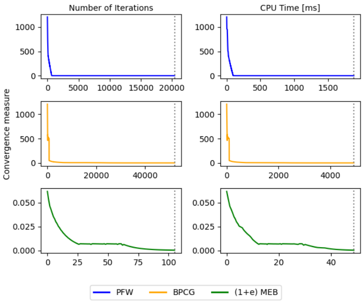



Figure 3: Convergence measure for Dataset 2

# Results



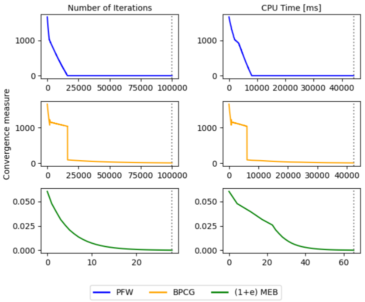




Figure 4: Convergence measure for Dataset 3

Figure 5: Convergence measure for Dataset 4

# Results

| | Recall Training | | | Recall Test | | |
|---|---|---|---|---|---|---|
| **Dataset** | **PFW** | **BPCG** | **MEB(A)** | **PFW** | **BPCG** | **MEB(A)** |
| **1** | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| **2** | 0.84 | 0.85 | 0.84 | 0.85 | 0.86 | 0.85 |
| **3** | 0.59 | 0.59 | 0.59 | 0.51 | 0.51 | 0.51 |
| **4** | 0.43 | 0.44 | 0.43 | 0.48 | 0.50 | 0.48 |

Table 4: Recall results with short step rule for BPCG and PFW.

| | Recall Training | | | Recall Test | | |
|---|---|---|---|---|---|---|
| **Dataset** | **PFW** | **BPCG** | **MEB(A)** | **PFW** | **BPCG** | **MEB(A)** |
| **1** | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| **2** | 0.84 | 0.85 | 0.84 | 0.85 | 0.85 | 0.85 |
| **3** | 0.59 | 0.59 | 0.59 | 0.51 | 0.51 | 0.51 |
| **4** | 0.43 | 0.44 | 0.43 | 0.48 | 0.48 | 0.48 |

Table 5: Recall results with diminishing step size

# Conclusion

- We presented the MEB problem approach to the anomaly detection task by means of three different algorithms related to the classical Frank-Wolfe.
- The three algorithms implemented allow drop steps, representing a significant improvement of the Frank-Wolfe algorithm, allowing sparser solutions.

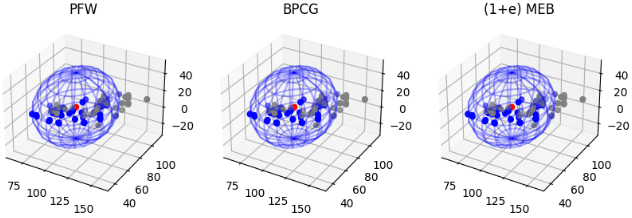



Figure 6: MEB for Maternity risk dataset

# Conclusion

- The $(1+\epsilon)$-approximation to the MEB algorithm outperformed the other methods on the datasets, showing faster convergence and lower computational cost.
- This is specially true when working with large-scale datasets.
- The initialization of the $\alpha$ and $\beta$ as coreset points seems to give a good approximation of the optimal diameter from the very beginning.
- Gradient computations are costly (PFW and BPCG).