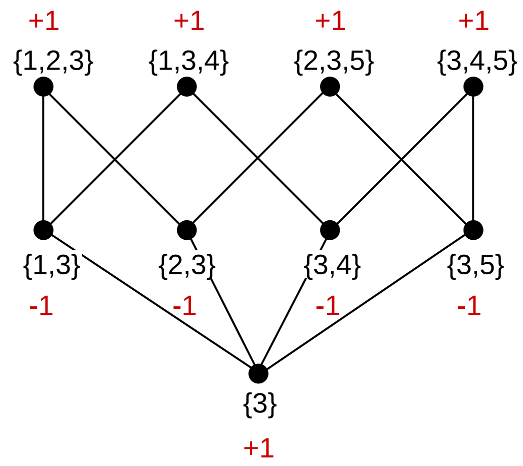
Belief Propagation on Partially Ordered Sets

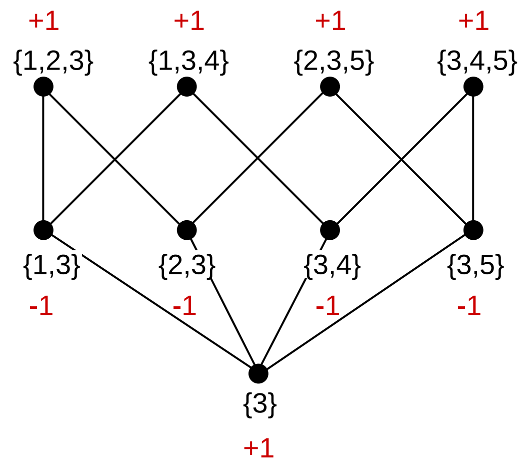
Robert J. McEliece
California Institute of Technology



International Symposium on
Mathematical Theory of Networks and Systems
University of Notre Dame
August 13, 2002

Belief Propagation on Partially Ordered Sets

Robert J. McEliece*
California Institute of Technology



International Symposium on
Mathematical Theory of Networks and Systems
University of Notre Dame
August 13, 2002

* *With much help from Muhammed Yildirim, Jonathan Harel, Jeremy Thorpe, and Ravi Palanki.*

A Big Tip of the Hat to:

Jonathan Yedidia, William Freeman,
and Yair Weiss, the authors of

*"Generalized Belief Propagation
and Free Energy Minimization,"*

which inspired this paper.

Outline of the Talk

Outline of the Talk

- The Problem Statement

Outline of the Talk

- The Problem Statement
- Significance of the Problem

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach
- Solution Via “Poset Belief Propagation”

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach
- Solution Via “Poset Belief Propagation”
- Experimental Results

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach
- Solution Via “Poset Belief Propagation”
- Experimental Results
- Connection Between the Statistical Physics and the BP Approaches

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach
- Solution Via “Poset Belief Propagation”
- Experimental Results
- Connection Between the Statistical Physics and the BP Approaches
- Open Problems

Outline of the Talk

- The Problem Statement
- Significance of the Problem
- A Statistical Physics Approach
- Solution Via “Poset Belief Propagation”
- Experimental Results
- Connection Between the Statistical Physics and the BP Approaches
- Open Problems
- Coffee Break

A General Computational Problem

- Variables $\{x_1, \ldots, x_n\}$, $x_i \in A = \{0, 1, \ldots, q-1\}$.
- $\mathcal{R} = \{R_1, \ldots, R_M\}$, a collection of subsets of $\{1, 2, \ldots, n\}$.
- A set of nonnegative "local potentials" $\{\alpha_R(\boldsymbol{x}_R) : R \in \mathcal{R}\}$.

A General Computational Problem

- Variables $\{x_1, \ldots, x_n\}$, $x_i \in A = \{0, 1, \ldots, q-1\}$.
- $\mathcal{R} = \{R_1, \ldots, R_M\}$, a collection of subsets of $\{1, 2, \ldots, n\}$.
- A set of nonnegative "local potentials" $\{\alpha_R(\boldsymbol{x}_R) : R \in \mathcal{R}\}$.
- Define the global (Boltzmann) probability density function;

$$B(\boldsymbol{x}) = \frac{1}{Z} \prod_{R \in \mathcal{R}} \alpha_R(\boldsymbol{x}_R),$$

where

$$Z = \sum_{\boldsymbol{x} \in A^n} \prod_{R \in \mathcal{R}} \alpha_R(\boldsymbol{x}_R) \qquad \textit{(Partition function)}$$

$$(F = -\ln Z = \textit{Helmholtz free energy}).$$

A General Computational Problem, Continued

Problem: Compute, *exactly or approximately,* the Helmholtz free energy F and some or all of the *local marginal densities* of the Boltzmann density:

$$\begin{aligned} B_R(\boldsymbol{x}_R) &= \sum_{\boldsymbol{x}_{R^c} \in A^{R^c}} B(\boldsymbol{x}) \\ &= \frac{1}{Z} \sum_{\boldsymbol{x}_{R^c} \in A^{R^c}} \prod_{S \in \mathcal{R}} \alpha_S(\boldsymbol{x}_S), \end{aligned}$$

for $R \in \mathcal{R}$.

Applications:

Appropriately interpreted, this computational problem includes:

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.
- Probabilistic inference in Bayesian networks.

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.
- Probabilistic inference in Bayesian networks.
- Finite Fourier transforms.

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.
- Probabilistic inference in Bayesian networks.
- Finite Fourier transforms.
- Free energy computations in statistical physics.

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.
- Probabilistic inference in Bayesian networks.
- Finite Fourier transforms.
- Free energy computations in statistical physics.

$$\vdots$$

Applications:

Appropriately interpreted, this computational problem includes:

- Turbo/LDPC decoding.
- Probabilistic inference in Bayesian networks.
- Finite Fourier transforms.
- Free energy computations in statistical physics.

$$\vdots$$

(But we won't discuss these applications)

A Simple Example.

Alphabet: $A = \{0, 1\}$.

Local domains: $\mathcal{R} = \{\{1, 2, 3\}, \{1, 3, 4\}, \{2, 3, 5\}, \{3, 4, 5\}\}$.

Local potentials:

$$\alpha_i(x, y, z) = \begin{cases} 1/2 & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, 2, 3, 4.$$

A Simple Example.

Alphabet: $A = \{0, 1\}$.

Local domains: $\mathcal{R} = \{\{1, 2, 3\}, \{1, 3, 4\}, \{2, 3, 5\}, \{3, 4, 5\}\}$.

Local potentials:

$$\alpha_i(x, y, z) = \begin{cases} 1/2 & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, 2, 3, 4.$$

Answers:

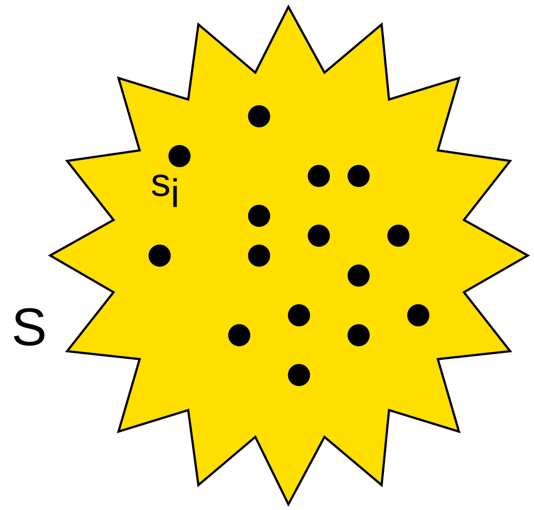
$$Z = 1/8$$

$$F = \ln 8.$$

$$B(x_1, x_2, x_3, x_4, x_5) = \begin{cases} 1/2 & \text{if } x_1 = x_2 = \cdots = x_5 \\ 0 & \text{otherwise.} \end{cases}$$

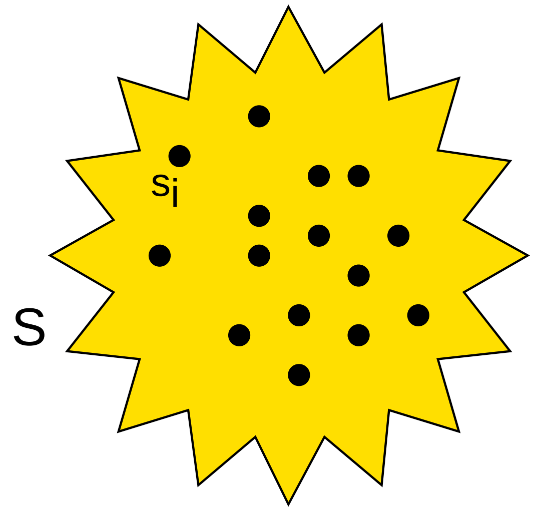
$$B_i(x, y, z) = \begin{cases} 1/2 & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases}$$

A Statistical Physics Approach



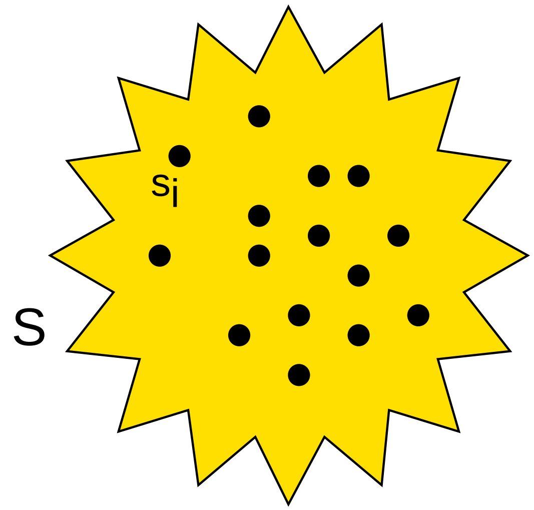
- $S = \{s_1, \ldots, s_n\} = n$ identical particles.

A Statistical Physics Approach



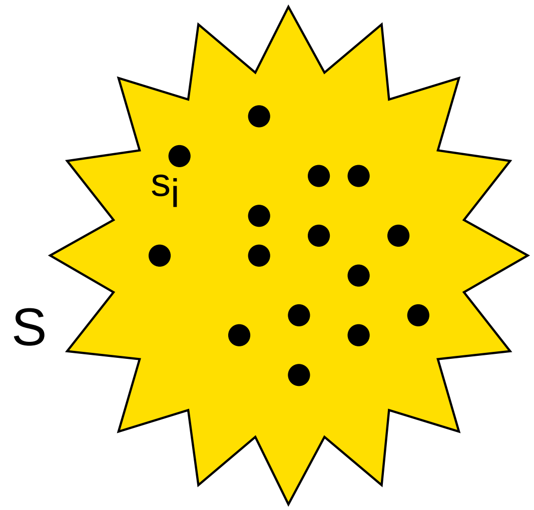
- $S = \{s_1, \ldots, s_n\} = n$ identical particles.
- "Spin" of $s_i = x_i \in A = \{0, 1, \ldots, q-1\}$.

A Statistical Physics Approach



- $S = \{s_1, \ldots, s_n\} = n$ identical particles.
- "Spin" of $s_i = x_i \in A = \{0, 1, \ldots, q-1\}$.
- System "configuration" $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$.

A Statistical Physics Approach



- $S = \{s_1, \ldots, s_n\} = n$ identical particles.
- "Spin" of $s_i = x_i \in A = \{0, 1, \ldots, q-1\}$.
- System "configuration" $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$.
- $E(x_1, \ldots, x_n) =$ energy of configuration $\boldsymbol{x}$. (Hamiltonian)

$$= -\sum_{R \in \mathcal{R}} \ln \alpha_R(\boldsymbol{x}_R).$$

A Statistical Physics Approach

- $b(\boldsymbol{x})$ = "Trial" probability of configuration $\boldsymbol{x}$.

A Statistical Physics Approach

- $b(\boldsymbol{x})$ = "Trial" probability of configuration $\boldsymbol{x}$.
- Average energy: $U = \sum_{\boldsymbol{x} \in A^n} b(\boldsymbol{x}) E(\boldsymbol{x})$.

A Statistical Physics Approach

- $b(\boldsymbol{x})$ = "Trial" probability of configuration $\boldsymbol{x}$.
- Average energy: $U = \sum_{\boldsymbol{x} \in A^n} b(\boldsymbol{x}) E(\boldsymbol{x})$.
- Entropy: $H = -\sum_{\boldsymbol{x} \in A^n} b(\boldsymbol{x}) \ln b(\boldsymbol{x})$.

A Statistical Physics Approach

- $b(\boldsymbol{x})$ = "Trial" probability of configuration $\boldsymbol{x}$.
- Average energy: $U = \sum_{\boldsymbol{x} \in A^n} b(\boldsymbol{x}) E(\boldsymbol{x})$.
- Entropy: $H = -\sum_{\boldsymbol{x} \in A^n} b(\boldsymbol{x}) \ln b(\boldsymbol{x})$.
- Variational free energy:

$$\widetilde{F}(b) = U - H.$$

A Famous Theorem from Statistical Mechanics

Theorem.

$$\widetilde{F}(b) \geq F,$$

with equality if and only if

$$b(\boldsymbol{x}) = B(\boldsymbol{x}) = \frac{1}{Z} e^{-E(\boldsymbol{x})},$$

the Boltzmann-Gibbs, *or equilibrium*, density.

Proof of the Famous Theorem

A simple calculation shows that

$$\widetilde{F}(b) = D(b \parallel B) + F.$$

Hence

$$\widetilde{F}(b) \geq F,$$

with equals iff $b = B$.

An Important Corollary

Corollary.

$$F = \min_{b(\boldsymbol{x})} \widetilde{F}(b)$$

$$B(\boldsymbol{x}) = \arg\min_{b(\boldsymbol{x})} \widetilde{F}(b).$$

An Important Corollary

Corollary.

$$F = \min_{b(\boldsymbol{x})} \widetilde{F}(b)$$

$$B(\boldsymbol{x}) = \arg\min_{b(\boldsymbol{x})} \widetilde{F}(b).$$

• Suggests a possible method for computing F (and $B(\boldsymbol{x})$), but as it stands, it's too complex, and anyway it doesn't yield the marginals $B_R(\boldsymbol{x})$ ...

A Solution Using Belief Propagation on a Partially Ordered Set

A Solution Using Belief Propagation on a Partially Ordered Set

(But What is a Partially Ordered Set?)

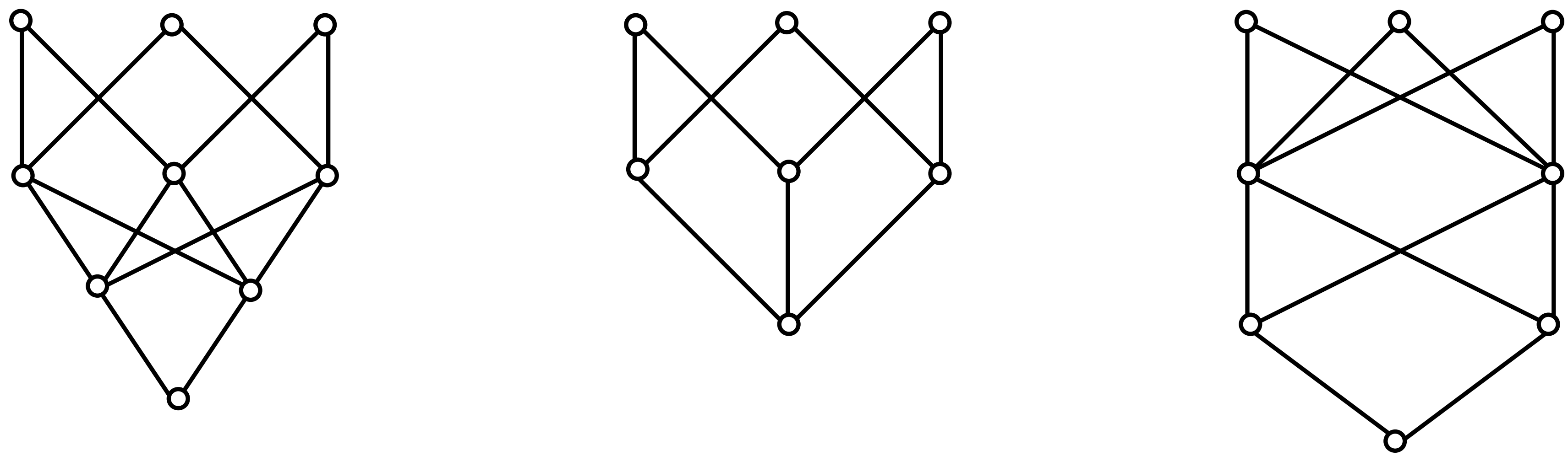
A Solution Using Belief Propagation on a Partially Ordered Set

(But What is a Partially Ordered Set?)

A finite *partially ordered set* is a finite set P together with a binary relation, denoted $\leq$, which satifies the following three axioms:

1. For all $\rho \in P$, $\rho \leq \rho$. *(reflexitivity)*
2. If $\rho \leq \sigma$ and $\sigma \leq \rho$, then $\rho = \sigma$. *(antisymmetry)*
3. If $\rho \leq \sigma$ and $\sigma \leq \tau$, then $\rho \leq \tau$. *(transitivity)*

Hasse Diagrams for Three Posets



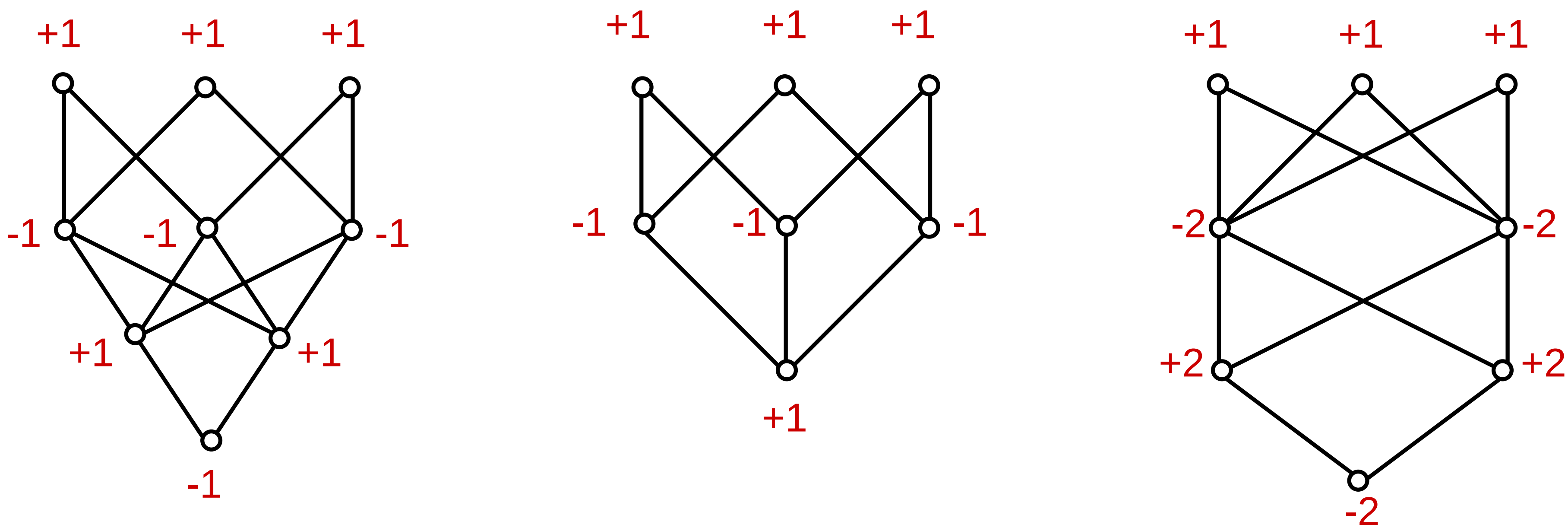
Overcounting Numbers for Posets

We assign an *overcounting number* $\phi(\rho)$ to each $\rho \in P$, such that

$$(1) \qquad \sum_{\rho:\rho \geq \sigma} \phi(\rho) = 1, \text{ for all } \sigma \in P.$$

The overcounting numbers $\phi(\rho)$ are integers and are determined uniquely by (1).

Some Overcounting Numbers



How to Distribute the Local Potentials in a Given Poset P.

Step 1: Assign each variable x_i to one or more elements of P. If $P_i = \{\rho \in P : x_i \text{ is assigned to } \rho\}$, then we require:

- P_i is connected;
- P_i is closed under $\geq$, i.e., if x_i is assigned to ρ it is also ssigned to all of ρ's "superiors;"
- $\phi(P_i) = 1$, i.e., the net number of appearances of x_i is 1.

We denote by $D(\rho)$ (the local domain at ρ) the set of variables which are assigned to ρ.

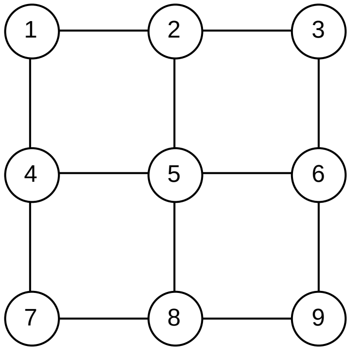
How to Distribute the Local Potentials in a Given Poset P.

Step 2: Assign each local potential $\alpha_R(\boldsymbol{x}_R)$ to one or more elements of P. If $P_R = \{\rho \in P : \alpha_R(\boldsymbol{x}_R) \text{ is assigned to } \rho\}$, then we require:

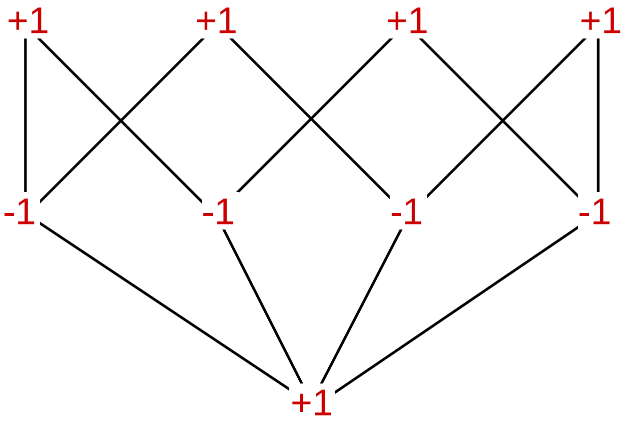
- $R \subseteq D(\rho)$ for all $\rho \in P_R$, i.e., the local domain at ρ supports $\alpha_R(\boldsymbol{x}_R)$;
- P_R is connected;
- P_R is closed under $\geq$, i.e., if α_R is known to ρ it is also known to all of ρ's superiors;
- $\phi(P_R) = 1$, i.e., the net number of appearances of $\alpha_R(\boldsymbol{x}_R)$ is 1.

Example: “Generalized Belief Propagation” (YF&W)

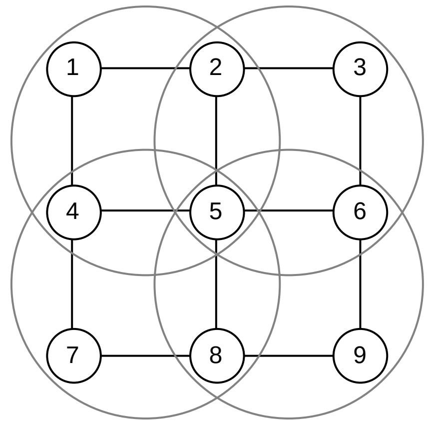
- $\mathcal{R} = \{\{1,2\},\{2,3\},\ldots,\{8,9\}\}$:

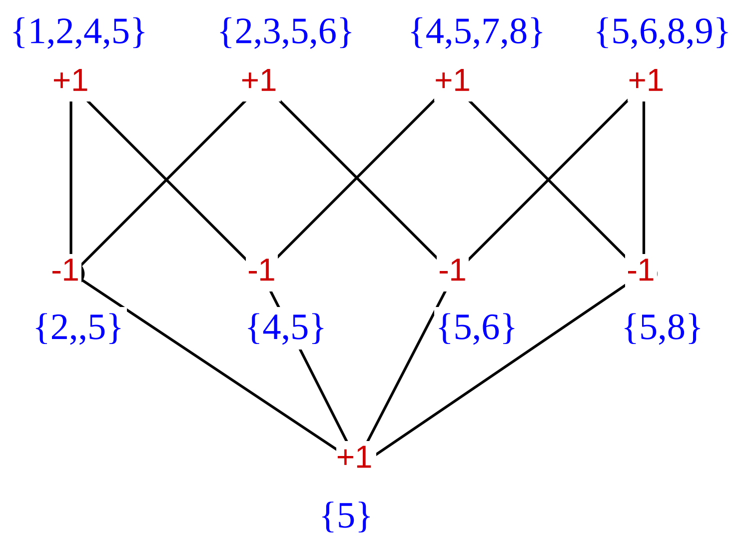


- Here is the poset:



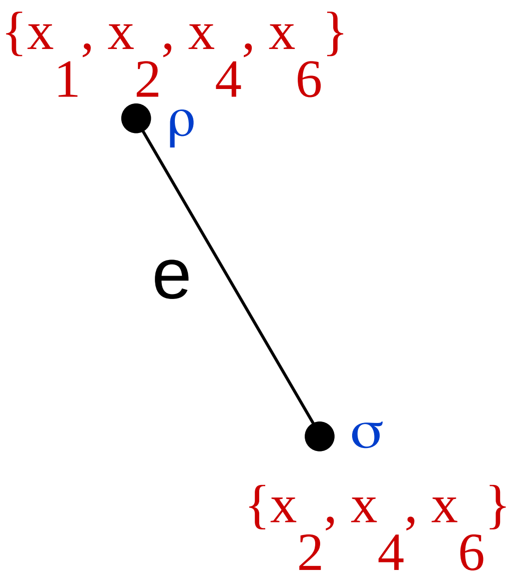
The Trick is to Cluster the Variables





The PBP Algorithm: The Messages.

- Throughout the algorithm, each edge $e = (\rho, \sigma)$ of the Hasse diagram carries a "message" m_e.
- The message m_e on the edge e is a function on the domain of σ: $m_e = m_e(\boldsymbol{x}_\sigma)$.
- **Example:**



(x_2, x_4, x_6)	$m_e(x_2.x_4, x_6)$
$(0, 0, 0)$	1
$(0, 0, 1)$	0
$\vdots$	$\vdots$
$(1, 1, 1)$	0

The PBP Algorithm: Calculating Beliefs.

- For a given set of messages $\{m_e : e \in E\}$, we define the *belief at* ρ as the following probability density on the domain at ρ:

$$b_\rho(\boldsymbol{x}_\rho) \propto \alpha_\rho(\boldsymbol{x}_\rho) \prod_{e \in E(\rho)} m_e(\boldsymbol{x}_{\mathrm{fin}\, e}),$$

where the normalization is such that $\sum_{\boldsymbol{x}_\rho} b_\rho(\boldsymbol{x}_\rho) = 1$.

The PBP Algorithm: Calculating Beliefs.

• For a given set of messages $\{m_e : e \in E\}$, we define the *belief at* ρ as the following probability density on the domain at ρ:

$$b_\rho(\boldsymbol{x}_\rho) \propto \alpha_\rho(\boldsymbol{x}_\rho) \prod_{e \in E(\rho)} m_e(\boldsymbol{x}_{\mathrm{fin}\, e}),$$

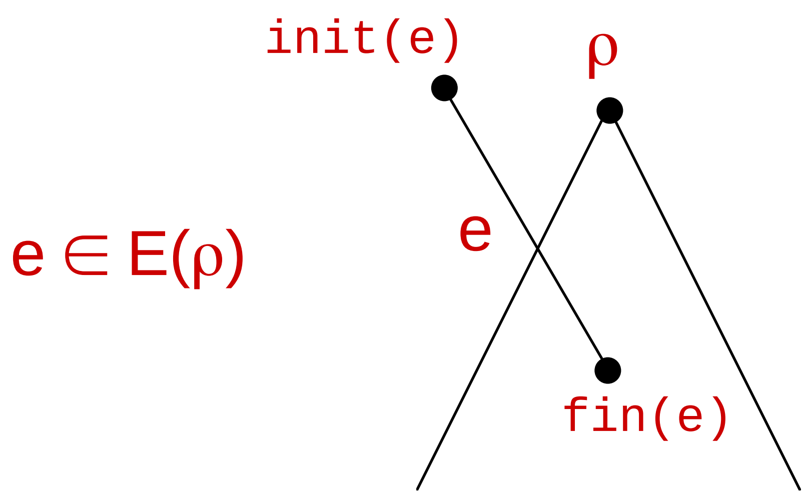
where the normalization is such that $\sum_{\boldsymbol{x}_\rho} b_\rho(\boldsymbol{x}_\rho) = 1$.

• Here $\alpha_\rho(\boldsymbol{x}_\rho)$ is the local potential at ρ, and $E(\rho)$ represents the set of messages which are "fused" at ρ.

The Messages that are "Fused" at ρ

$$b_\rho(\boldsymbol{x}_\rho) \propto \alpha_\rho(\boldsymbol{x}_\rho) \prod_{e \in E(\rho)} m_e(\boldsymbol{x}_{\mathrm{fin}\, e}),$$

- The messages in $E(\rho)$ are those that originate outside ρ's "field of view" but terminate inside it.



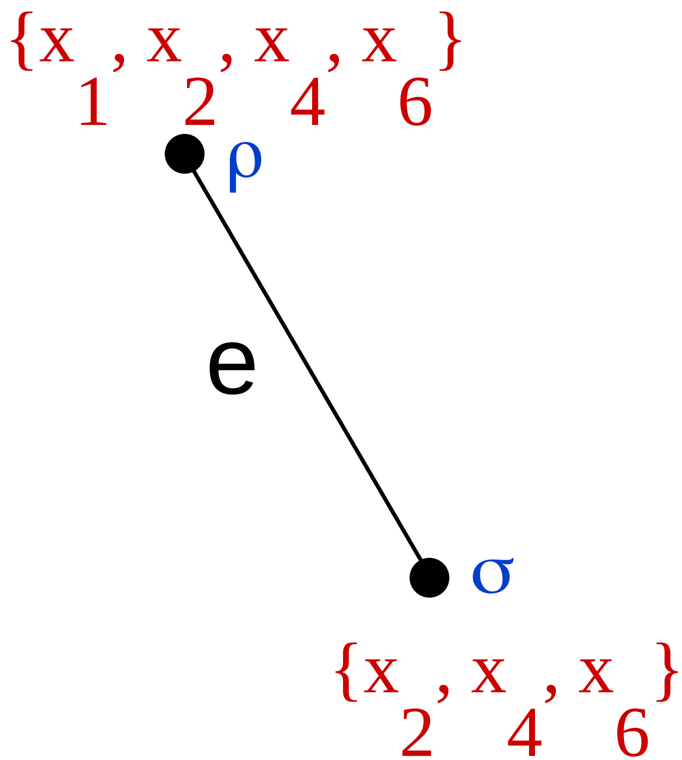
The PBP Algorithm: Updating the Messages.

• An edge $e = (\rho, \sigma)$ is said to be *consistent* with respect to a given set $\{m_e : E \in E\}$ of messages if

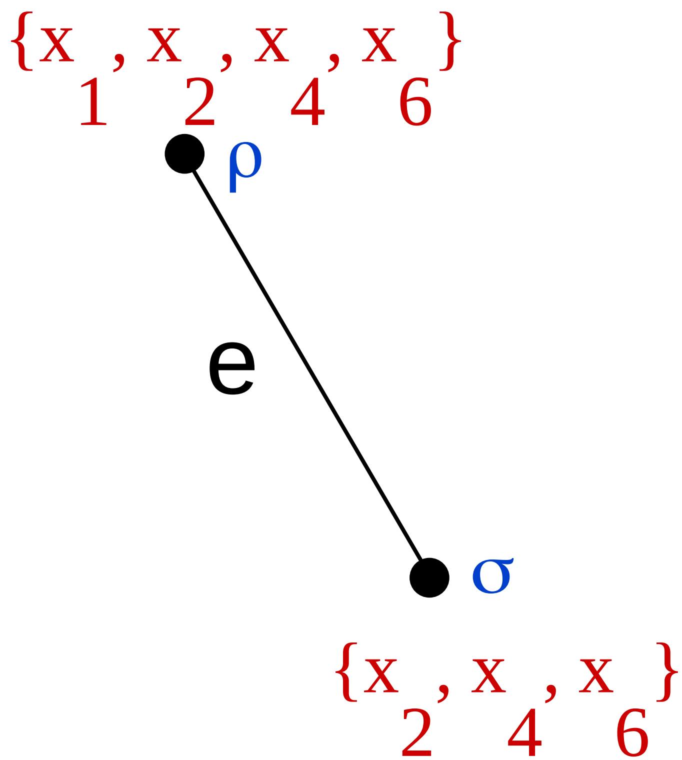
$$\sum_{\boldsymbol{x}_\rho \setminus \boldsymbol{x}_\sigma} b_\rho(\boldsymbol{x}_\rho) = b_\sigma(\boldsymbol{x}_\sigma) \qquad \text{for all } \boldsymbol{x}_\sigma \in A^{D(\sigma)}.$$

In words, this says that the belief at ρ in $\boldsymbol{x}_\sigma$, obtained by marginalization, agrees with the belief at σ in $\boldsymbol{x}_\sigma$.

Example of Edge Consistency



Example of Edge Consistency



- $e = (\rho, \sigma)$ is consistent if

$$\sum_{x_1 \in A} b_\rho(x_1, x_2, x_4, x_6) = b_\sigma(x_2, x_4, x_6)$$

The PBP Algorithm: The Update Rule

• When the message m_e is updated, it is adjusted so that the edge e becomes *consistent*. Explicitly,

$$m_e(\boldsymbol{x}_\sigma) \propto \frac{\sum_{\boldsymbol{x}_\rho \setminus \boldsymbol{x}_\sigma} \left(\alpha_{\rho\setminus\sigma}(\boldsymbol{x}_\rho) \prod_{g \in E(\rho)\setminus E(\sigma)} m_g(\boldsymbol{x}_{\mathrm{fin}(g)}) \right)}{\prod_{f \in E(\sigma)\setminus\{E(\rho)\cup e\}} m_f(\boldsymbol{x}_{\mathrm{fin}(f)})}.$$

The PBP Algorithm: The Update Rule

• When the message m_e is updated, it is adjusted so that the edge e becomes *consistent*. Explicitly,

$$m_e(\boldsymbol{x}_\sigma) \propto \frac{\sum_{\boldsymbol{x}_\rho \backslash \boldsymbol{x}_\sigma} \left(\alpha_{\rho\backslash\sigma}(\boldsymbol{x}_\rho) \prod_{g \in E(\rho)\backslash E(\sigma)} m_g(\boldsymbol{x}_{\mathrm{fin}(g)}) \right)}{\prod_{f \in E(\sigma)\backslash\{E(\rho)\cup e\}} m_f(\boldsymbol{x}_{\mathrm{fin}(f)})}.$$

• The PBP algorithm proceeds by updating messages according to this rule. The hope is that the messages will converge to a fixed point whose associated beliefs are good approximations to the desired marginals, i.e.,

$$b_\rho(\boldsymbol{x}_\rho) \approx B_\rho(\boldsymbol{x}_\rho),$$

where $B(\boldsymbol{x})$ is the global or “Boltzmann” density.

How Well Does it Work?

How Well Does it Work?

- A series of experiments:

How Well Does it Work?

- A series of experiments:
- Local domains:

$$\mathcal{R} = \{\{1,2,3\},\{1,3,4\},\{2,3,5\},\{3,4,5\}\}.$$

How Well Does it Work?

- A series of experiments:
- Local domains:

$$\mathcal{R} = \{\{1,2,3\},\{1,3,4\},\{2,3,5\},\{3,4,5\}\}.$$

- Local potentials: $\alpha_1(x_1,x_2,x_3)$, $\alpha_2(x_1,x_3,x_4)$, $\alpha_3(x_2,x_3,x_5)$, and $\alpha_4(x_3,x_4,x_5)$ *selected at random* from the 8-simplex.

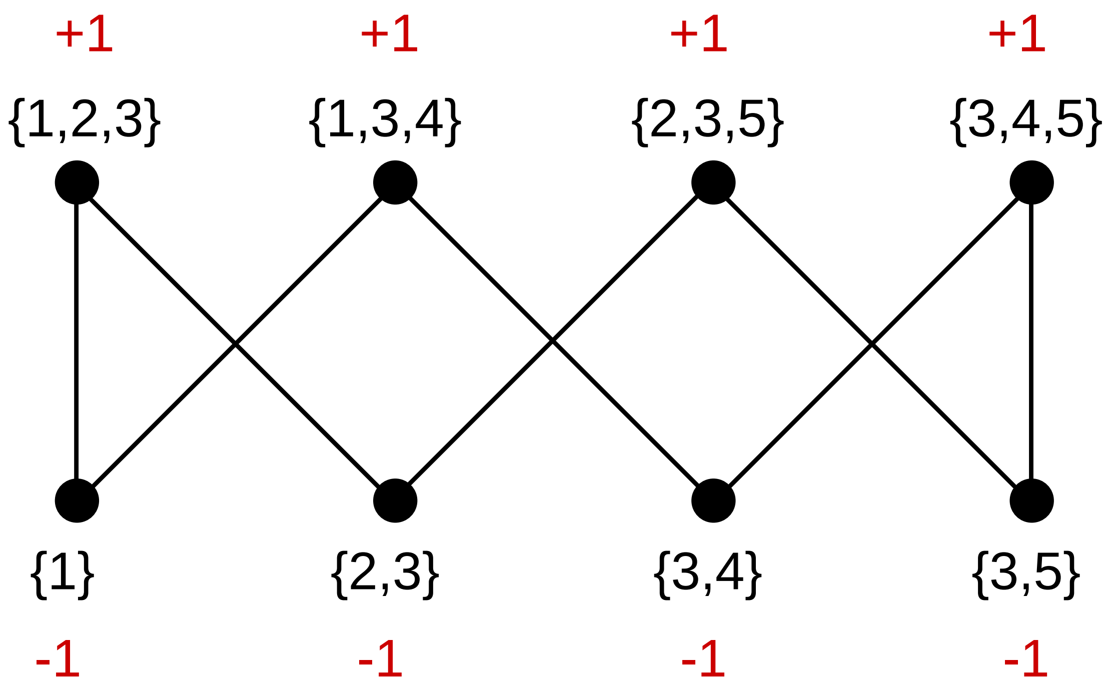
How Well Does it Work?

- A series of experiments:
- Local domains:

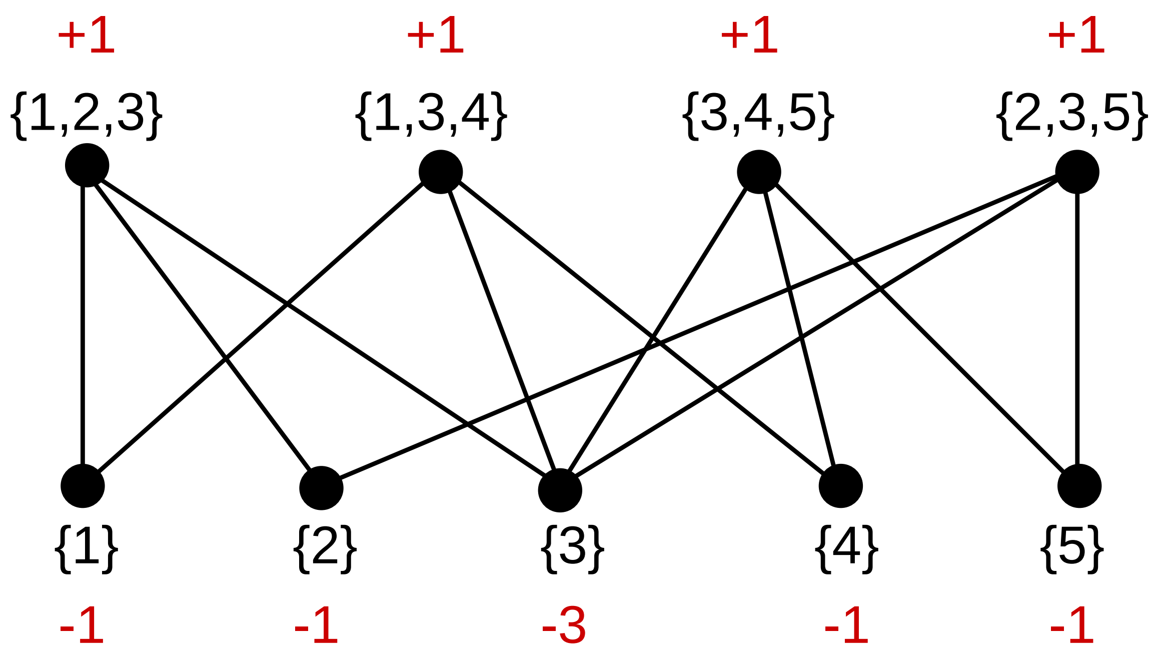
$$\mathcal{R} = \{\{1,2,3\},\{1,3,4\},\{2,3,5\},\{3,4,5\}\}.$$

- Local potentials: $\alpha_1(x_1, x_2, x_3)$, $\alpha_2(x_1, x_3, x_4)$, $\alpha_3(x_2, x_3, x_5)$, and $\alpha_4(x_3, x_4, x_5)$ *selected at random* from the 8-simplex.
- There are many posets that support this choice of domains; we will investigate only three.

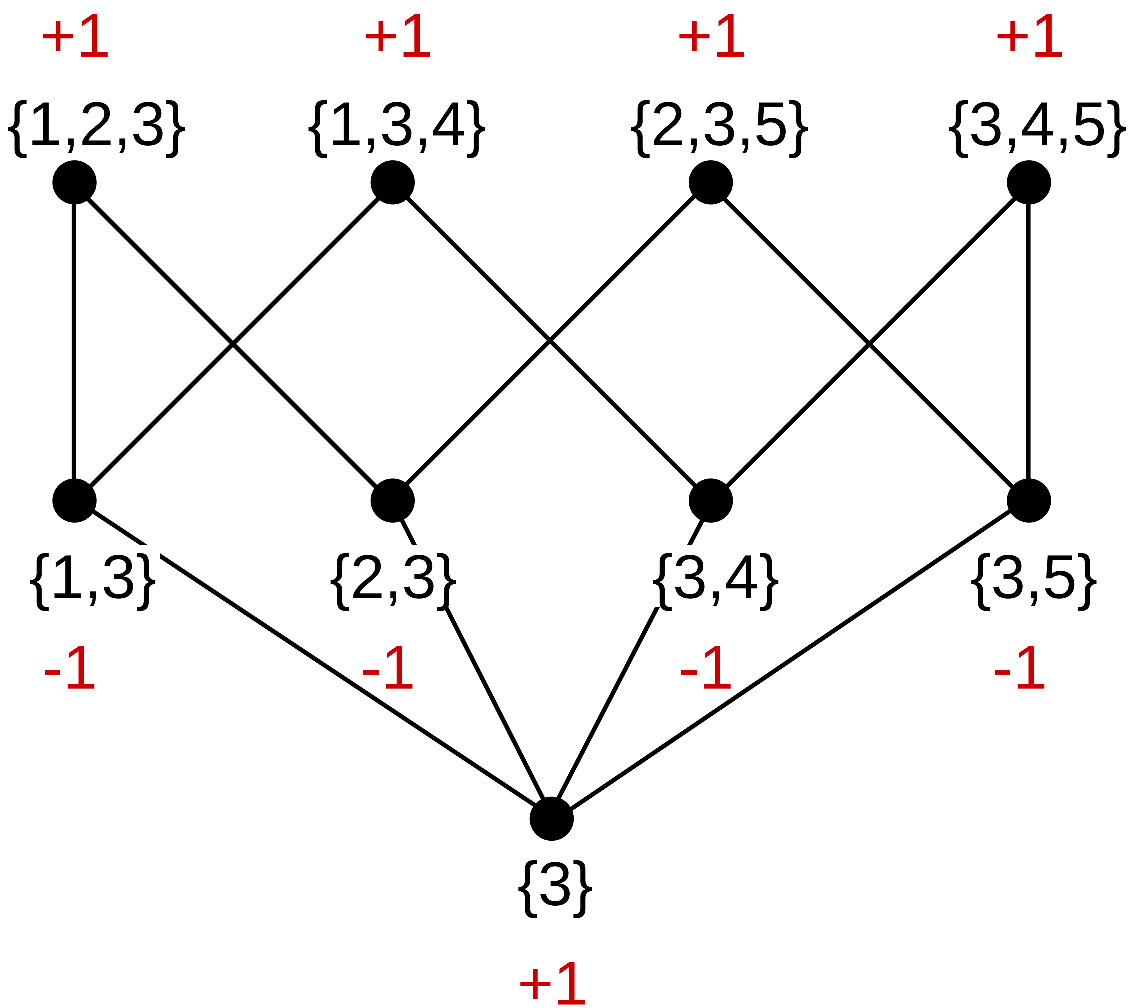
Poset Number One (Junction Graph Construction)



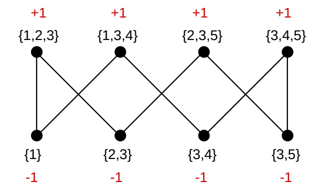
Poset Number Two (Factor Graph Construction)

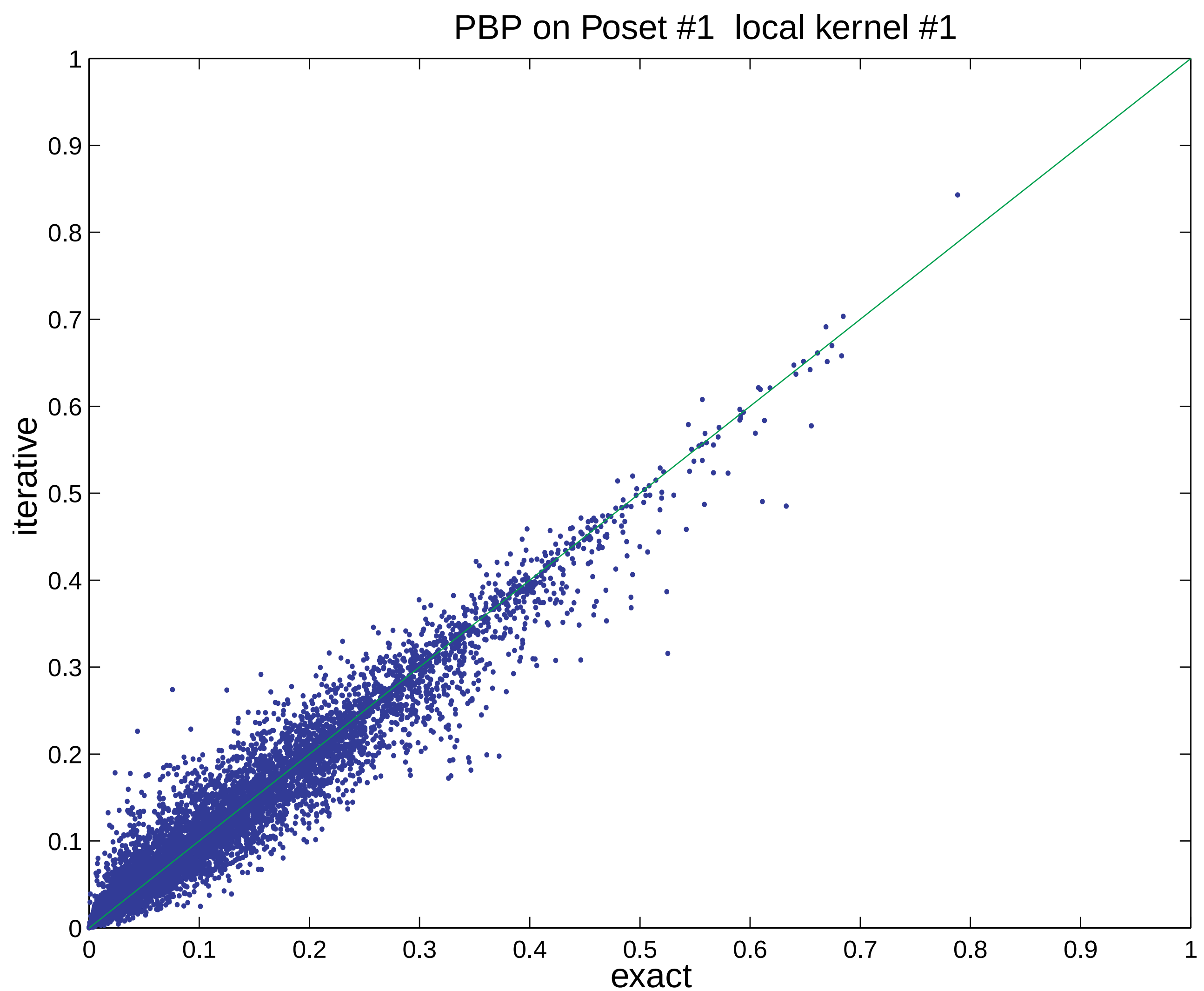


Poset Number Three (Cluster Variational Method)

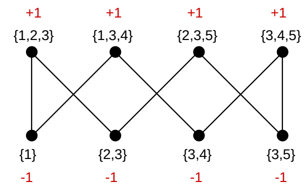


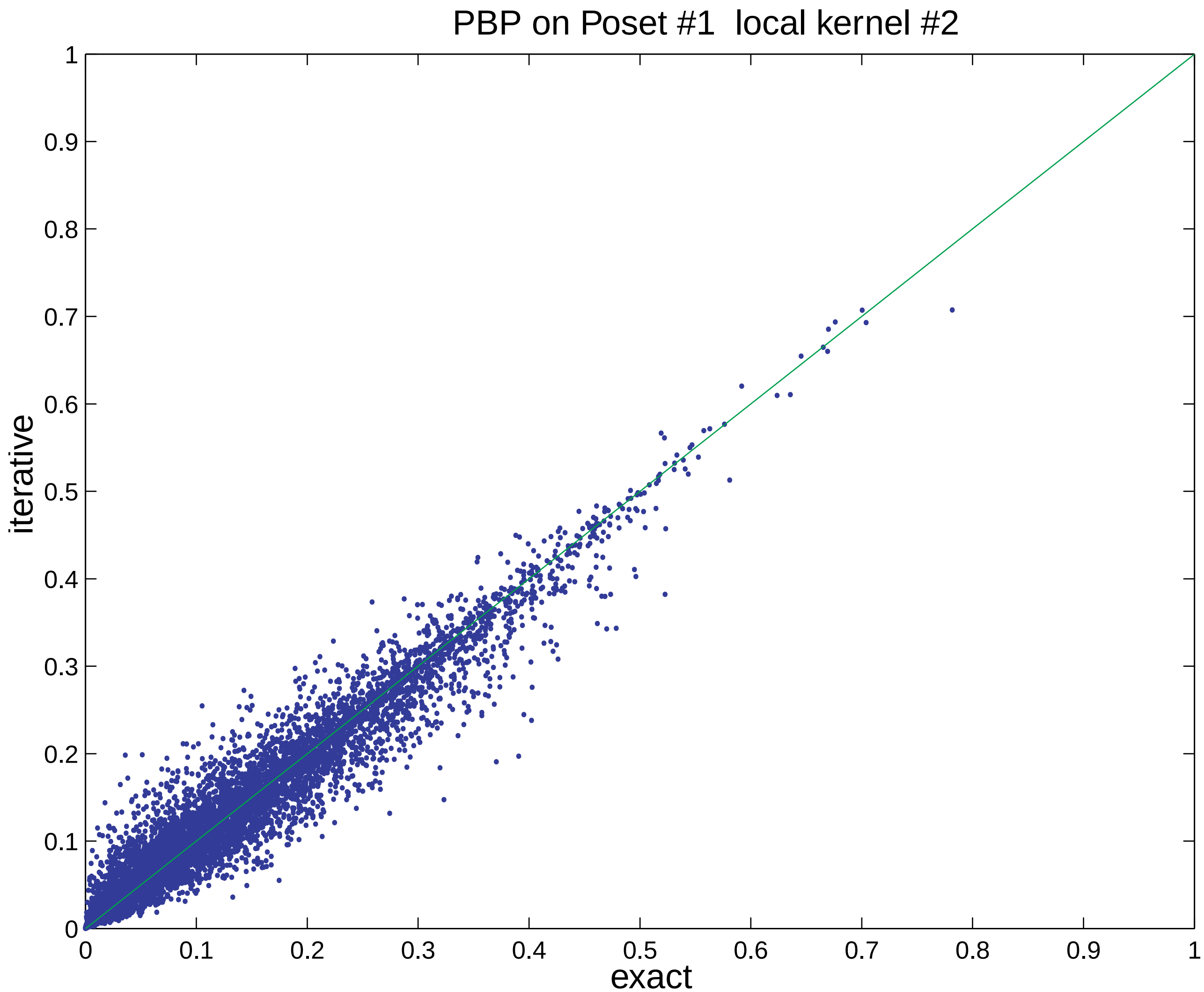
Poset Number One



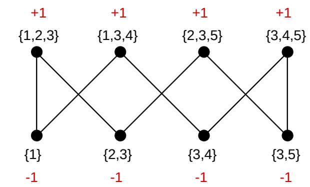


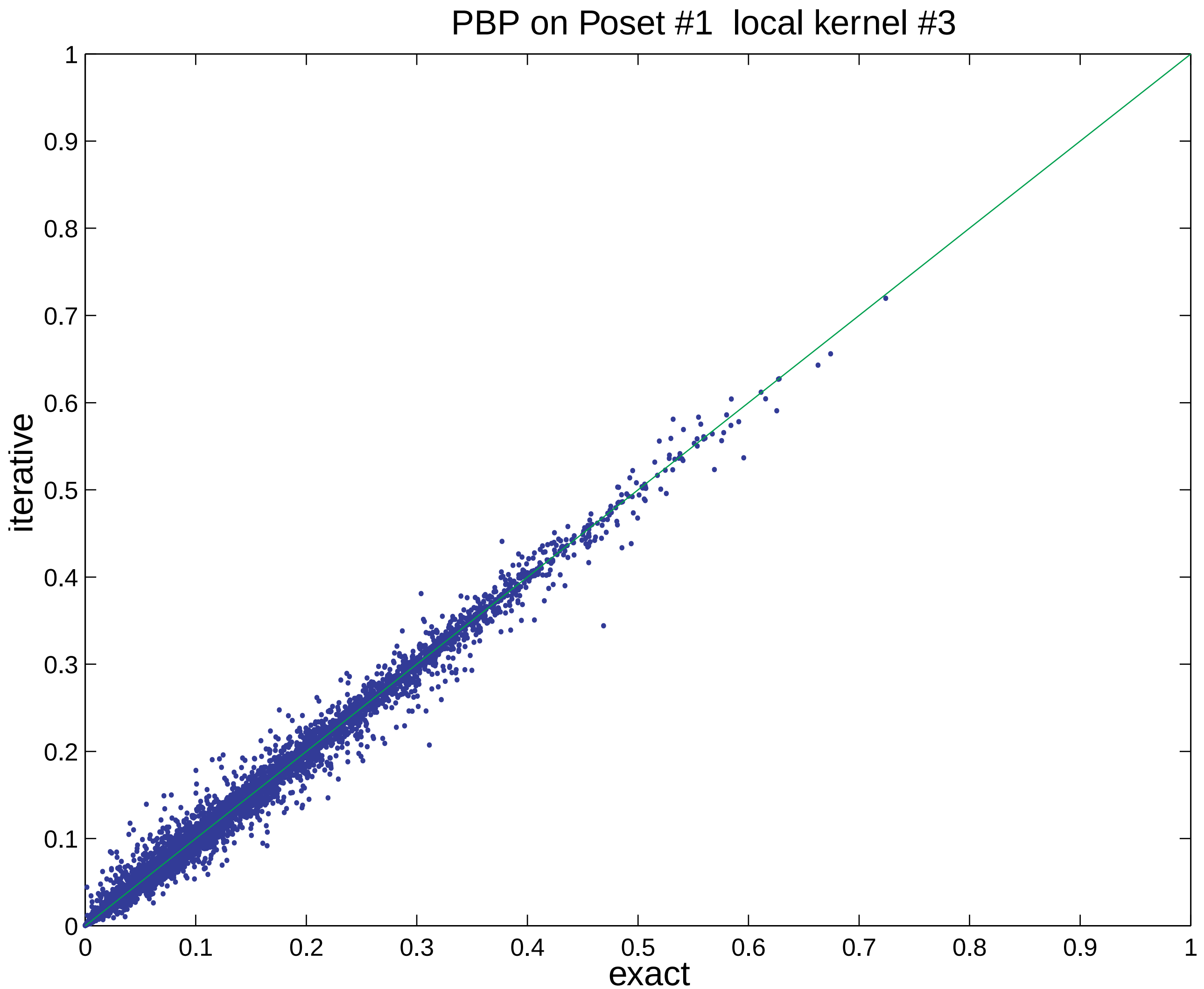
Poset Number One



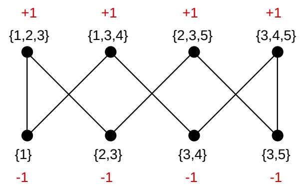


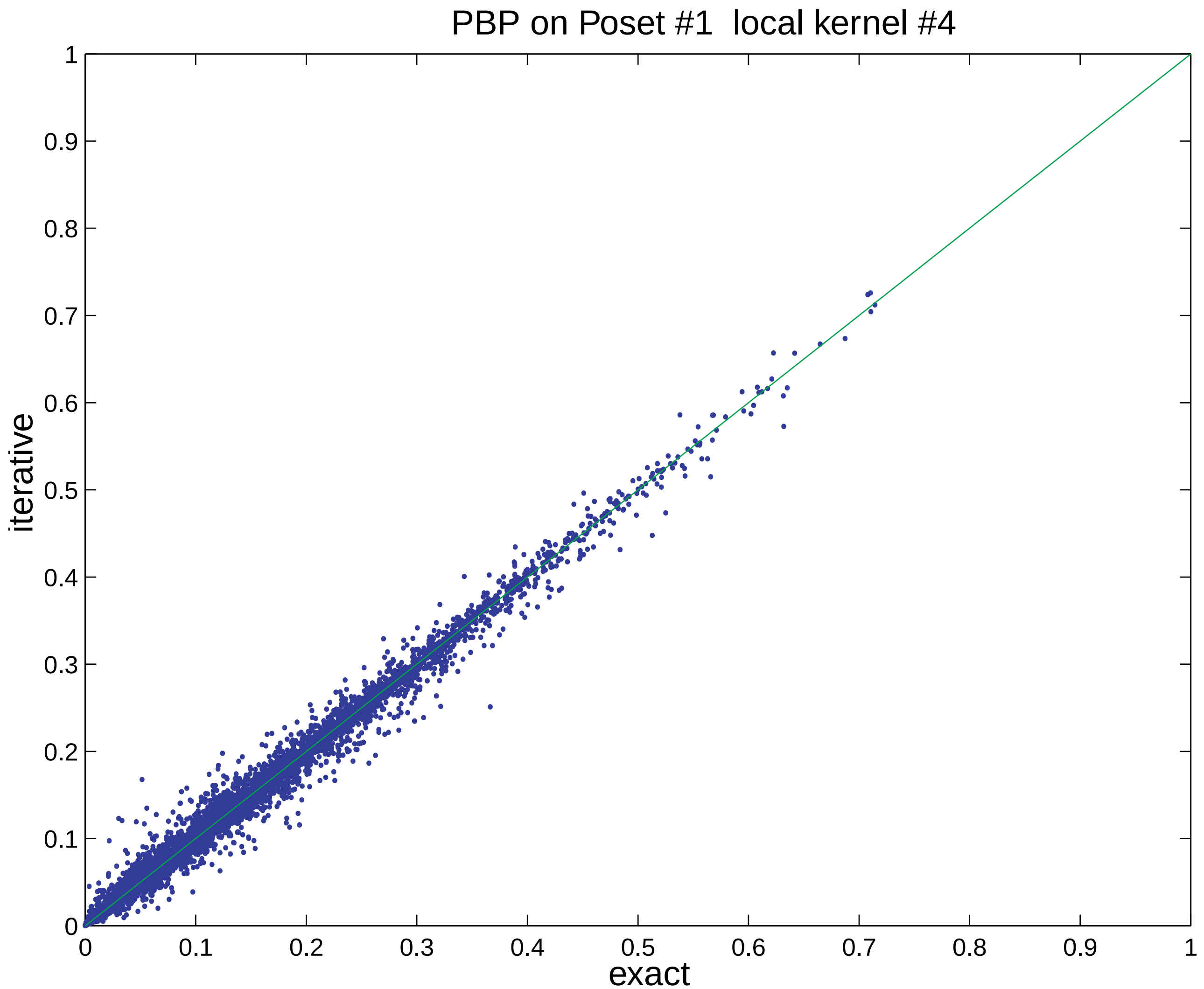
Poset Number One



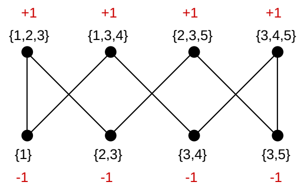


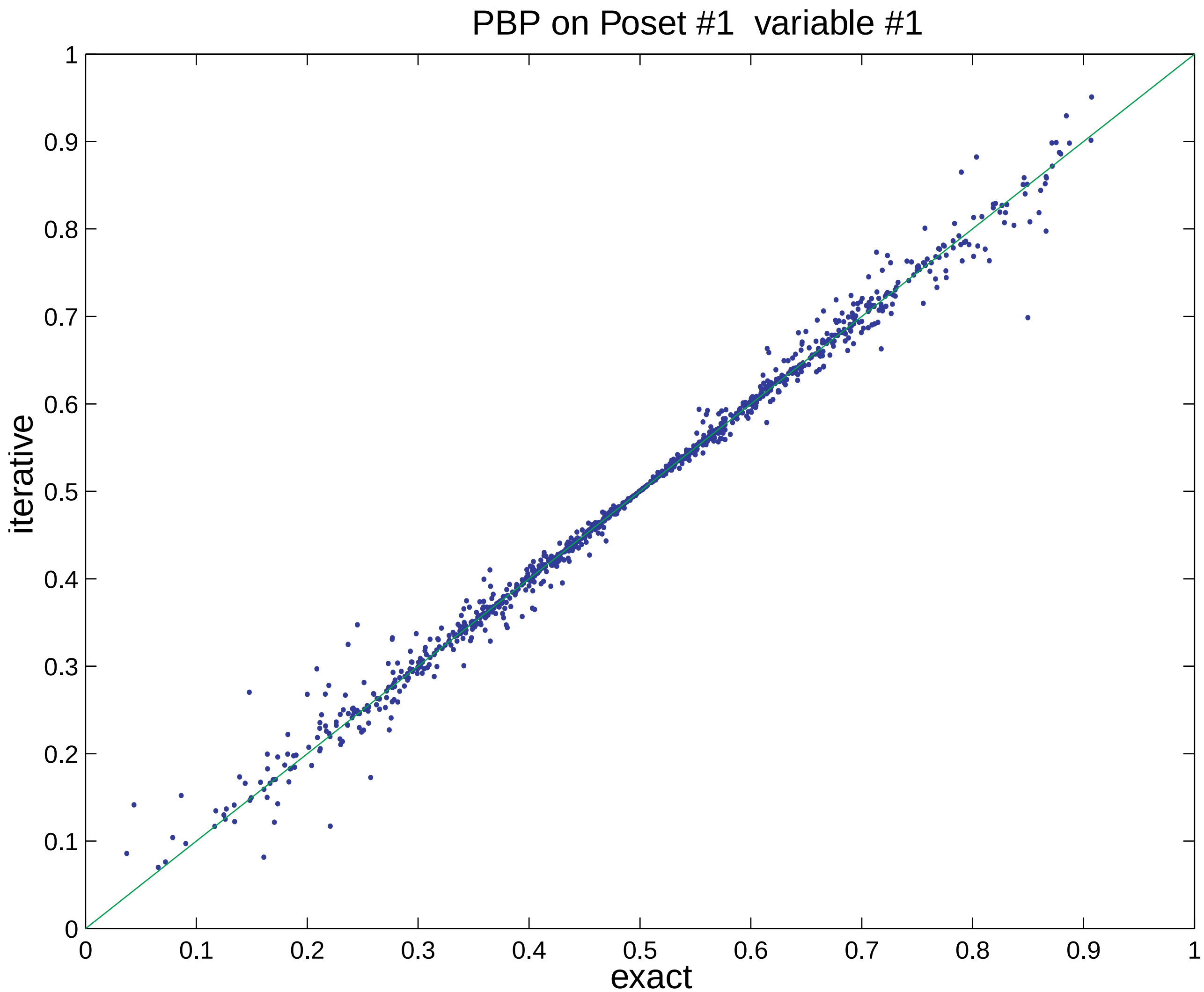
Poset Number One



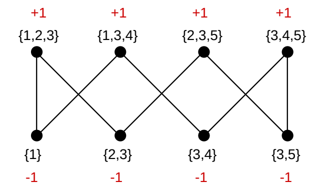


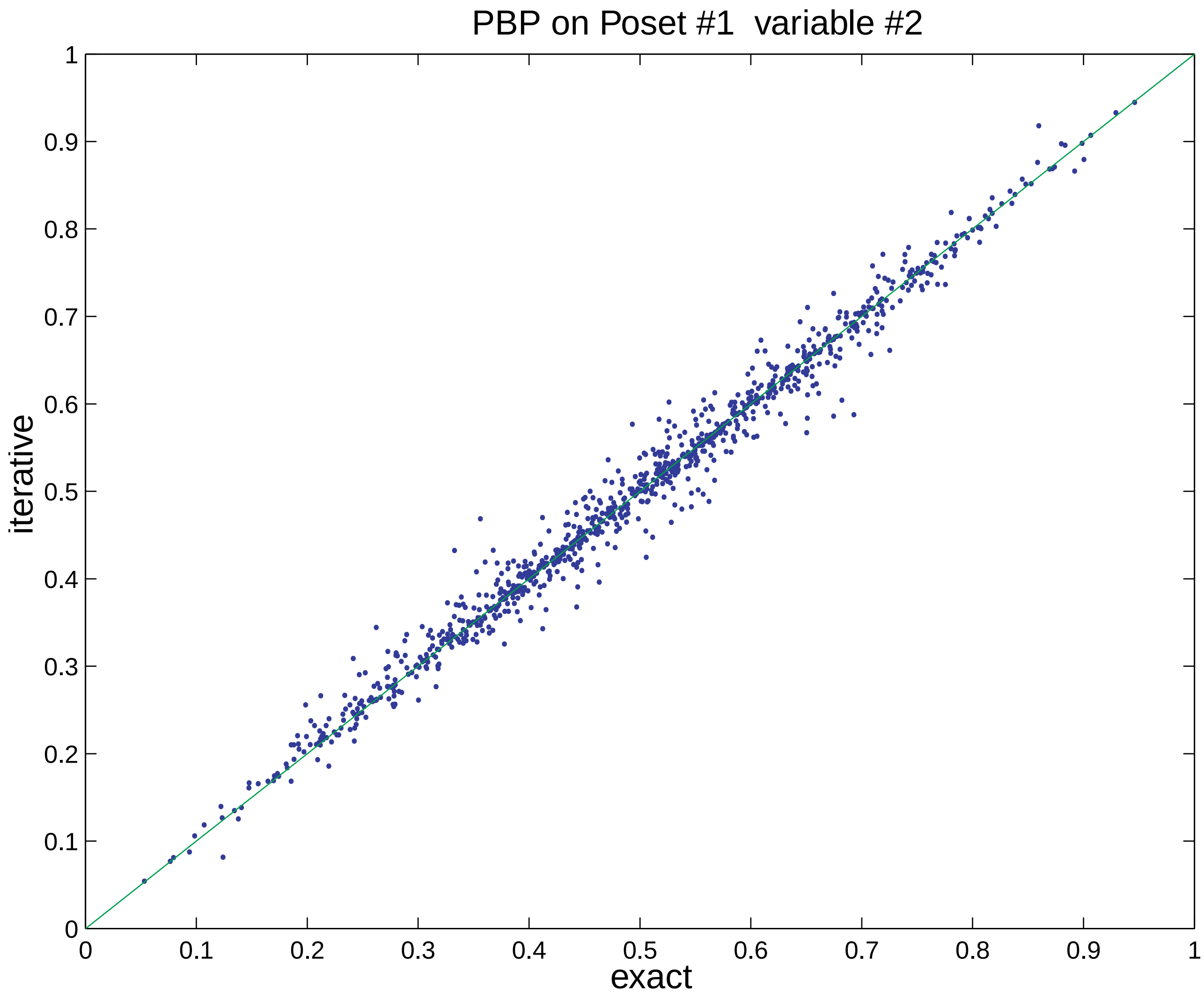
Poset Number One



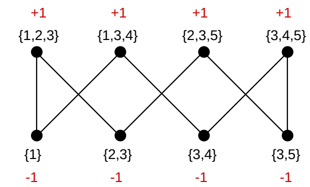


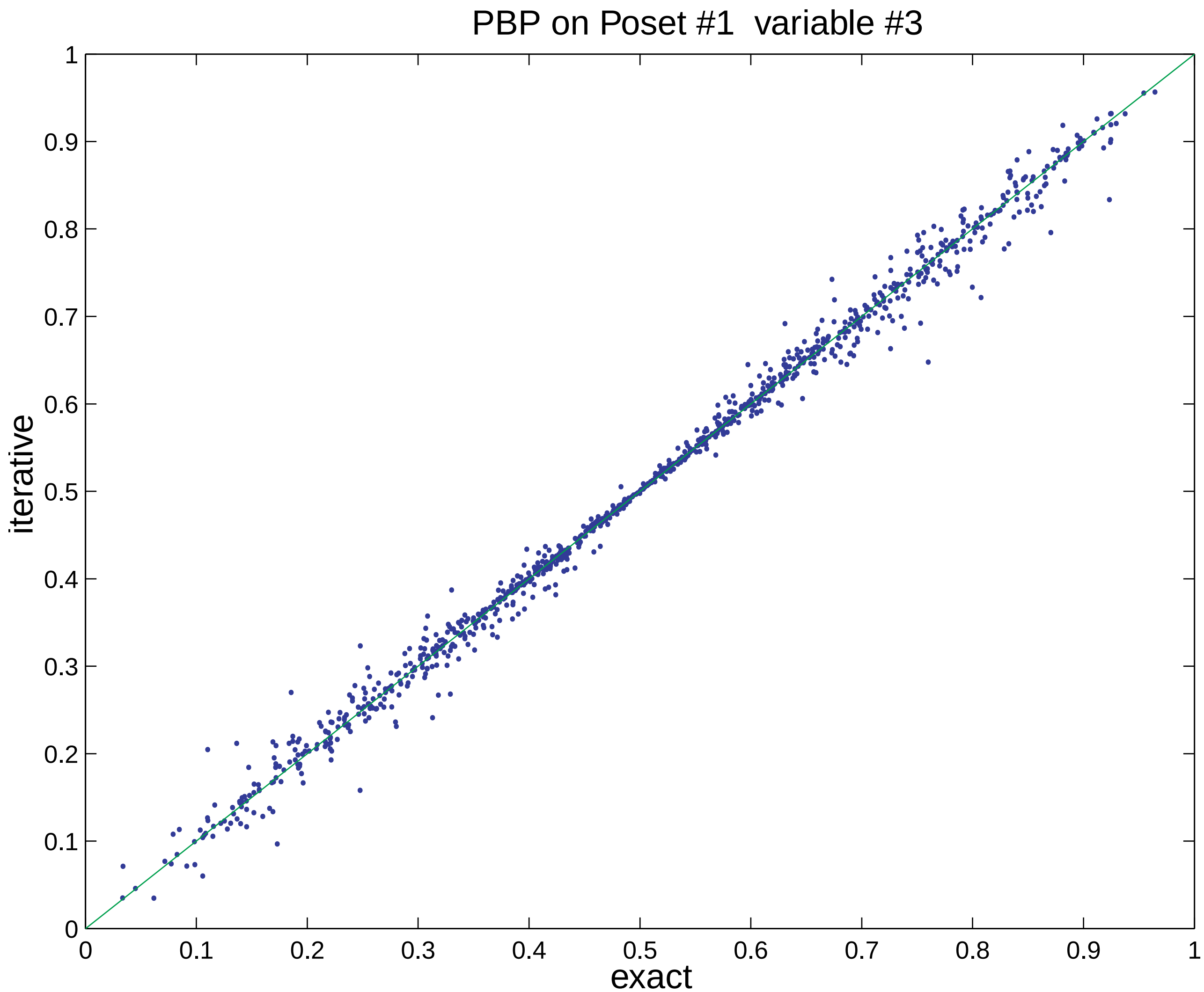
Poset Number One



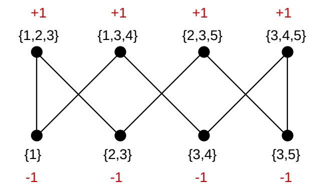


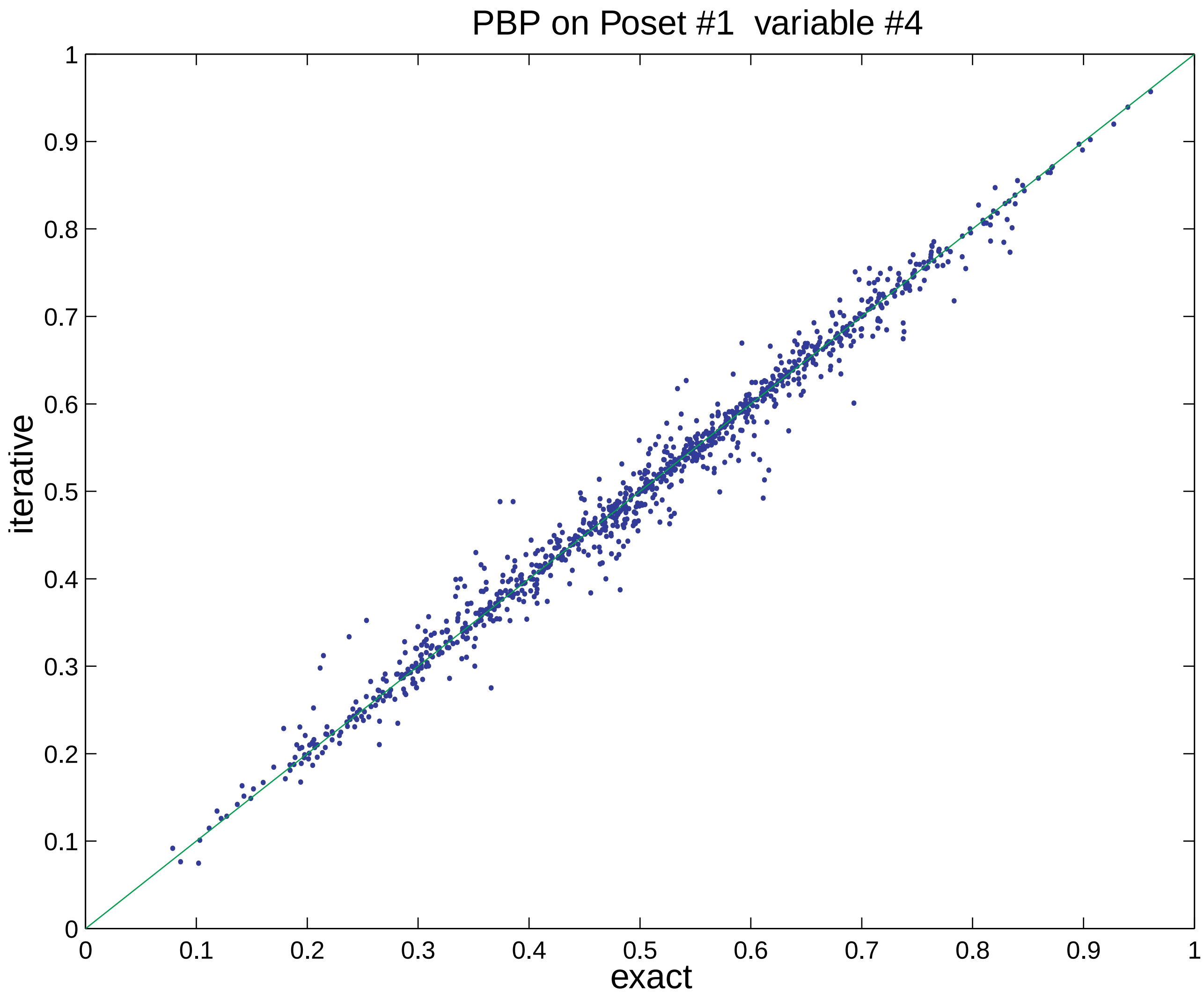
Poset Number One



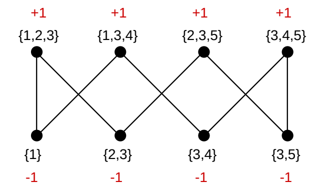


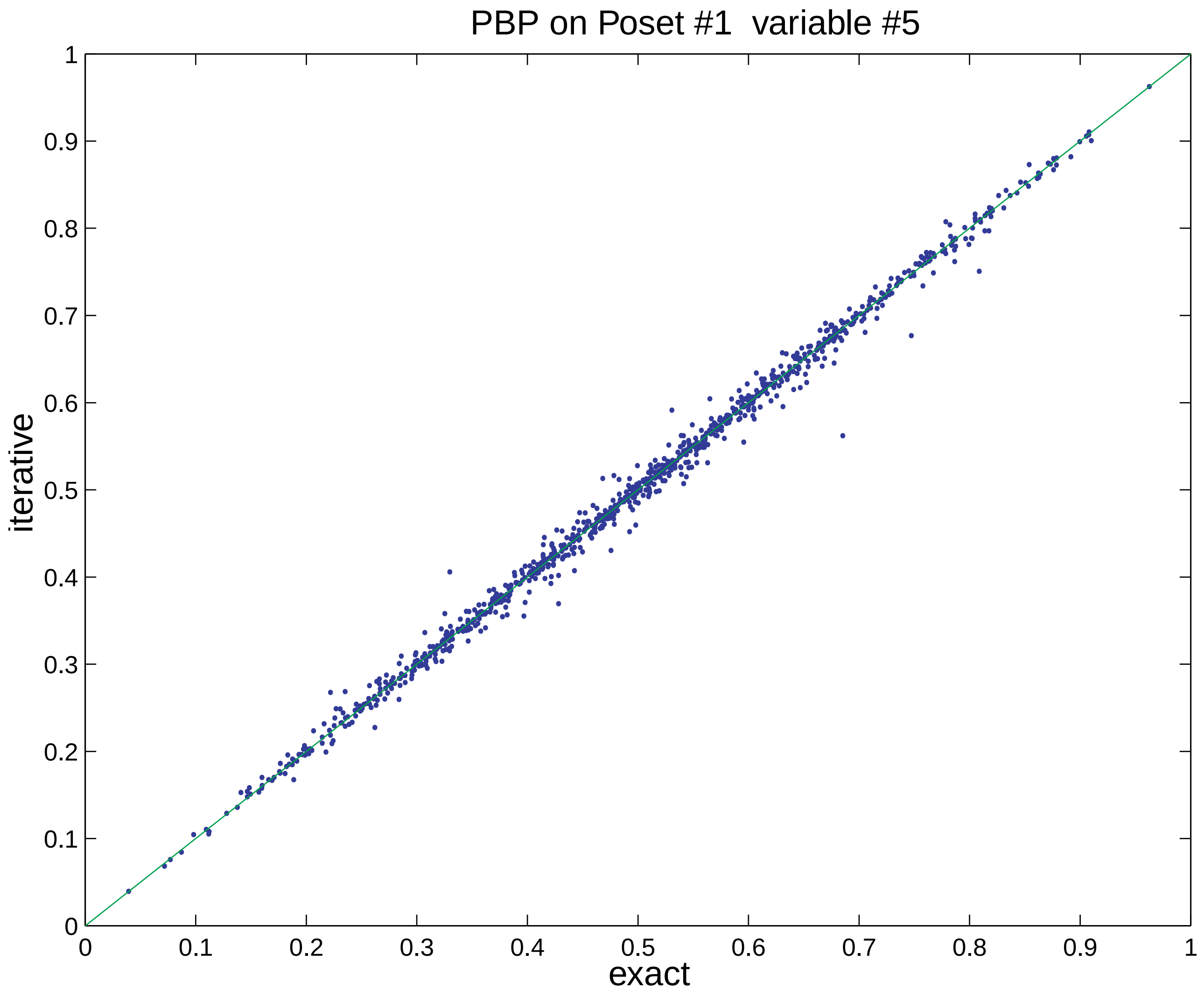
Poset Number One



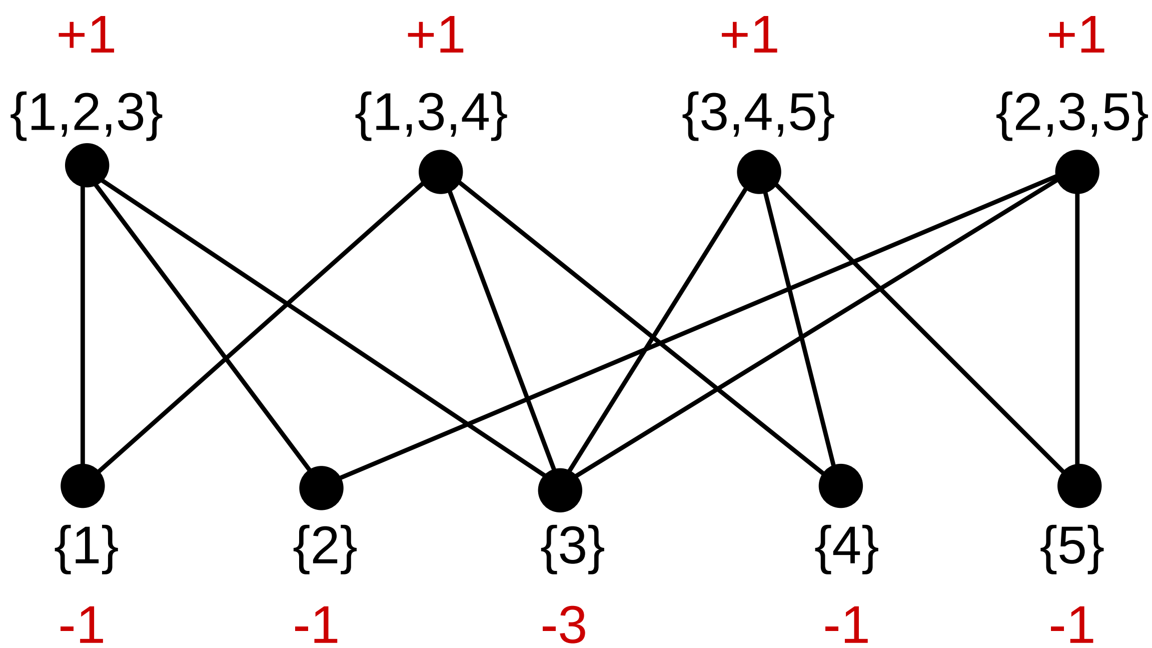


Poset Number One

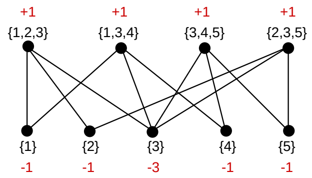


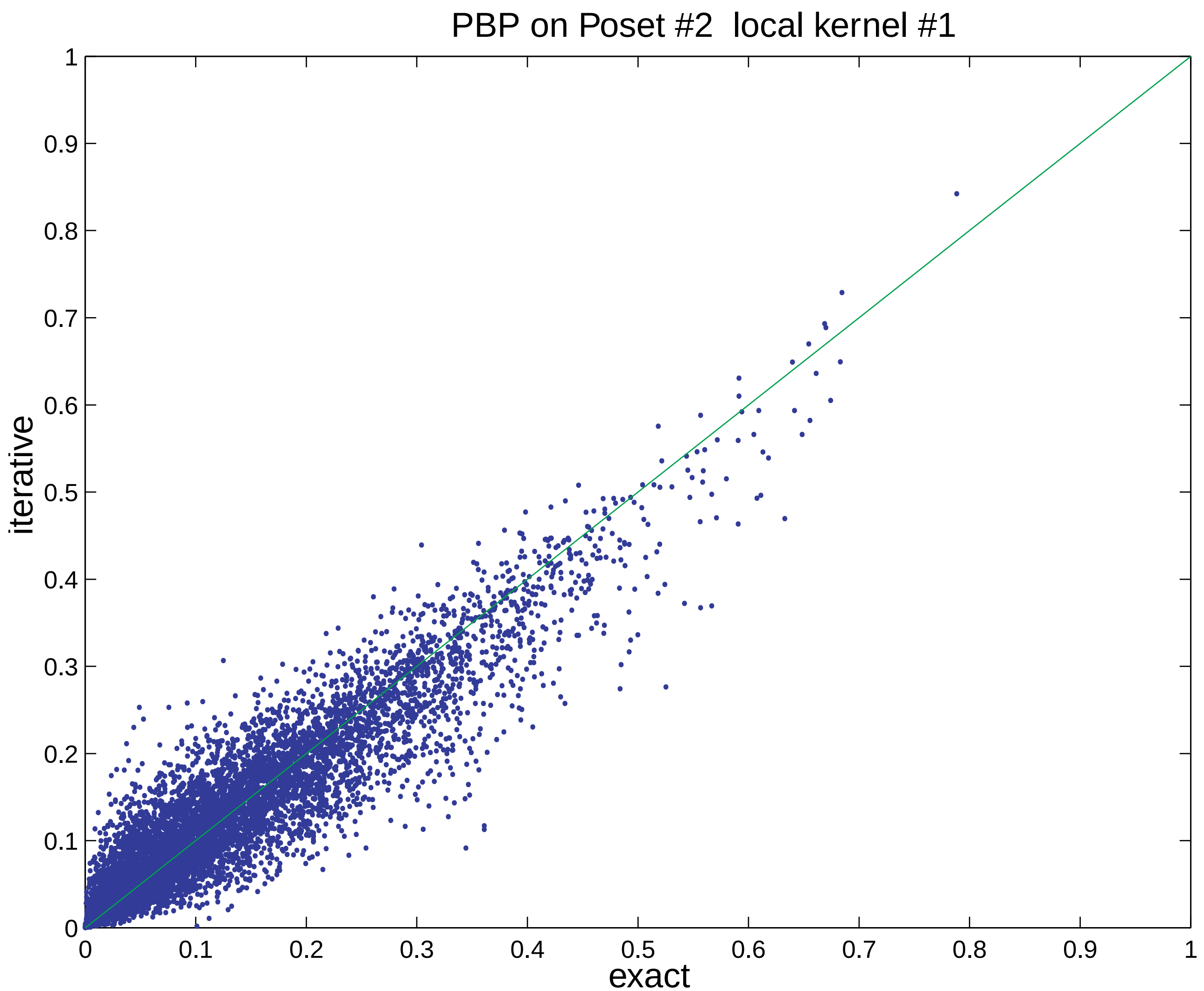


Poset Number Two (Factor Graph Construction)

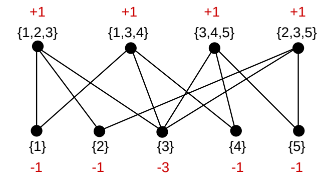


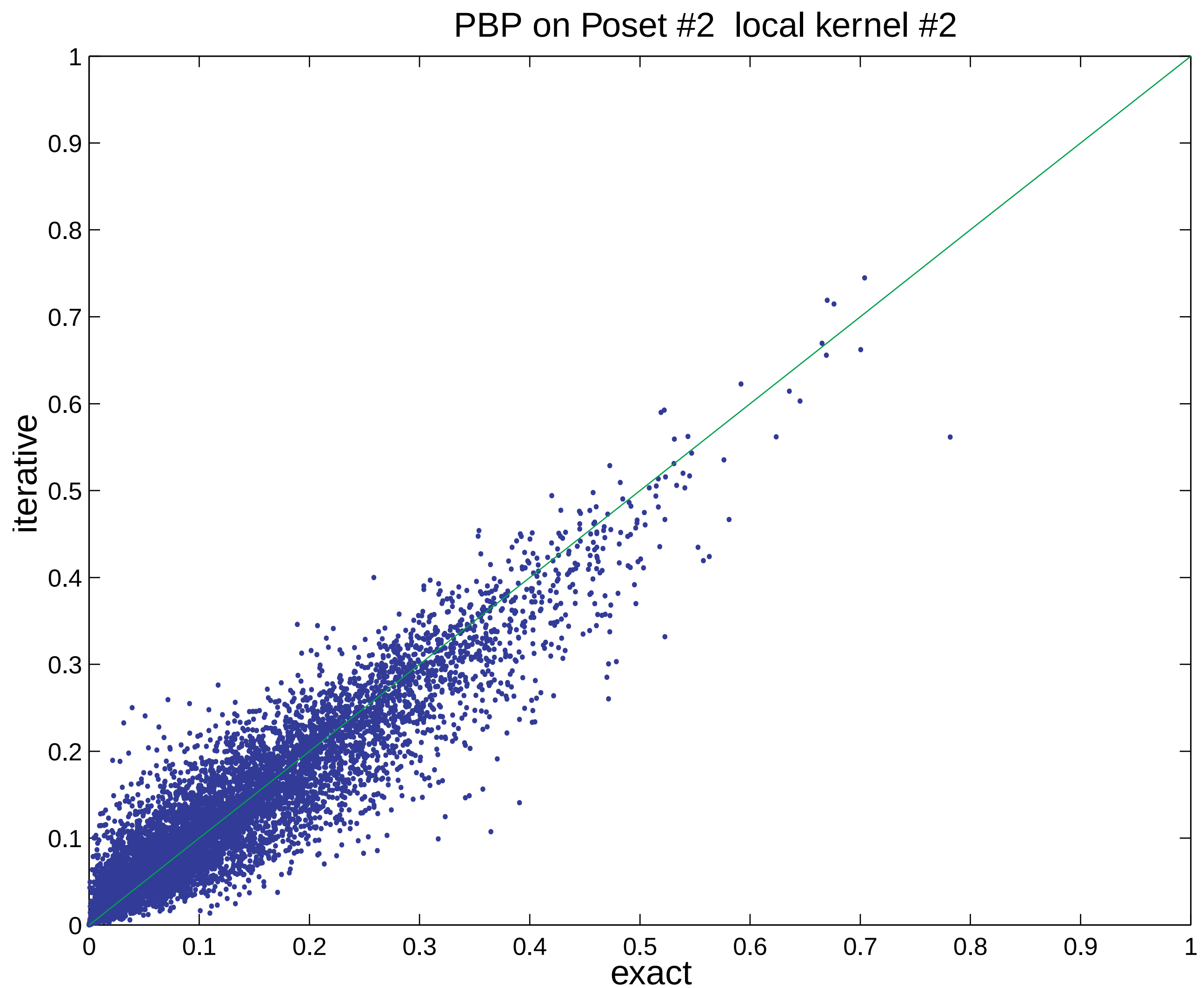
Poset Number Two



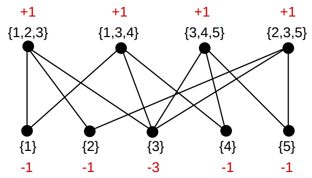


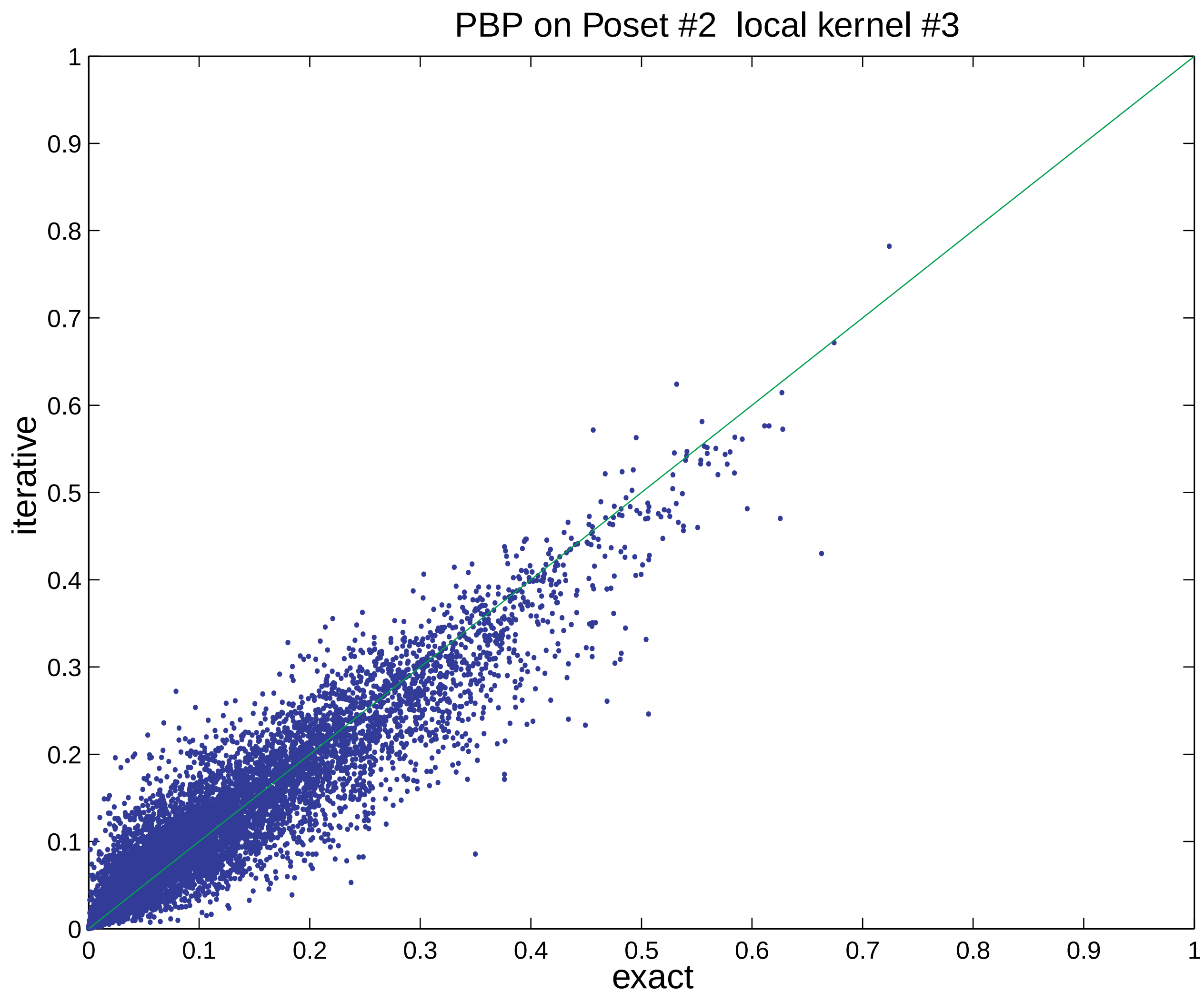
Poset Number Two



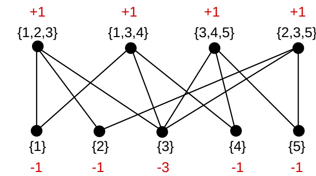


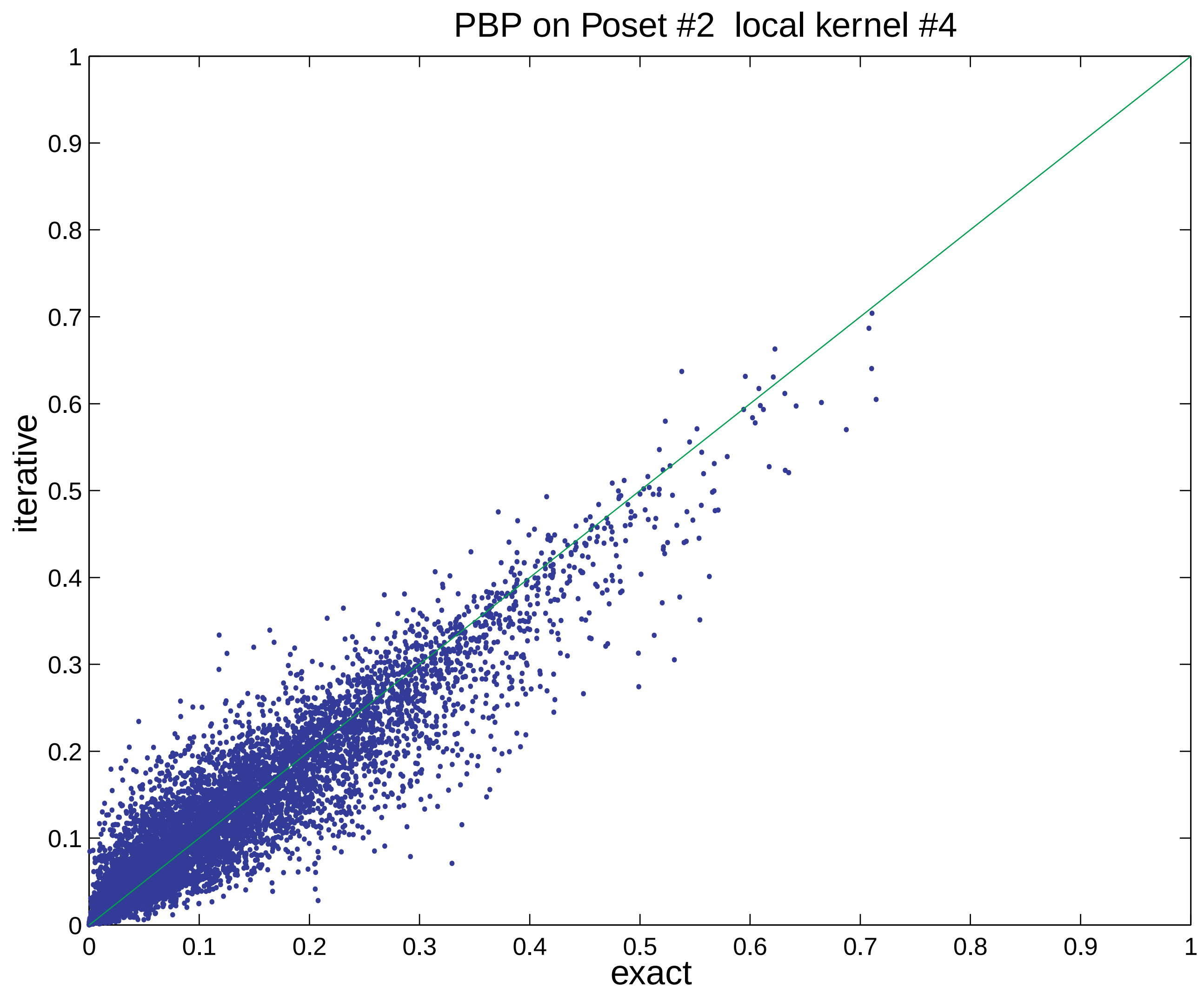
Poset Number Two



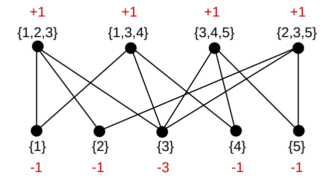


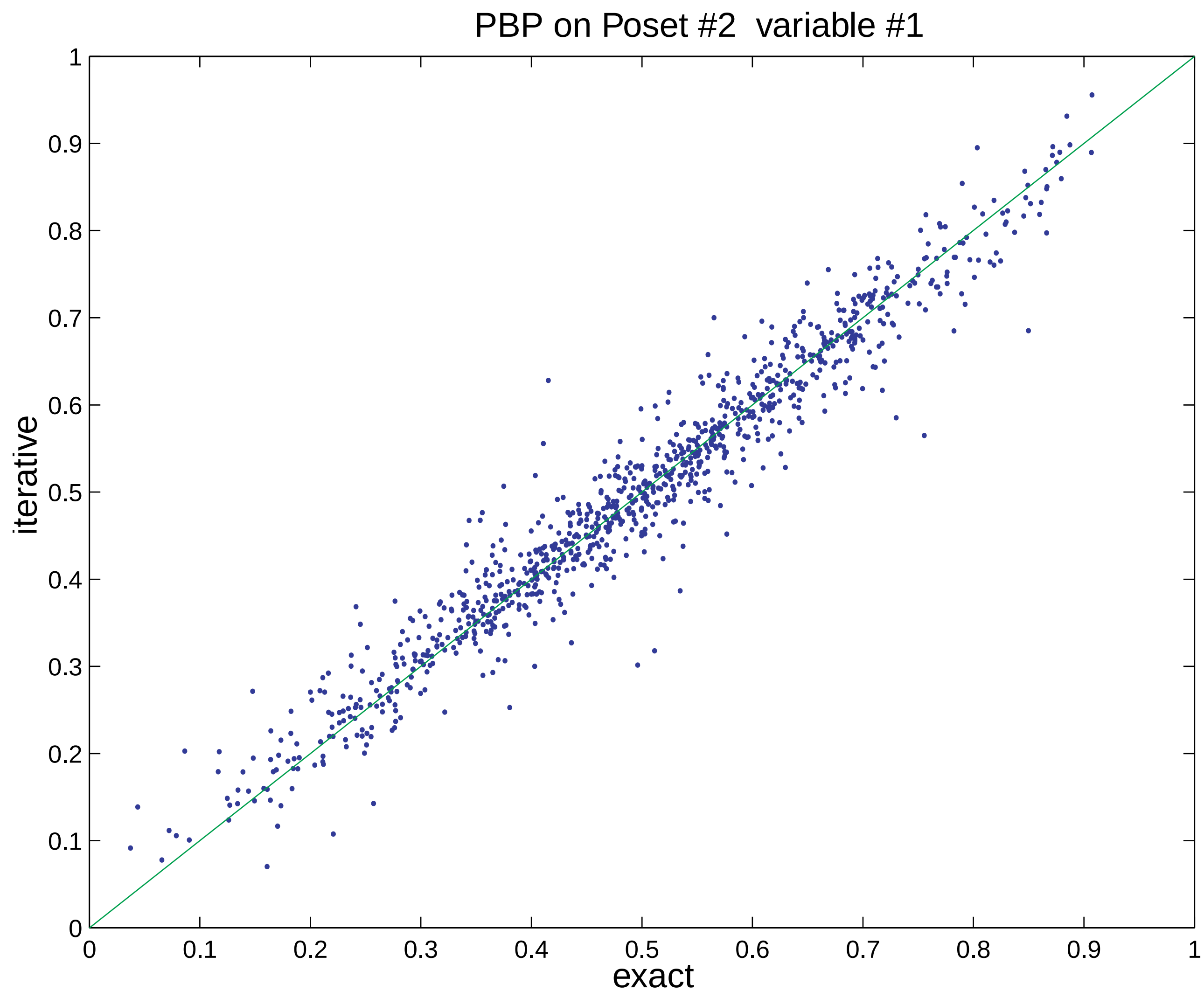
Poset Number Two



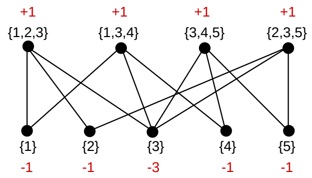


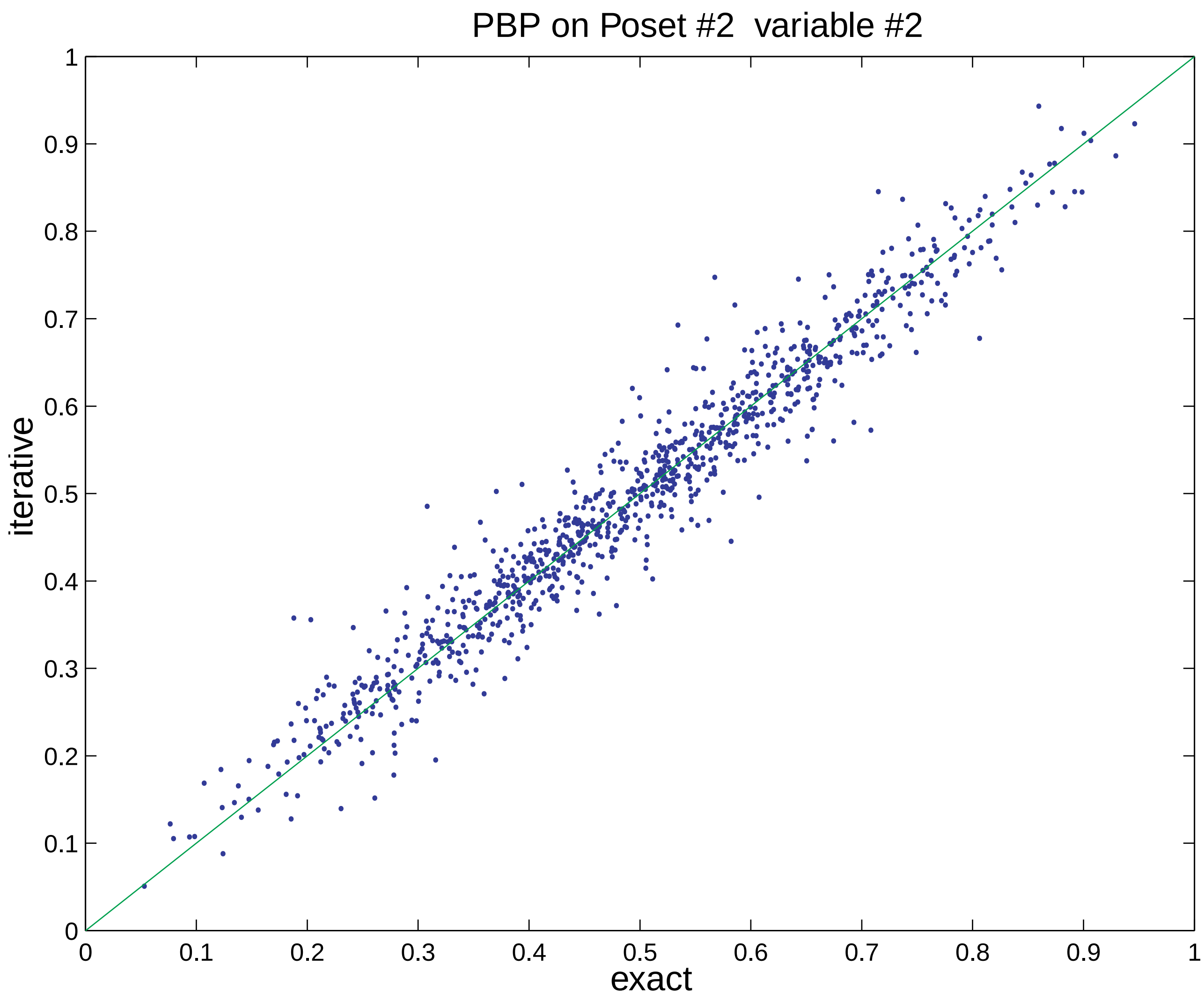
Poset Number Two



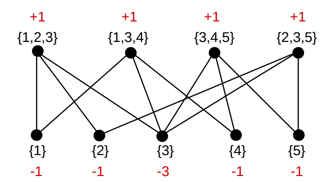


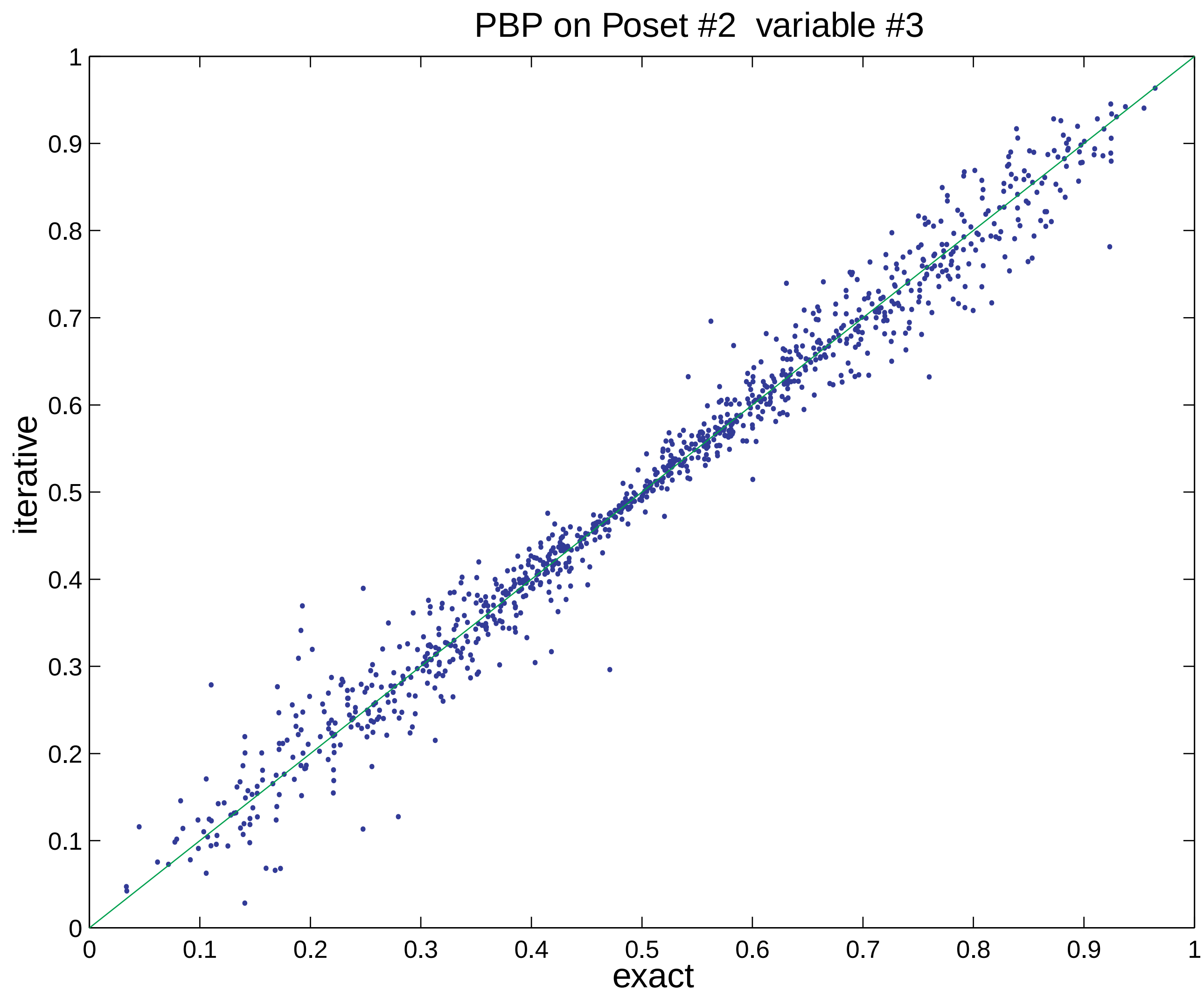
Poset Number Two



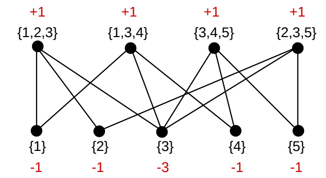


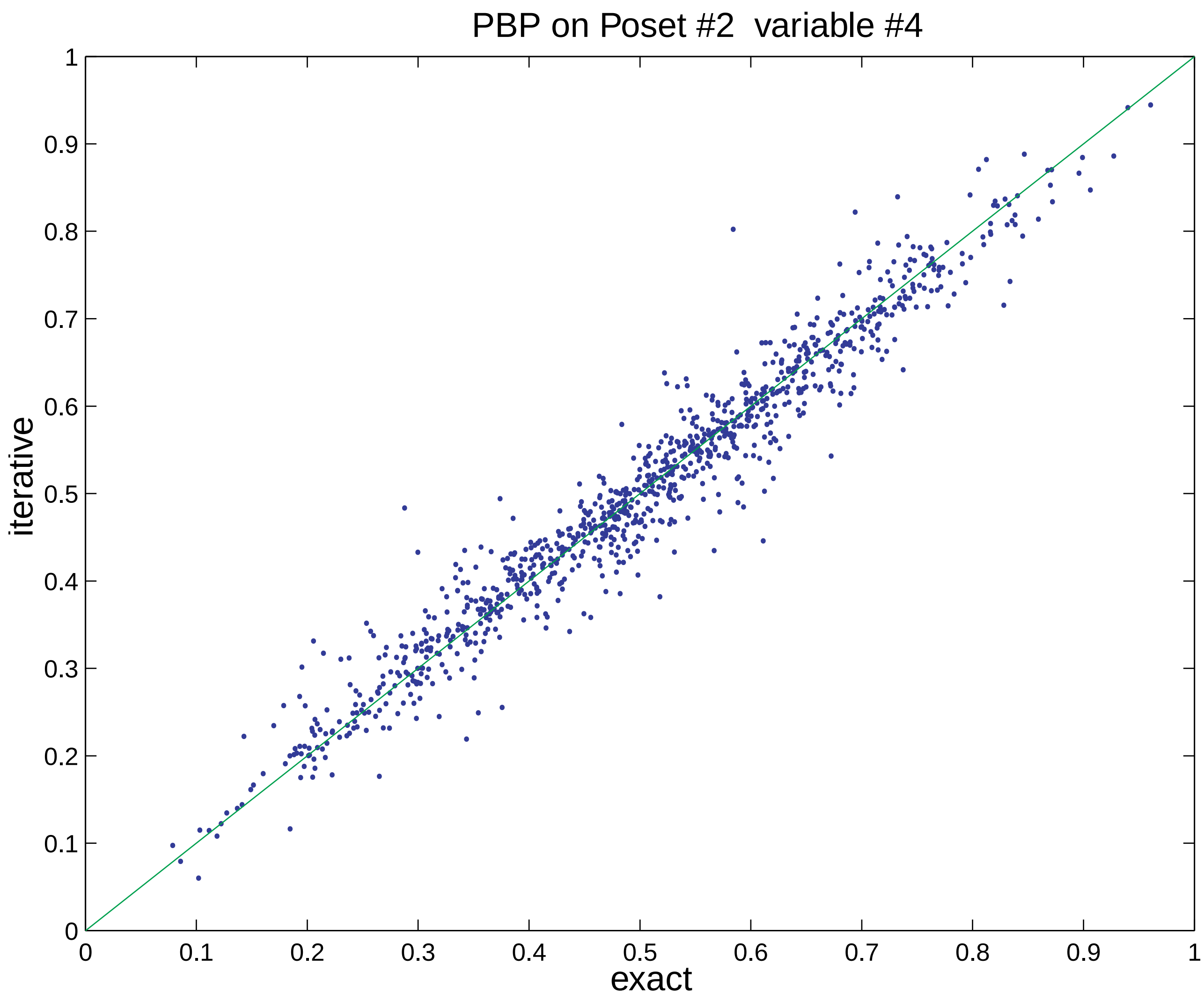
Poset Number Two



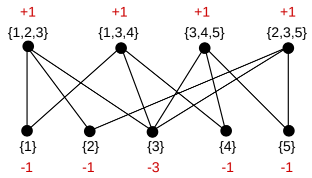


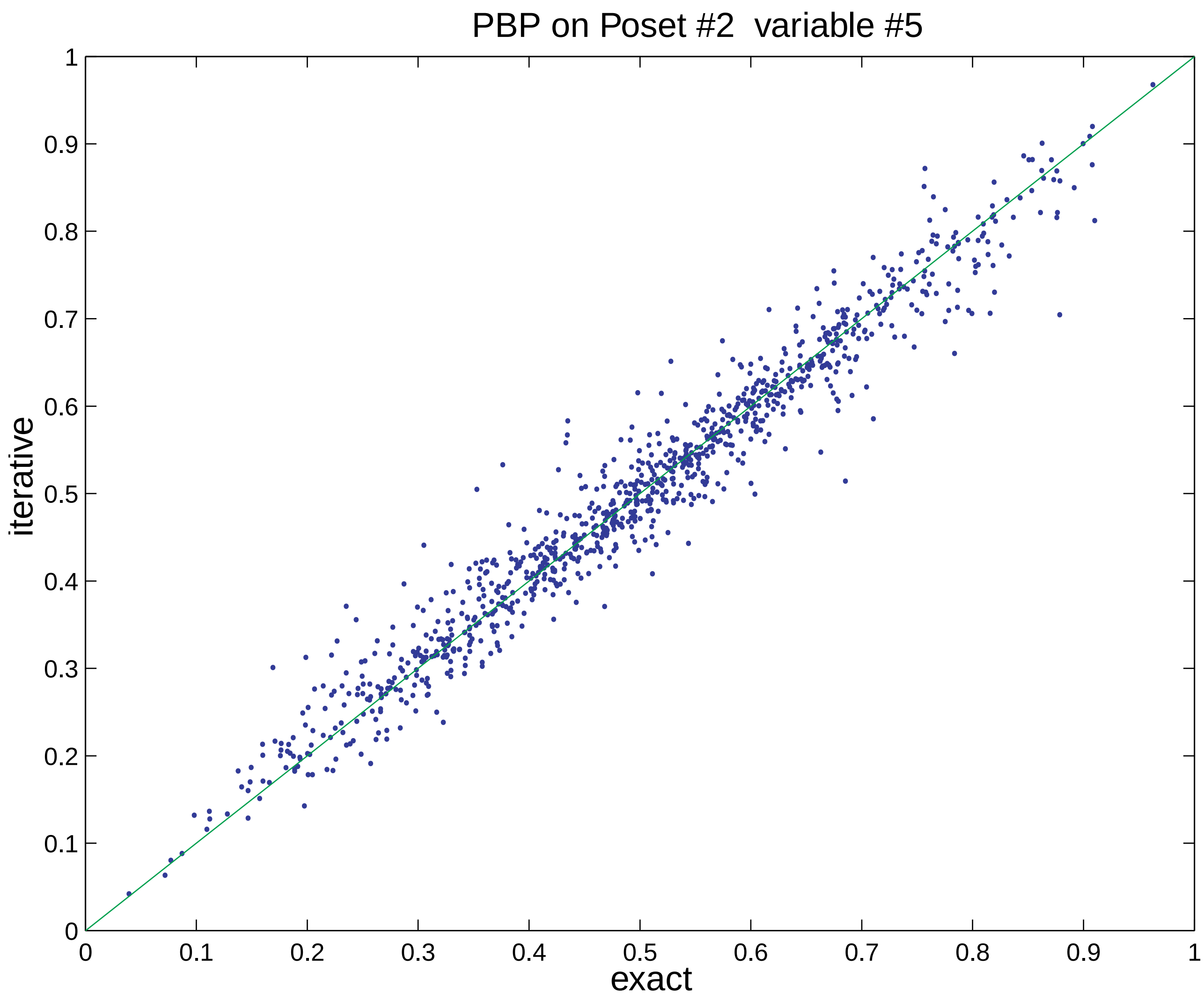
Poset Number Two



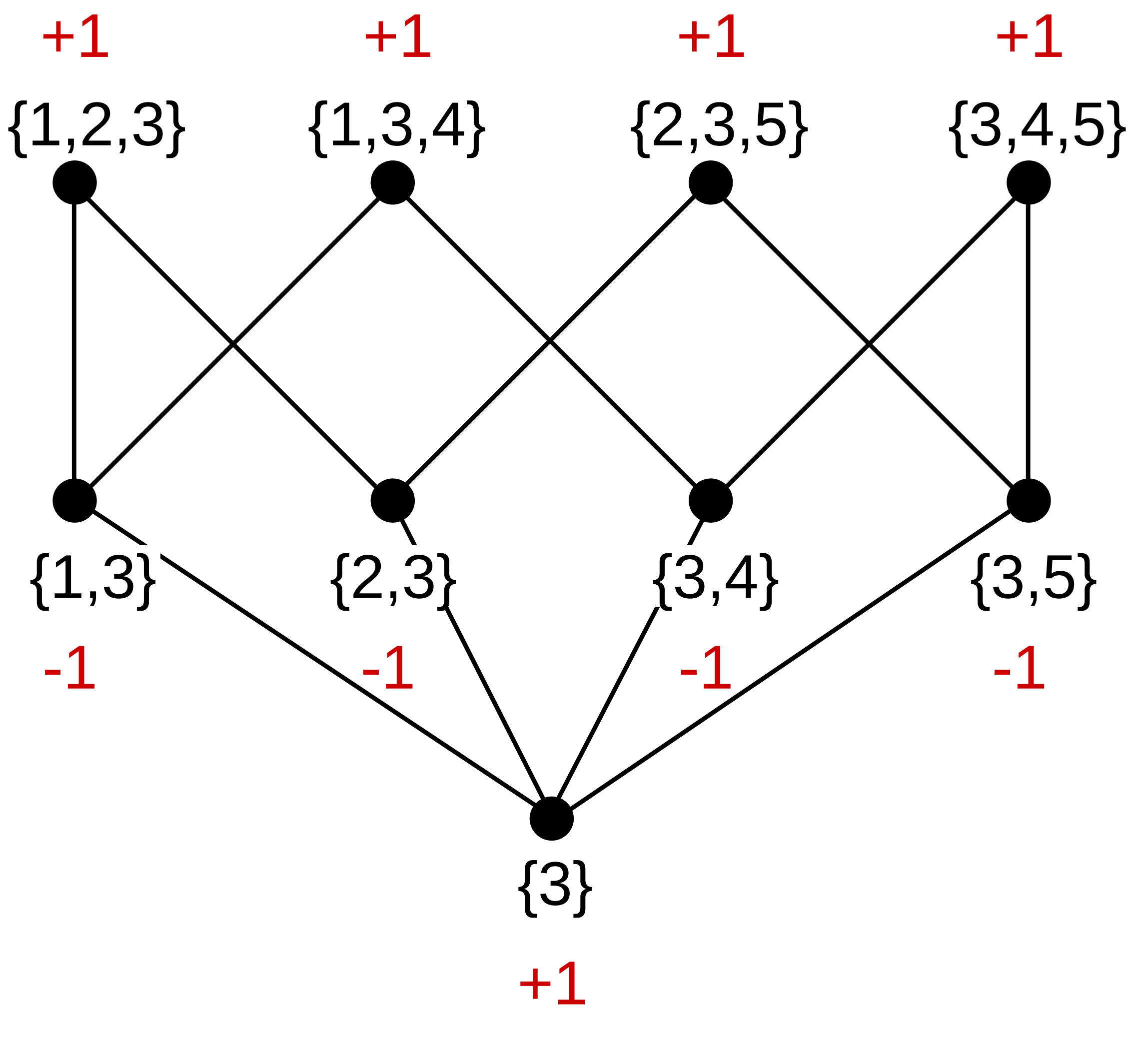


Poset Number Two

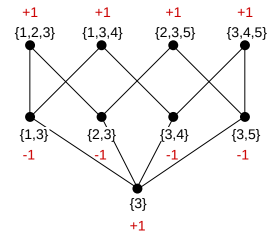


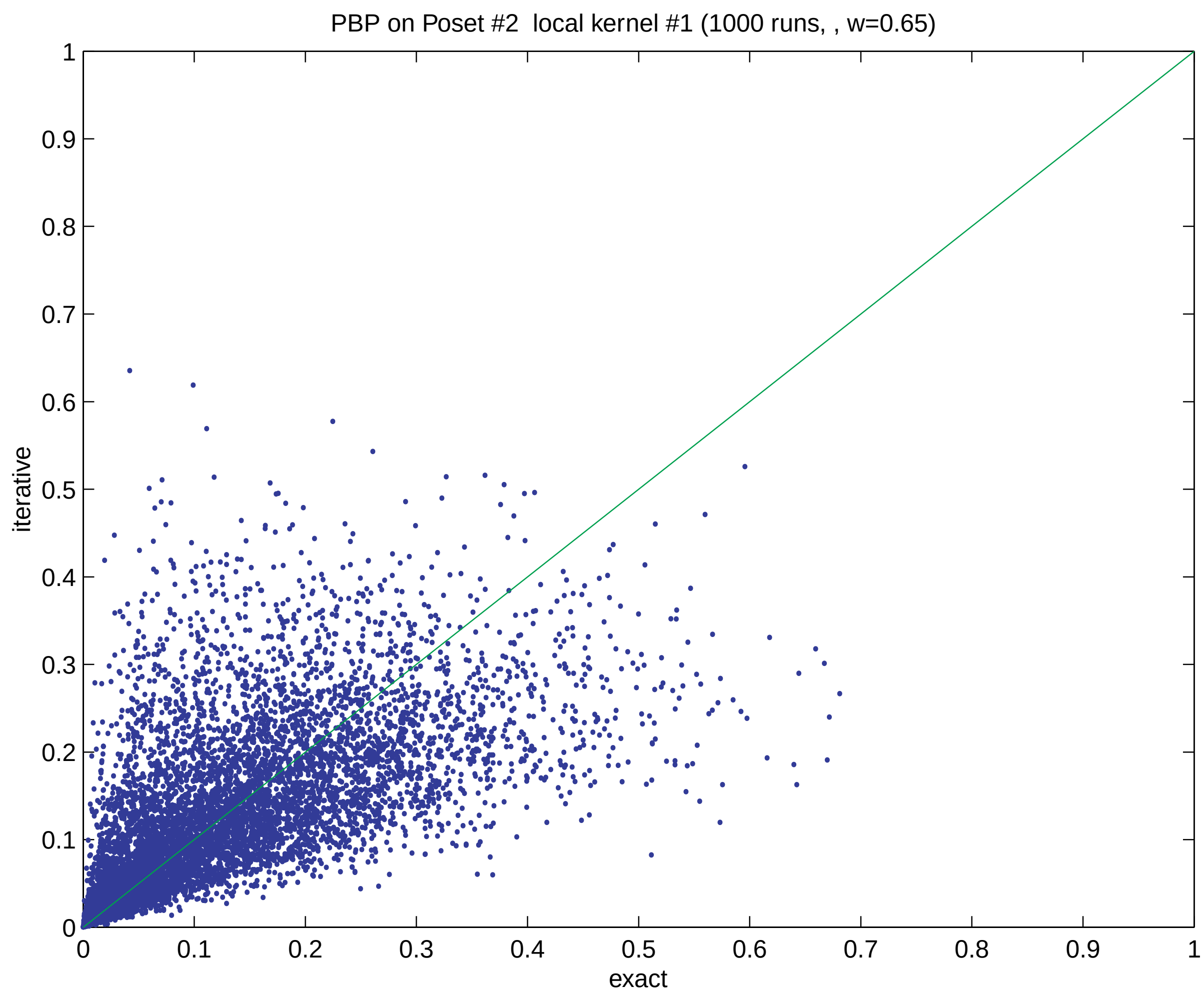


Poset Number Three (Cluster Variational Method)

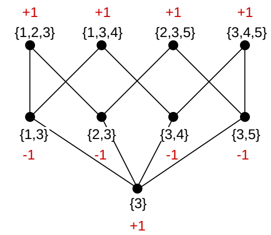


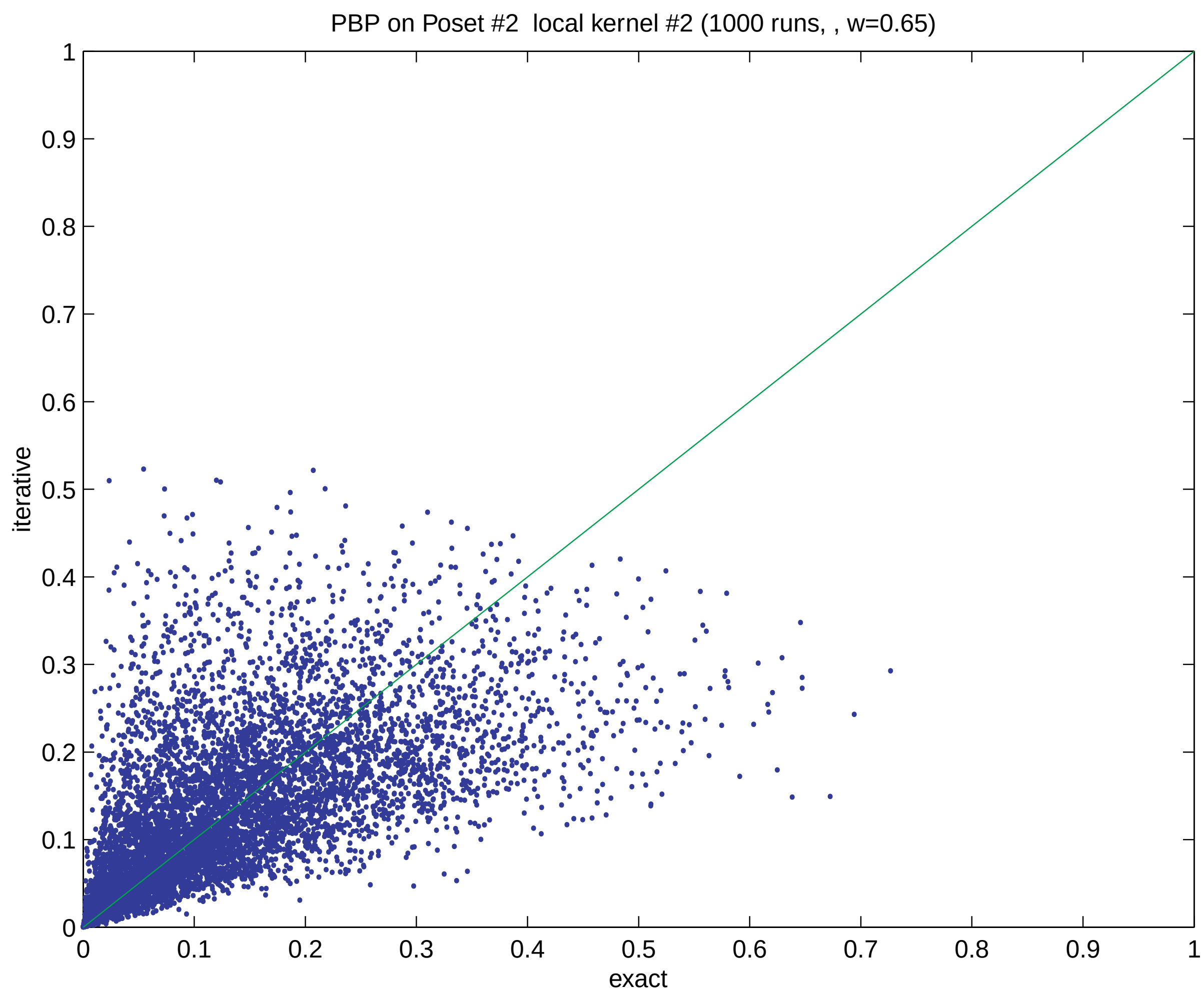
Poset Number Three



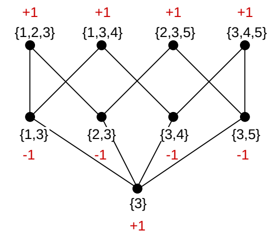


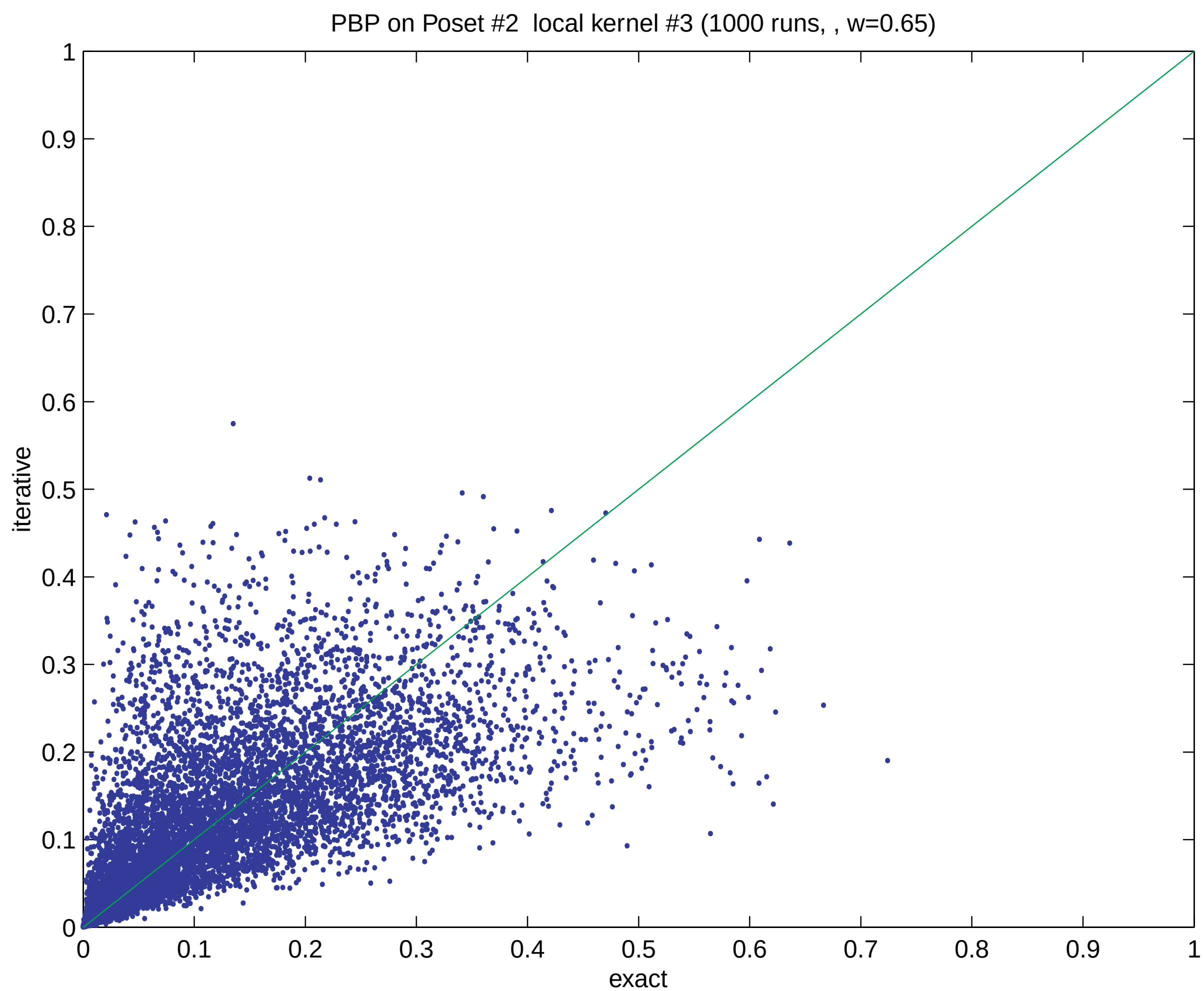
Poset Number Three



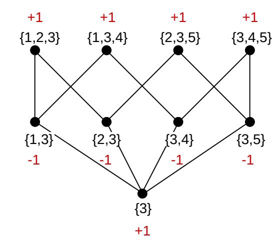


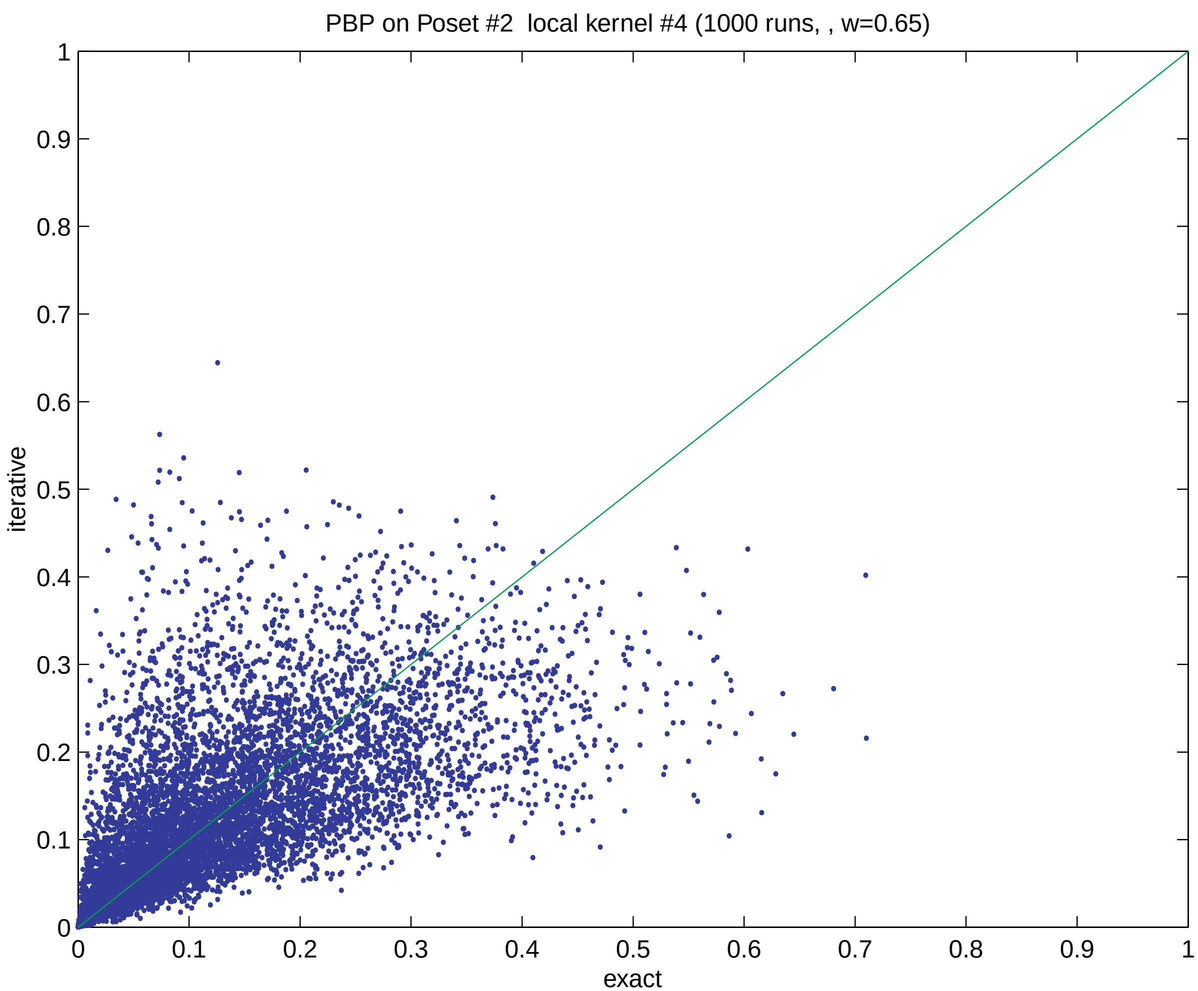
Poset Number Three





Poset Number Three





The Problem is Non-Convergence, and is Easily Repaired.

$$y_{n+1} = F(y_n) \qquad \text{(too aggressive)}$$

The Problem is Non-Convergence, and is Easily Repaired.

$$y_{n+1} = F(y_n) \qquad \text{(too aggressive)}$$

$$y_{n+1} = \sqrt{y_n F(y_n)} \qquad \text{(slower, but surer)}$$

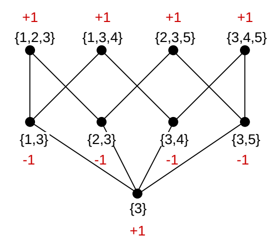
The Problem is Non-Convergence, and is Easily Repaired.

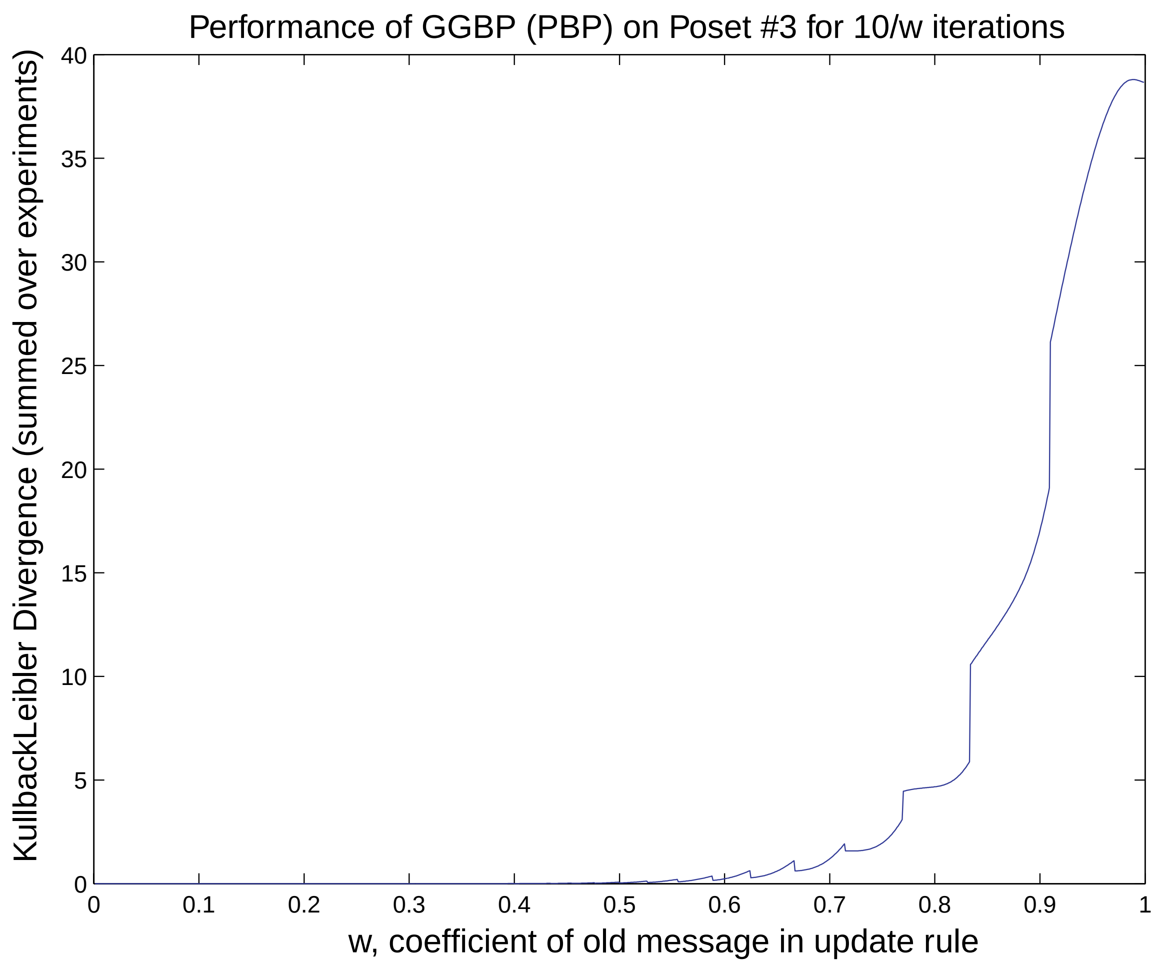
$$y_{n+1} = F(y_n) \qquad \text{(too aggressive)}$$

$$y_{n+1} = \sqrt{y_n F(y_n)} \qquad \text{(slower, but surer)}$$

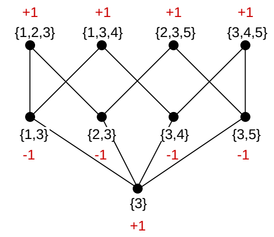
$$y_{n+1} = y_n^{1-w} F(y_n)^w \qquad \text{for } 0 < w < 1.$$

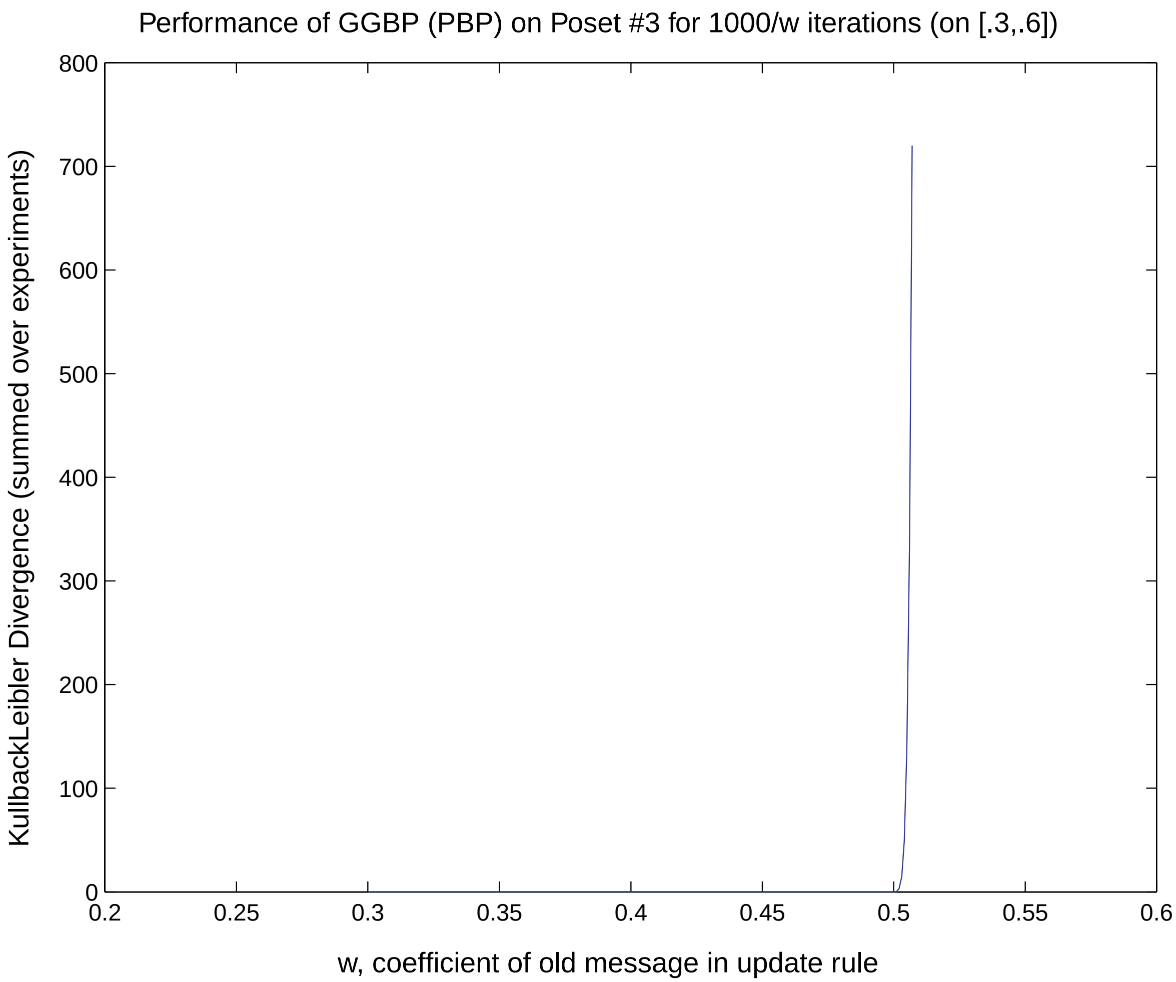
Poset Number Three



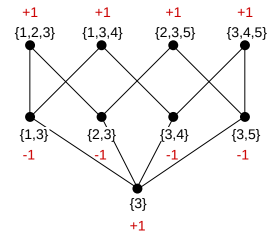


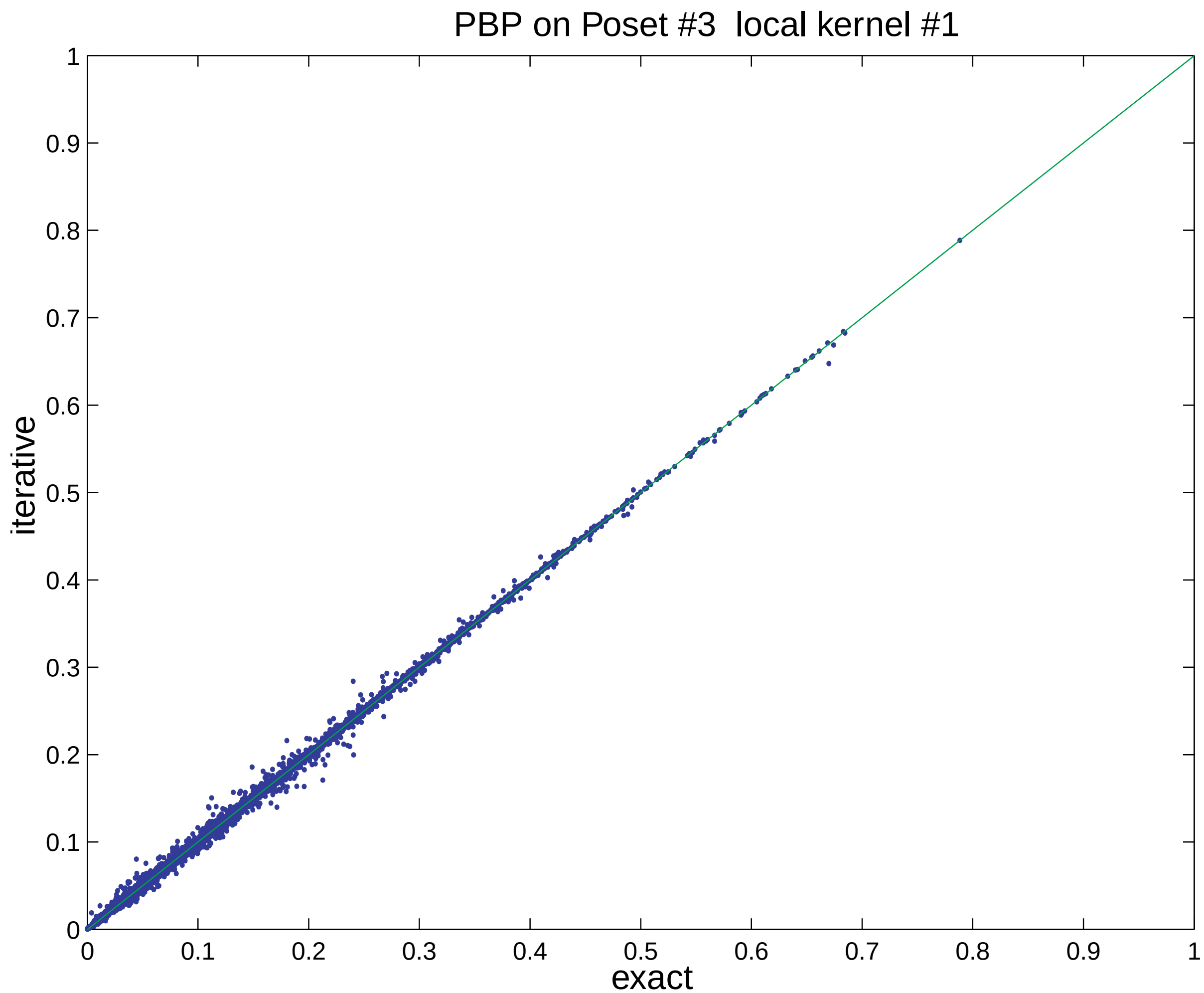
Poset Number Three



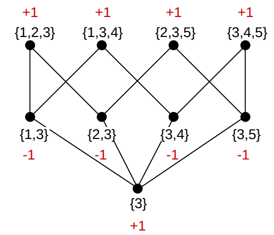


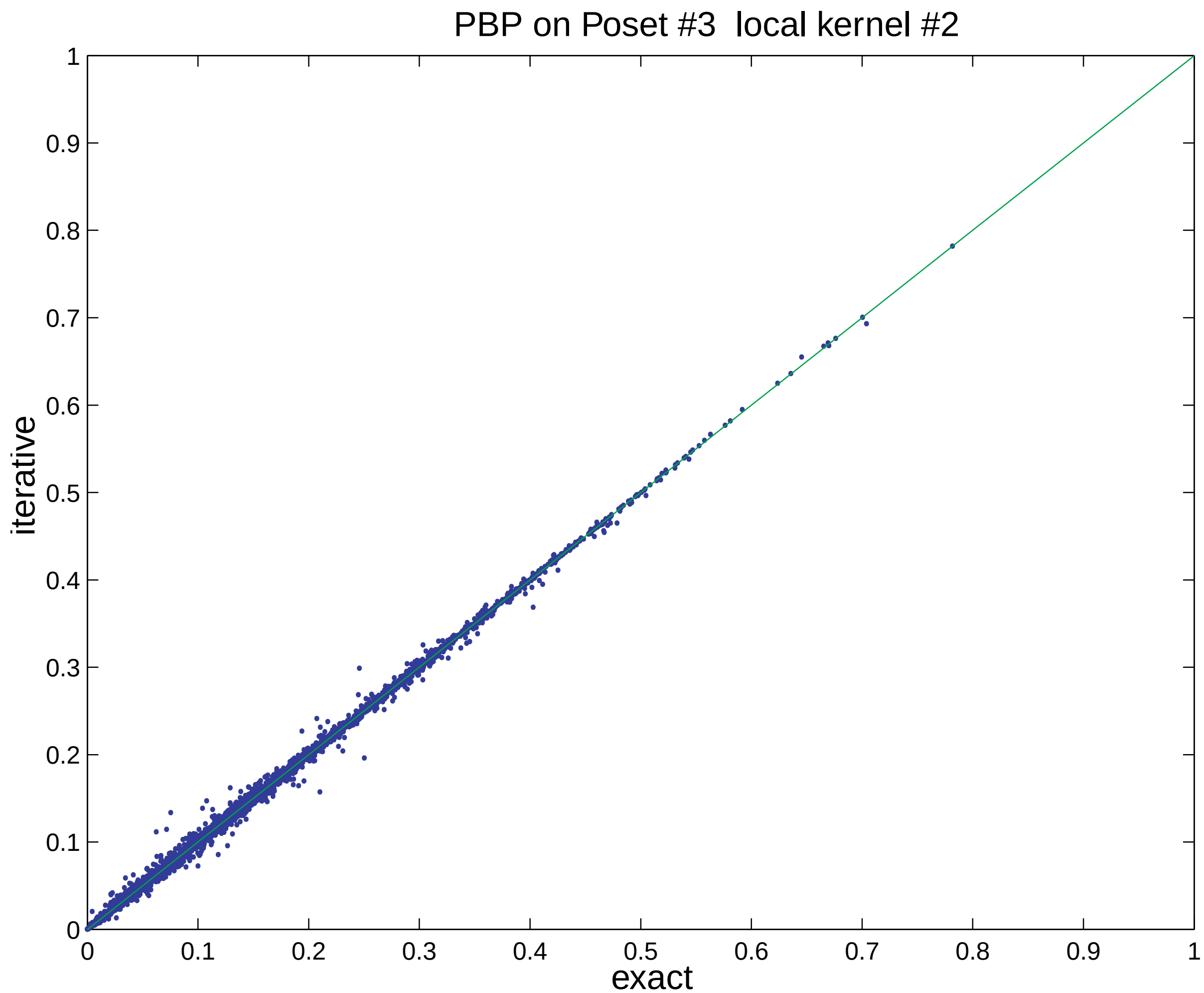
Poset Number Three



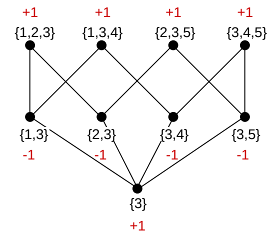


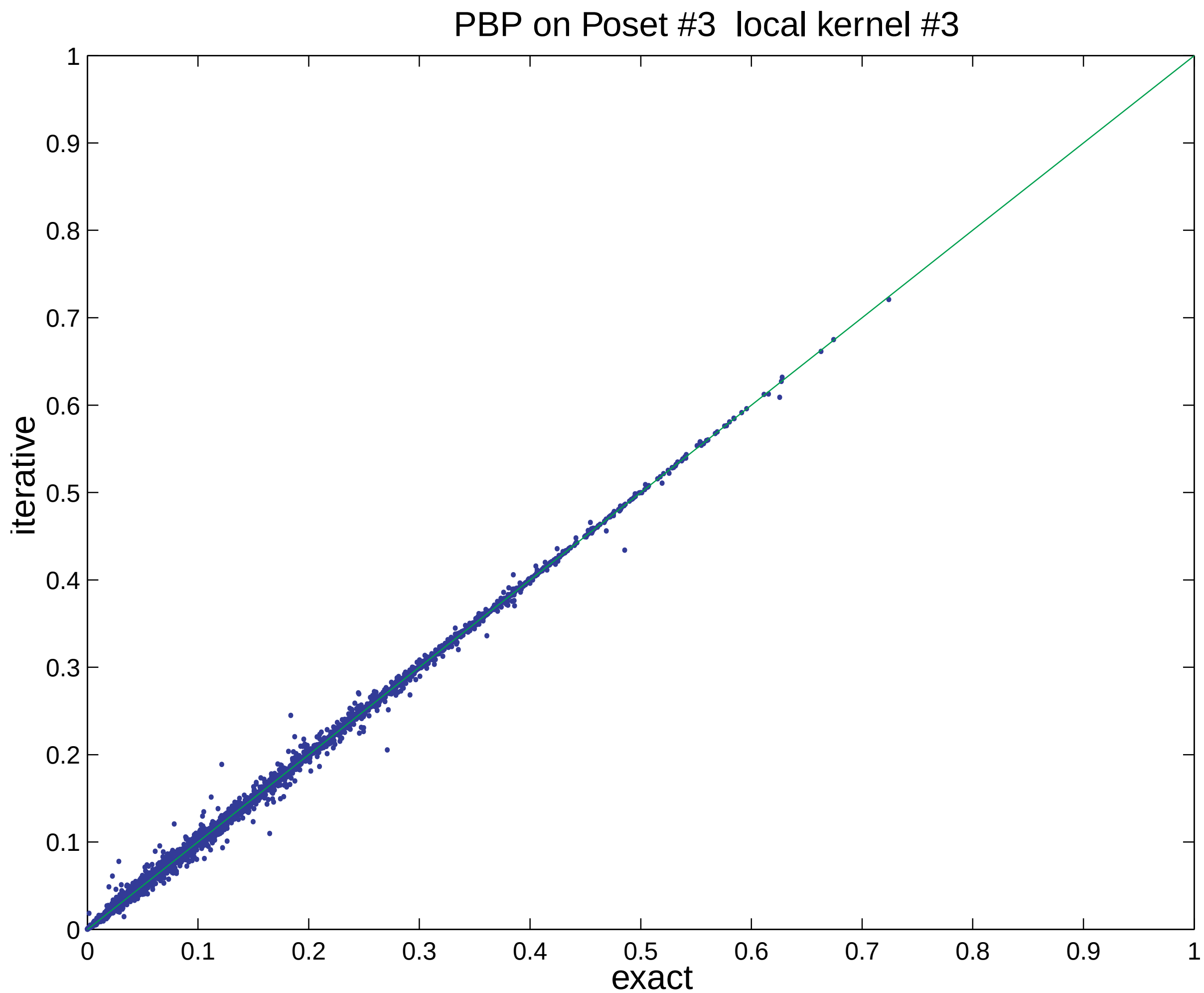
Poset Number Three



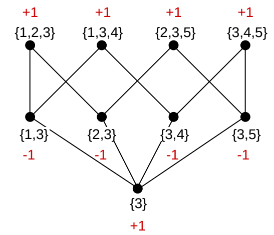


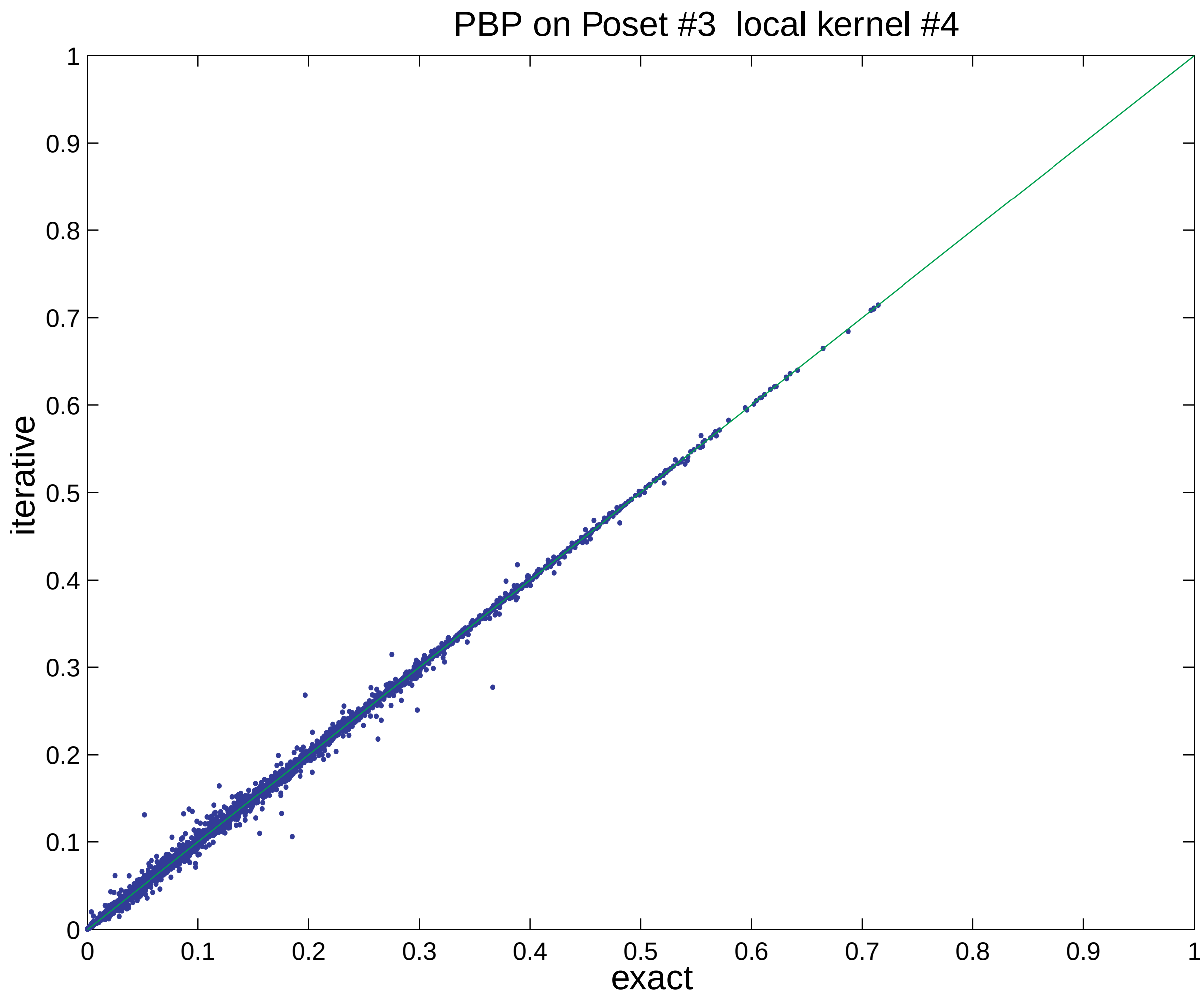
Poset Number Three



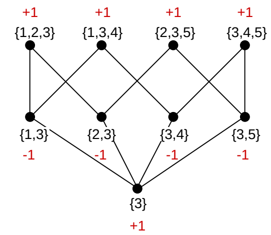


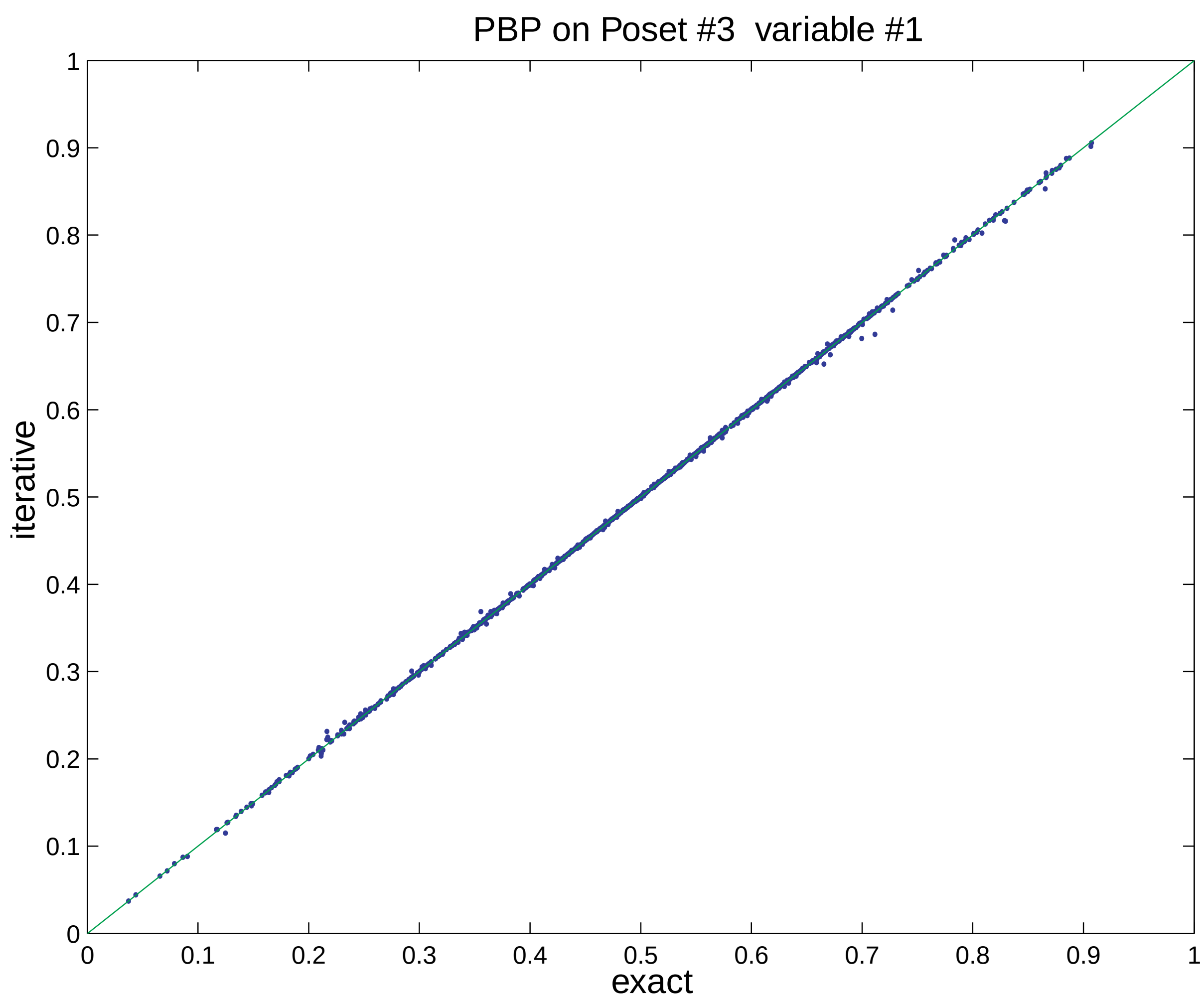
Poset Number Three



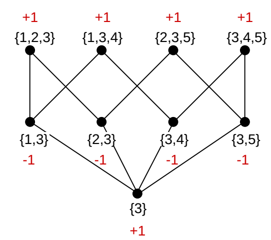


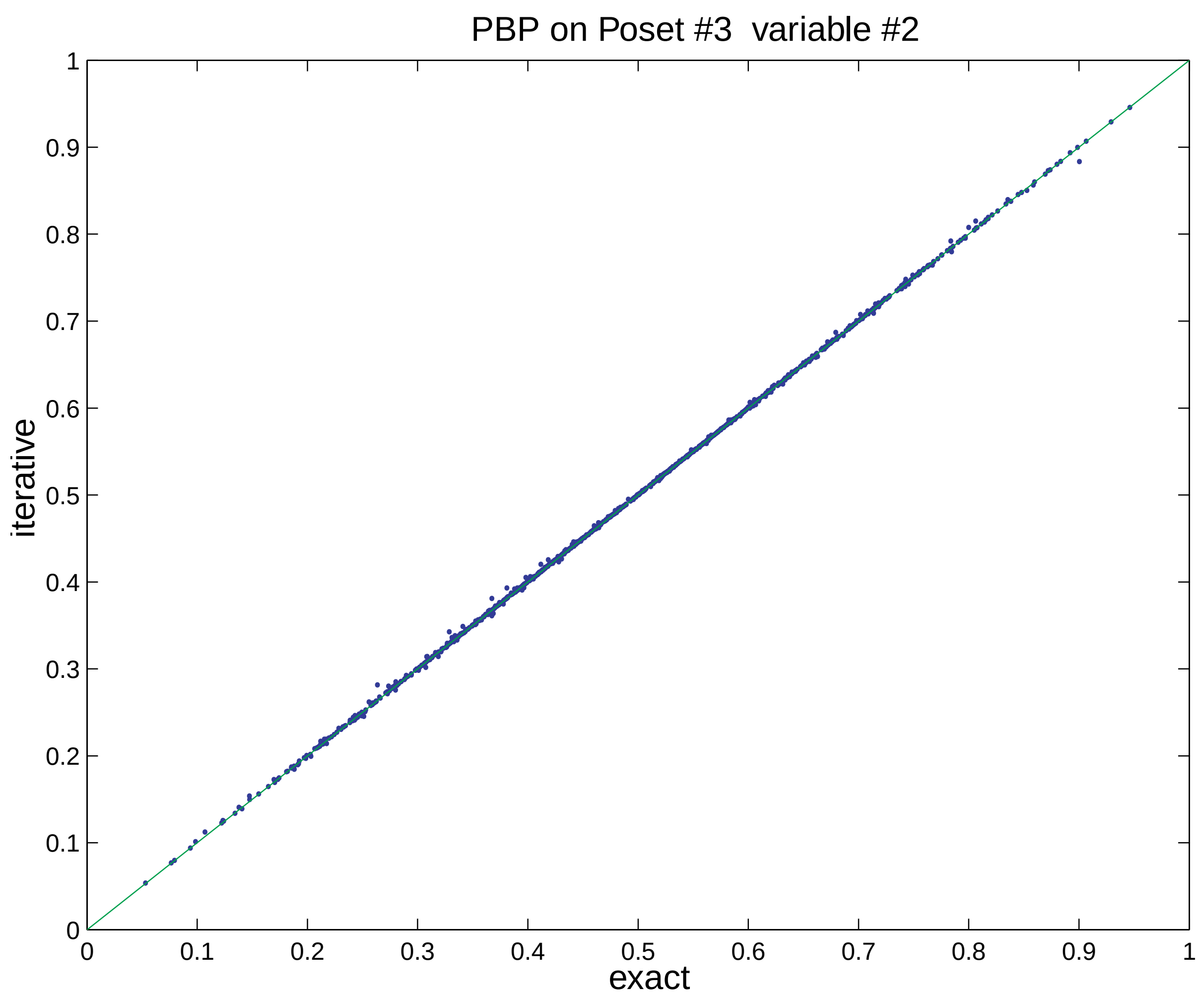
Poset Number Three



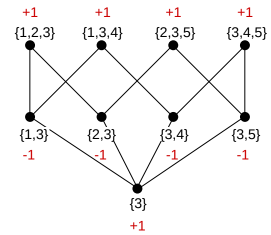


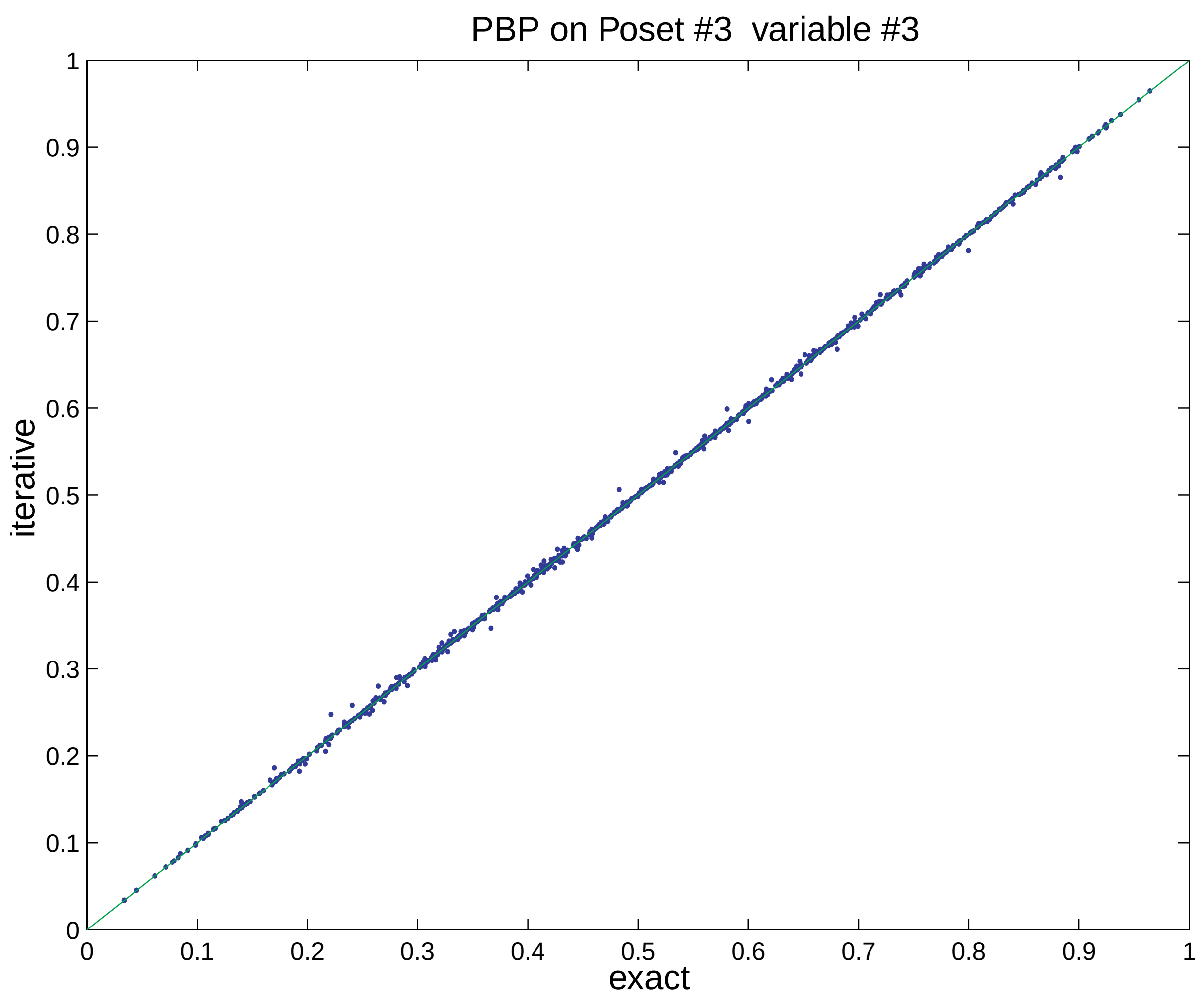
Poset Number Three



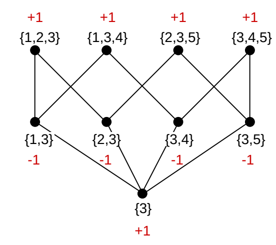


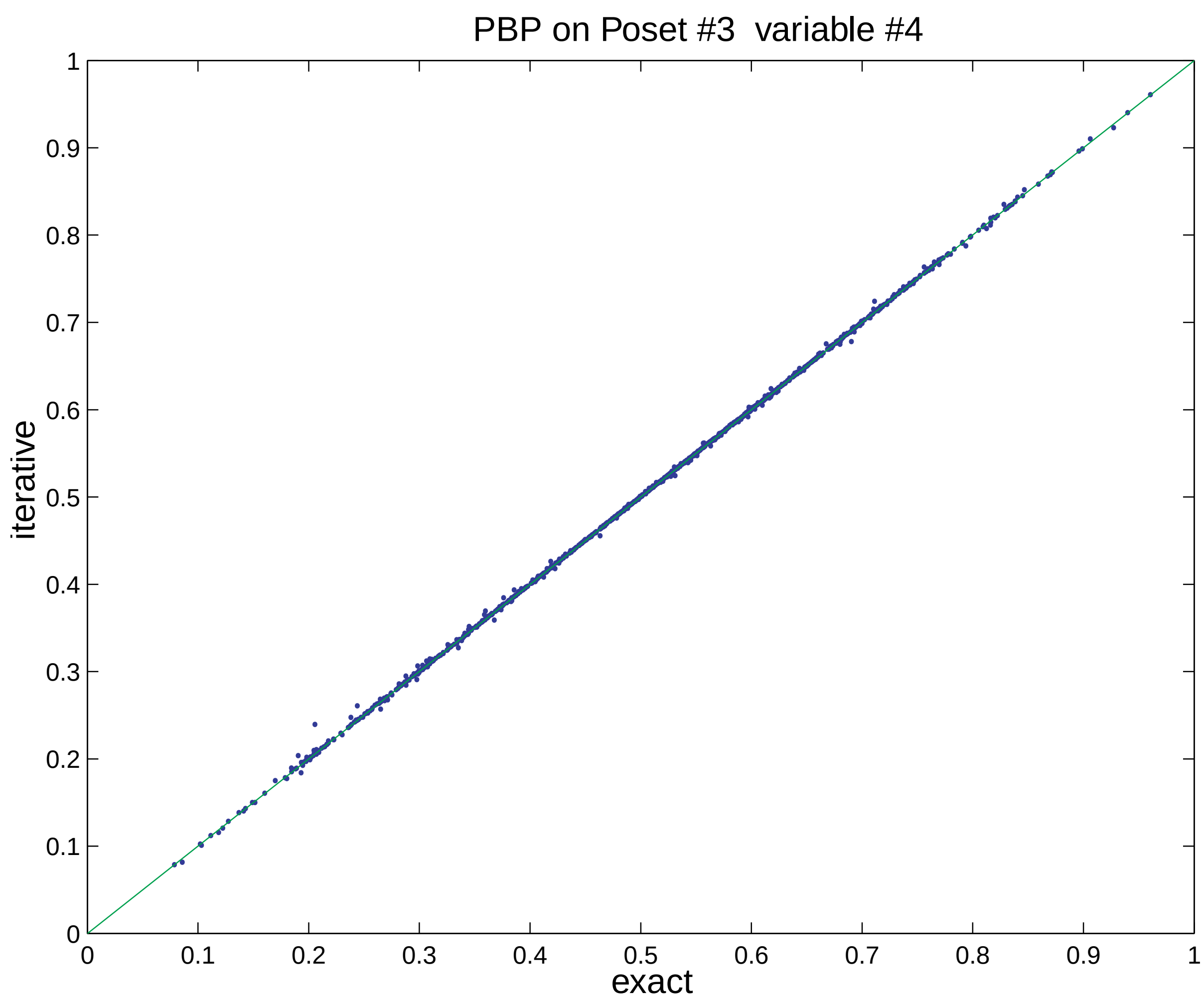
Poset Number Three



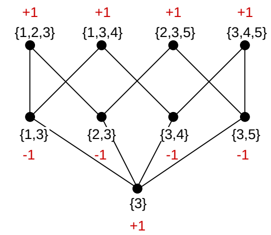


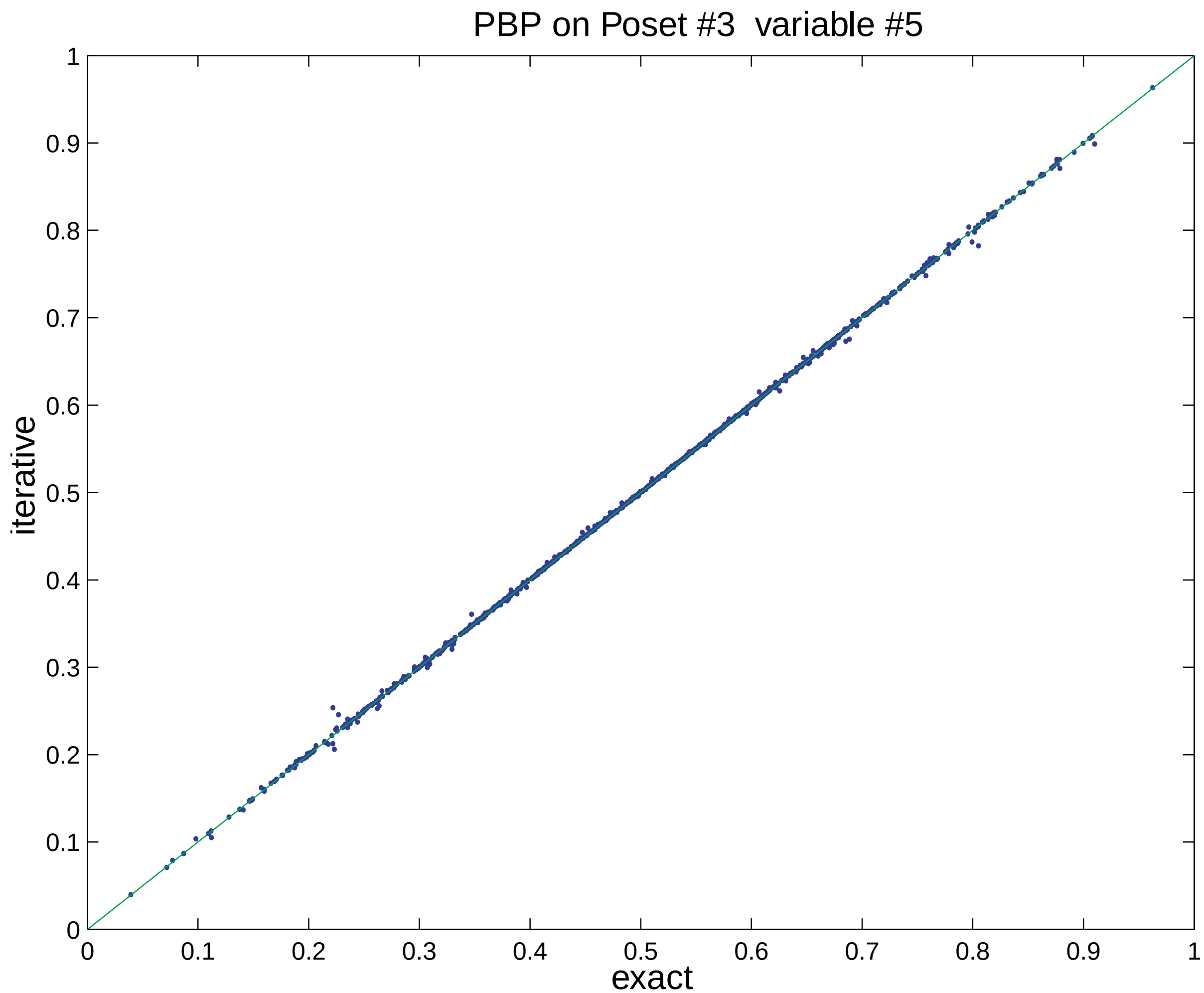
Poset Number Three





Poset Number Three





Why Does It Work?

Why Does It Work?

- No one really knows, but . . .

Why Does It Work?

- No one really knows, but ...
- The PBP algorithm can be viewed as an algorithm for minimizing a certain "energy" function. There is a one-to-one correspondence between the fixed points of PBP and the stationary points of this energy surface.

More Precisely: The Bethe-Kikuchi Approximation

- We know

$$F = \min_{b(\boldsymbol{x})} \widetilde{F}(b(\boldsymbol{x}))$$

$$B(\boldsymbol{x}) = \operatorname*{argmin}_{b(\boldsymbol{x})} \widetilde{F}(b(\boldsymbol{x})).$$

More Precisely: The Bethe-Kikuchi Approximation

- We know

$$F = \min_{b(\boldsymbol{x})} \widetilde{F}(b(\boldsymbol{x}))$$

$$B(\boldsymbol{x}) = \operatorname*{argmin}_{b(\boldsymbol{x})} \widetilde{F}(b(\boldsymbol{x})).$$

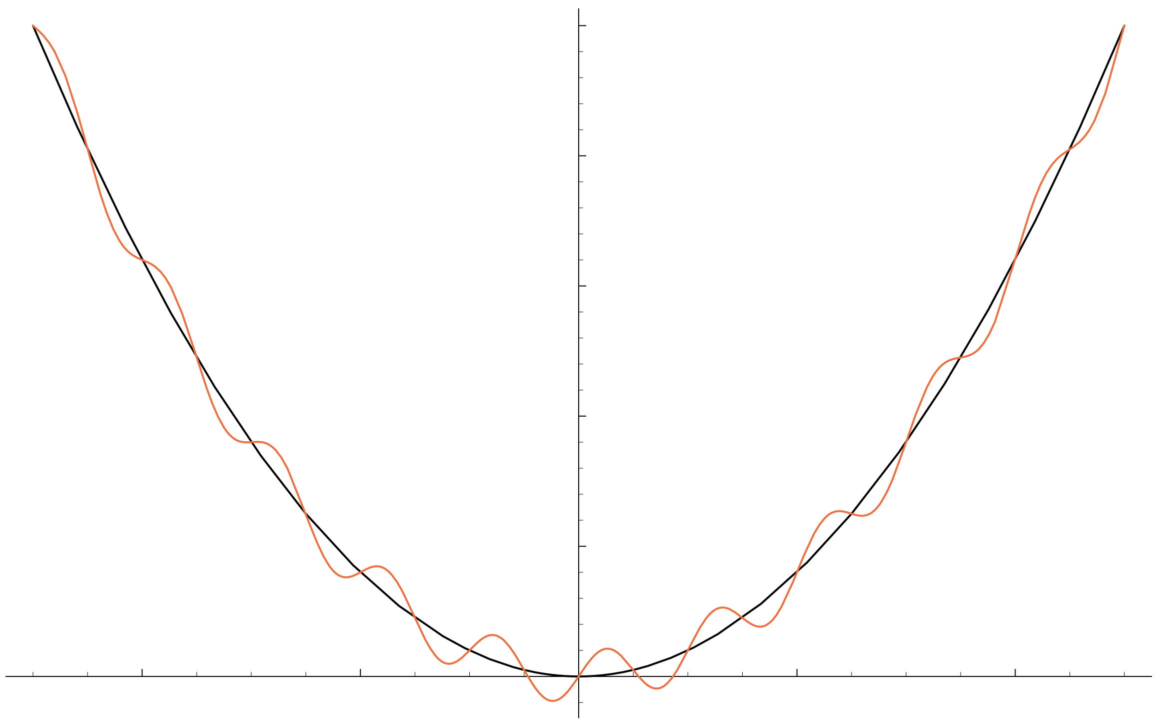
- We approximate $\widetilde{F}(b(\boldsymbol{x}))$ with something that depends only on the poset P and the marginals $b_\rho(\boldsymbol{x}_\rho)$ of $b(\boldsymbol{x})$:

$$\widetilde{F}_P(b(\boldsymbol{x})) = \sum_{\rho \in P} \phi(\rho) \widetilde{F}_\rho(b_\rho(\boldsymbol{x}_\rho),$$

where $\widetilde{F}_\rho(b_\rho(\boldsymbol{x}_\rho))$ is the local free energy at ρ, defined as

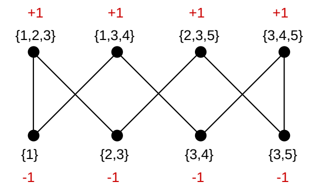
$$\sum_{\boldsymbol{x}_\rho} b_\rho(\boldsymbol{x}_\rho) E_\rho(\boldsymbol{x}_\rho) + \sum_{\boldsymbol{x}_\rho} b_\rho(\boldsymbol{x}_\rho) \ln b_\rho(\boldsymbol{x}_\rho).$$

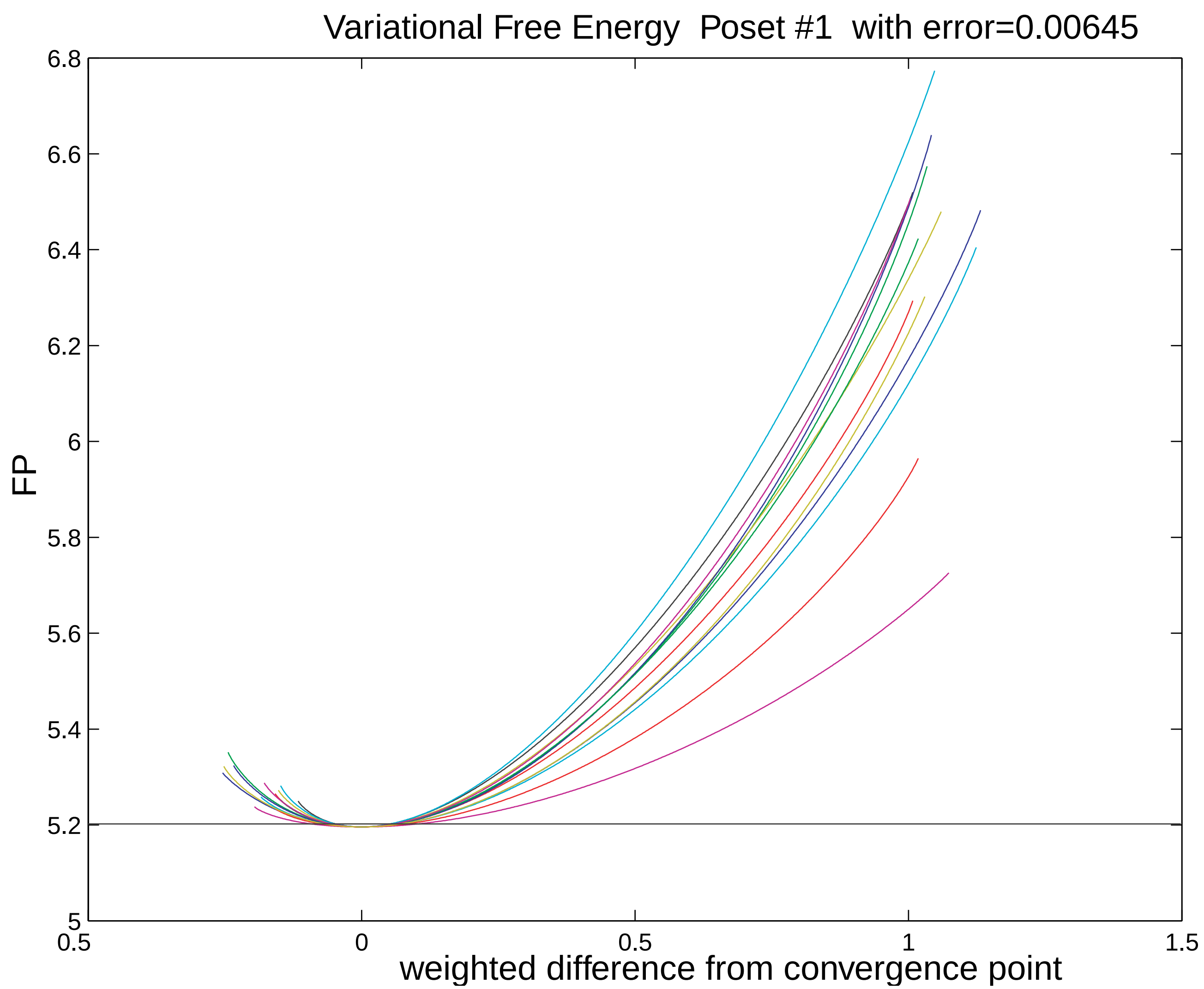
An Analogy:



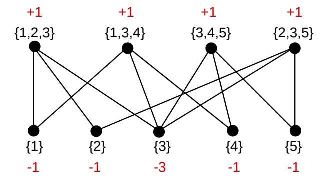
- Black line $= \widetilde{F}$
- Colored line $= \widetilde{F}_P$.
- The hope is that $\min_{\{b_\rho(\boldsymbol{x}_\rho)\}} \widetilde{F}_P \approx \min_{b(\boldsymbol{x})} \widetilde{F} = F$.

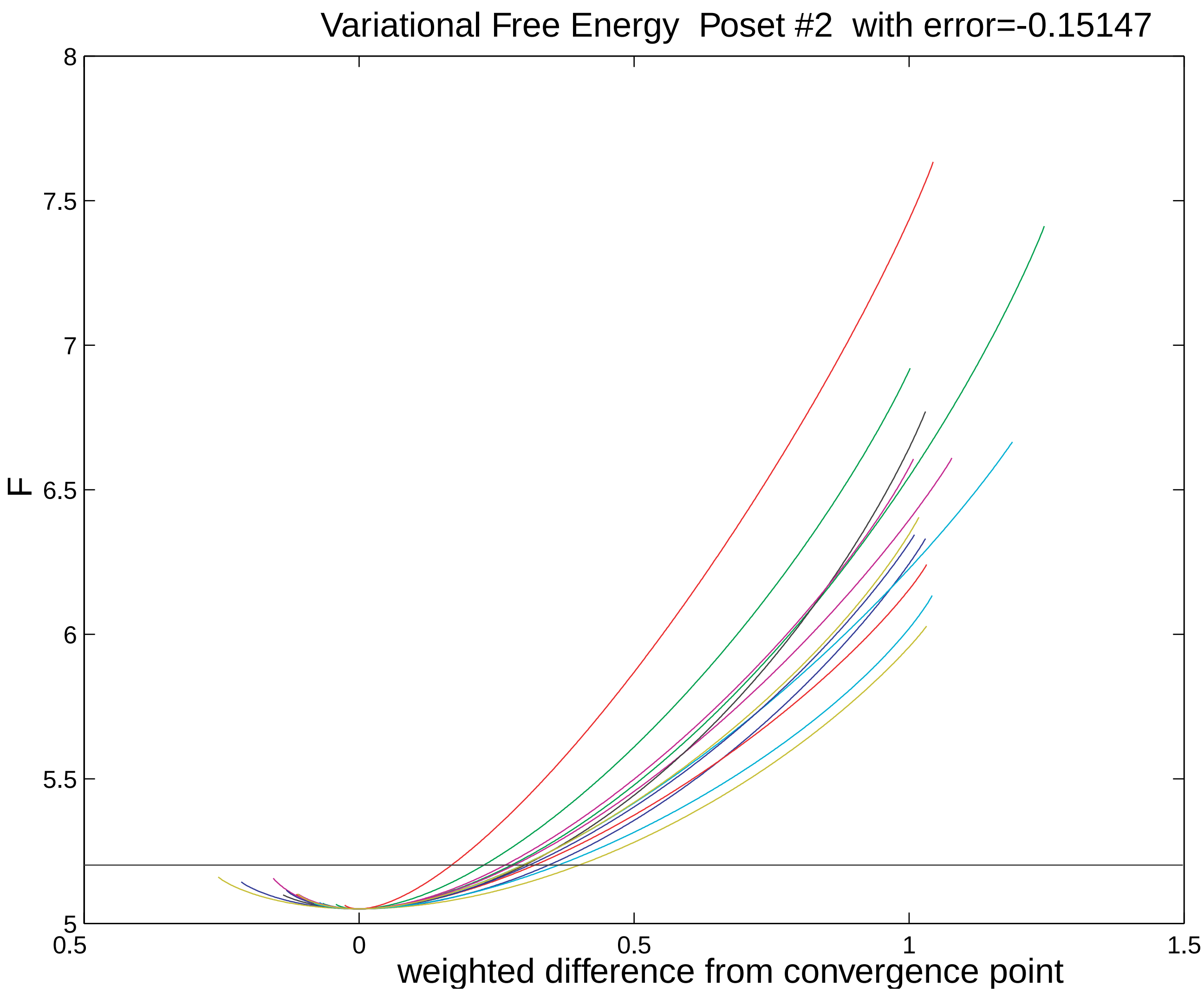
Poset Number One — The BK Approximation



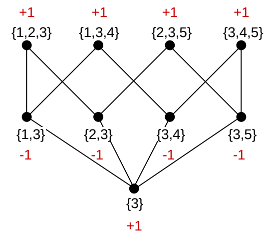


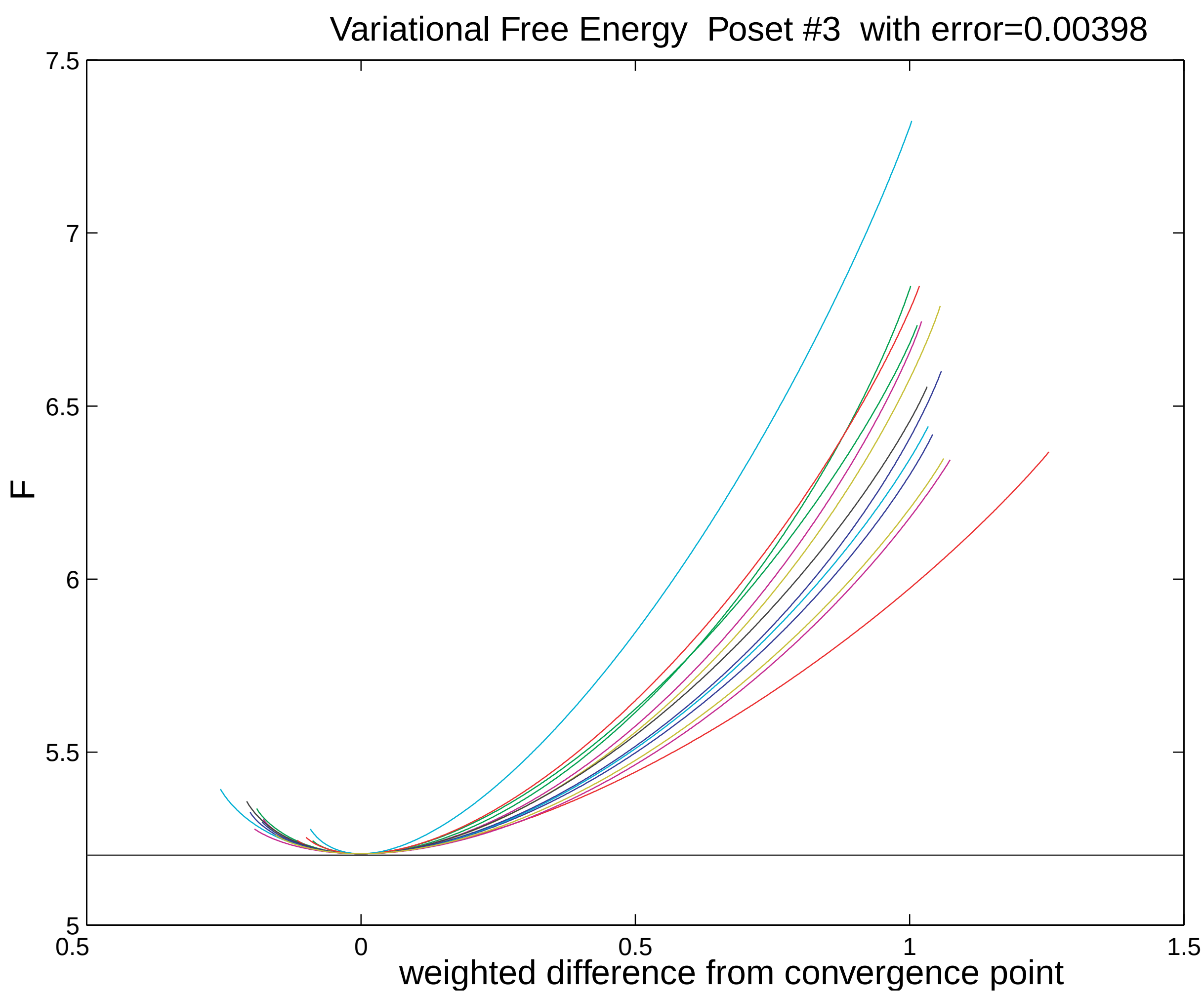
Poset Number Two — The BK Approximation





Poset Number Three — The BK Approximation





Some Open Questions

Some Open Questions

- What is the “best” poset for a given MPD problem?

Some Open Questions

- What is the "best" poset for a given MPD problem?
- Prove (or find a counterexample) that the *stable* fixed points of PBP correspond to the *local minima* of the BK variational free energy.

Some Open Questions

- What is the "best" poset for a given MPD problem?
- Prove (or find a counterexample) that the *stable* fixed points of PBP correspond to the *local minima* of the BK variational free energy.
- Can the PBP algorithm always be made to converge using the "smoothing" trick?

Some Open Questions

- What is the "best" poset for a given MPD problem?
- Prove (or find a counterexample) that the *stable* fixed points of PBP correspond to the *local minima* of the BK variational free energy.
- Can the PBP algorithm always be made to converge using the "smoothing" trick?
- What is the relationship between the BK approximate free energy and the exact (Helmholtz), free energy?

Some Open Questions

- What is the "best" poset for a given MPD problem?
- Prove (or find a counterexample) that the *stable* fixed points of PBP correspond to the *local minima* of the BK variational free energy.
- Can the PBP algorithm always be made to converge using the "smoothing" trick?
- What is the relationship between the BK approximate free energy and the exact (Helmholtz), free energy?
- Can other combinatorial optimization methods, e.g. simulated annealing, be used to minimize $\widetilde{F}_P$, thereby leading to alternative "BP" algorithms?