DGP formulations and methods

- System of equations
- Unconstrained global optimization (GO)
- Constrained global optimization
- SDP relaxations and their properties
- Diagonal dominance
- Concentration of measure in SDP
- Isomap for DGP

System of quadratic equations

$$\forall \{u,v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2 \tag{8}$$

Computationally: useless

(less than 10 vertices with $K = 3$ using Octave)

Unconstrained Global Optimization

$$\min_{x} \sum_{\{u,v\}\in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2 \qquad \textbf{(9)}$$

Globally optimal obj. fun. value of (9) is 0 iff x solves (8)

Computational experiments in [Liberti et al., 2006]:

- **GO solvers from 10 years ago**
- **randomly generated protein data: ≤ 50 atoms**
- **cubic crystallographic grids: ≤ 64 atoms**

Constrained global optimization

- $\min_x \sum_{\{u,v\}\in E} |\|x_u - x_v\|^2 - d_{uv}^2|$ **exactly reformulates (8)**
- **Relax objective f to concave part, remove constant term, rewrite** $\min -f$ **as** $\max f$
- **Reformulate convex part of obj. fun. to convex constraints**
- *Exact reformulation*

$$\left.\begin{array}{rc} \max_x & \sum\limits_{\{u,v\}\in E} \|x_u - x_v\|^2 \\ \forall \{u,v\} \in E & \|x_u - x_v\|^2 \le d_{uv}^2 \end{array}\right\} \qquad \textbf{(10)}$$

Theorem (Activity)

At a glob. opt. x^* of a YES instance, all constraints of (10) *are active*

Linearization

$$\Rightarrow \quad \forall \{i,j\} \in E \quad \|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i \cdot x_j = d_{ij}^2$$

$$\Rightarrow \left\{ \begin{array}{rcl} \forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\ X & = & x\,x^\top \end{array} \right.$$

Relaxation

$$\begin{array}{rrcl} & X & = & x\,x^\top \\ \Rightarrow & X - x\,x^\top & = & 0 \end{array}$$

$$\begin{array}{rrcl} \text{(relax)} & \Rightarrow \quad X - x\,x^\top & \succeq & 0 \end{array}$$

$$\mathsf{Schur}(X,x) = \begin{pmatrix} I_K & x^\top \\ x & X \end{pmatrix} \succeq 0$$

If x does not appear elsewhere $\Rightarrow$ get rid of it (e.g. choose $x = 0$):

replace $\mathsf{Schur}(X,x) \succeq 0$ ***by*** $X \succeq 0$

SDP relaxation

$$
\begin{array}{rcl}
\min F \bullet X & & \\
\forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} & = & d_{ij}^2 \\
X & \succeq & 0
\end{array}
$$

How do we choose F?

Some possible objective functions

- **For protein conformation:**

$$\max \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij})$$

  with $=$ changed to $\leq$ in constraints *(or* $\min$ *and* $\geq$*)*
  "push-and-pull" the realization

- [Ye, 2003], application to wireless sensors localization

$$\min \mathsf{Tr}(X)$$

  improve covariance estimator accuracy

- *How about "just random"?*

How do you choose?

for want of some better criterion...

TEST!

- Download protein files from Protein Data Bank (PDB)
  they contain atom realizations
- Mimick a Nuclear Magnetic Resonance experiment
  Keep only pairwise distances < 5.5
- Try and reconstruct the protein shape from those weighted graphs
- Quality evaluation of results:
  - $\textbf{LDE}(x) = \max_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij} \,|$
  - $\textbf{MDE}(x) = \frac{1}{|E|} \sum_{\{i,j\}\in E} |\, \|x_i - x_j\| - d_{ij} \,|$

Objective function tests

SDP solved with Mosek

SDP + PCA

Instance			LDE			MDE			CPU		
Name	$\lvert V\rvert$	$\lvert E\rvert$	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*
C0700odd.1	15	39	**3.31**	4.57	4.44	**1.92**	2.52	2.50	0.13	**0.07**	0.08
C0700odd.C	36	242	10.61	**4.85**	**4.85**	3.02	3.02	3.02	0.69	**0.43**	0.44
C0700.odd.G	36	308	**4.57**	4.77	4.77	**2.41**	2.84	2.84	0.86	**0.54**	**0.54**
C0150alter.1	37	335	**4.66**	4.88	4.86	**2.52**	3.00	3.00	0.97	0.59	**0.58**
C0080create.1	60	681	7.17	**4.86**	**4.86**	**3.08**	3.19	3.19	2.48	**1.46**	**1.46**
tiny	37	335	**4.66**	4.88	4.88	**2.52**	3.00	3.00	0.97	**0.60**	**0.60**
1guu-1	150	959	10.20	**4.93**	**4.93**	3.43	3.43	3.43	9.23	**5.68**	5.70

SDP + PCA + NLP

Instance			LDE			MDE			CPU		
Name	$\lvert V\rvert$	$\lvert E\rvert$	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*	*PP*	*Ye*	*Rnd*
1b03	89	456	0.00	0.00	0.00	0.00	0.00	0.00	8.69	**6.28**	9.91
1crn	138	846	0.81	0.81	0.81	0.07	0.07	0.07	33.33	**31.32**	44.48
1guu-1	150	959	0.97	4.93	**0.92**	0.10	3.43	**0.08**	56.45	**7.89**	65.33

Choice

- *Ye* very fast but often imprecise
- *Random* good but nondeterministic
- ***Push-and-Pull*** **relaxes** $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2$ **to** $X_{ii} + X_{jj} - 2X_{ij} \geq d_{ij}^2$, **feasibility easier to satisfy**

  ...will be useful later on

Focus on Push-and-Pull objective

When SDP solvers hit their size limit

- **SDP solver: technological bottleneck**
- **How can we best use an LP solver?**
- **Diagonally Dominant (DD) matrices are PSD**
- **Not *vice versa*: inner approximate PSD cone $Y \succeq 0$**
- ***Idea by A.A. Ahmadi [Ahmadi & Hall 2015]***

Diagonally dominant matrices

$n \times n$ **matrix** X **is DD if**

$$\forall i \leq n \quad X_{ii} \geq \sum_{j \neq i} |X_{ij}|.$$

E.g.

$$\begin{pmatrix} 1 & 0.1 & -0.2 & 0 & 0.04 & 0 \\ 0.1 & 1 & -0.05 & 0.1 & 0 & 0 \\ -0.2 & -0.05 & 1 & 0.1 & 0.01 & 0 \\ 0 & 0.1 & 0.1 & 1 & 0.2 & 0.3 \\ 0.04 & 0 & 0.01 & 0.2 & 1 & -0.3 \\ 0 & 0 & 0 & 0.3 & -0.3 & 1 \end{pmatrix}$$

DD Linearization

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} |X_{ij}| \qquad (*)$$

- **introduce "sandwiching" variable T**
- **write $|X|$ as T**
- **add constraints $-T \le X \le T$**
- **by $\ge$ constraint sense, write $(*)$ as**

$$X_{ii} \ge \sum_{j \ne i} T_{ij}$$

DD Programming (DDP)

$$\left.\begin{array}{rcl}\forall\{i,j\}\in E \quad X_{ii}+X_{jj}-2X_{ij} & = & d_{ij}^2 \\ X & \textbf{is} & \textbf{DD}\end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{lrcl}\forall\{i,j\}\in E & X_{ii}+X_{jj}-2X_{ij} & = & d_{ij}^2 \\ \forall i\le n+K & \sum\limits_{\substack{j\le n+K \\ j\neq i}} T_{ij} & \le & X_{ii} \\ & -T\le X & \le & T\end{array}\right.$$

DDP formulation for the DGP

$$\left.\begin{array}{rrcl}
\min & \sum\limits_{\{i,j\}\in E} (X_{ii} + X_{jj} - 2X_{ij}) & & \\
\forall \{i,j\} \in E & X_{ii} + X_{jj} - 2X_{ij} & \ge & d_{ij}^2 \\
\forall i \le n+K & \sum\limits_{\substack{j \le n+K \\ j \ne i}} T_{ij} & \le & X_{ii} \\
& -T \le X & \le & T \\
& T & \ge & 0
\end{array}\right\}$$

SDP vs. DDP: tests

Using "push-and-pull" objective in SDP
SDP solved with Mosek, DDP with CPLEX

SDP + PCA

	SDP			DDP		
Instance	*LDE*	*MDE*	*CPU modl/soln*	*LDE*	*MDE*	*CPU modl/soln*
`C0700odd.1`	0.79	0.34	0.06/**0.12**	**0.38**	**0.30**	0.15/0.15
`C0700.odd.G`	2.38	0.89	0.57/1.16	**1.86**	**0.58**	1.11/**0.95**
`C0150alter.1`	**1.48**	**0.45**	0.73/1.33	1.54	0.55	1.23/**1.04**
`C0080create.1`	2.49	0.82	1.63/7.86	**0.98**	**0.67**	3.39/**4.07**
`1guu-1`	**0.50**	**0.15**	6.67/684.89	1.00	0.85	37.74/**153.17**

Concentration of measure

From [Barvinok, 1997]

> *The value of a "well behaved" function at a random point of a "big" probability space X is "very close" to the mean value of the function.*

and

> *In a sense, measure concentration can be considered as an extension of the law of large numbers.*

Concentration of measure

Given Lipschitz function $f : X \to \mathbb{R}$ s.t.

$$\forall x, y \in X \quad |f(x) - f(y)| \leq L\|x - y\|_2$$

for some $L \geq 0$, there is *concentration of measure* if $\exists$ constants c, C s.t.

$$\forall \varepsilon > 0 \quad \mathsf{P}_x(|f(x) - \mathsf{E}(f)| > \varepsilon) \leq c\, e^{-C\varepsilon^2/L^2}$$

$\equiv$ *"discrepancy from mean is unlikely"*

Barvinok's theorem

Consider:

- for each $k \leq m$, manifolds $\mathcal{X}_k = \{x \in \mathbb{R}^n \mid x^\top Q^k x = a_k\}$
- a feasibility problem $x \in \bigcap_{k \leq m} \mathcal{X}_k$
- its SDP relaxation $\forall x \leq m \ (Q^k \bullet X = a_k)$ with soln. $\bar{X}$

Let $\boxed{T = \mathsf{factor}(\bar{X})}$, $\boxed{y \sim \mathcal{N}^n(0,1) \text{ and } x' = Ty}$

Then $\exists c$ **and** $n_0 \in \mathbb{N}$ **s.t. if** $n \geq n_0$,

$$\mathsf{Prob}\left(\forall k \leq m \ \mathsf{dist}(x', \mathcal{X}_k) \leq c\sqrt{\|\bar{X}\|_2 \ln n}\right) \geq 0.9.$$

IDEA: since x' is "close" to each $\mathcal{X}_k$, try local descent!

Application to the DGP

- $\forall \{i,j\} \in E \quad \mathcal{X}_{ij} = \{x \mid \|x_i - x_j\|_2^2 = d_{ij}^2\}$
- **DGP can be written as** $\bigcap\limits_{\{i,j\} \in E} \mathcal{X}_{ij}$
- **SDP relaxation** $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}^2 \wedge X \succeq 0$ **with soln.** $\bar{X}$

- **Difference with Barvinok:** $x \in \mathbb{R}^{Kn}$, **rk**$(\bar{X}) \leq K$
- **IDEA: sample** $y \sim \mathcal{N}^{nK}(0, \frac{1}{\sqrt{K}})$
- **Thm. Barvinok's theorem works in rank** K

The heuristic

1. **Solve SDP relaxation of DGP, get soln.** $\bar{X}$
   use DDP+LP if SDP+IPM too slow

2. **a.** $T = \mathsf{factor}(\bar{X})$
   b. $y \sim \mathcal{N}^{nK}(0, \frac{1}{\sqrt{K}})$
   c. $x' = Ty$

3. **Use x' as starting point for a local NLP solver on formulation**

$$\min_x \sum_{\{i,j\} \in E} \left(\|x_i - x_j\|^2 - d_{ij}^2 \right)^2$$

and return improved solution x

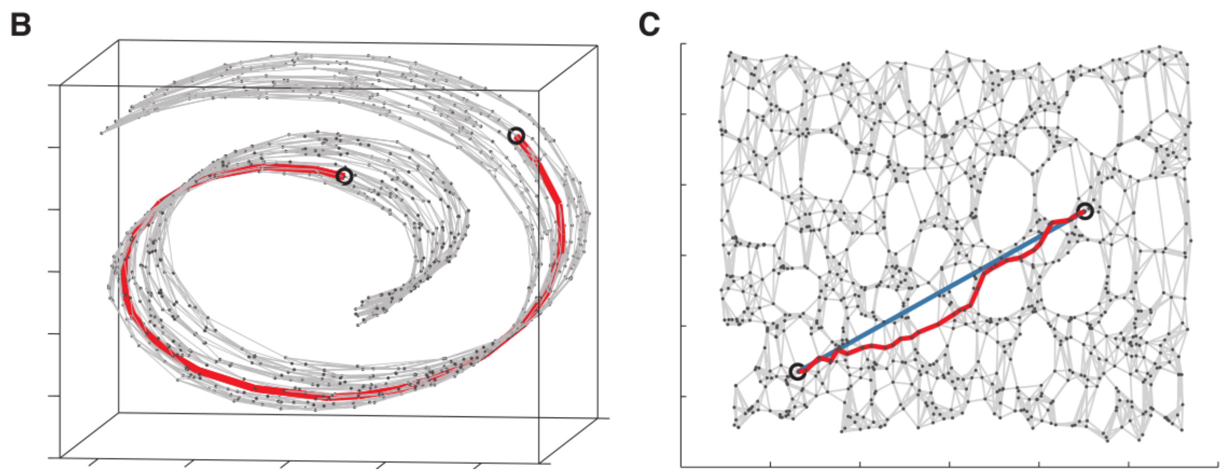
SDP+Barvinok vs. DDP+Barvinok

	SDP			DDP		
Instance	*LDE*	*MDE*	*CPU*	*LDE*	*MDE*	*CPU*
C0700odd.1	**0.00**	**0.00**	**0.63**	**0.00**	**0.00**	1.49
C0700.odd.G	**0.00**	**0.00**	**21.67**	0.42	0.01	30.51
C0150alter.1	**0.00**	**0.00**	**29.30**	**0.00**	**0.00**	34.13
C0080create.1	**0.00**	**0.00**	**139.52**	**0.00**	**0.00**	141.49
1b03	**0.18**	0.01	132.16	0.38	0.05	**101.04**
1crn	0.78	**0.02**	800.67	**0.76**	0.04	**522.60**
1guu-1	**0.79**	**0.01**	1900.48	0.90	0.04	**667.03**

Most of the CPU time taken by local NLP solver

Isomap for DG

1. Let D' be the (square) weighted adjacency matrix of G
2. Complete D' to *approximate* sqEDM $\tilde{D}$
3. Let $\tilde{B} = -(1/2)J\tilde{D}J$, where $J = I - (1/n)\mathbf{1}\mathbf{1}^\top$
4. Find eigenval/vects Λ, P so $\tilde{B} = P^\top \Lambda P$
5. Keep $\leq K$ largest nonneg. eigenv. of Λ to get $\tilde{\Lambda}$ (MDS/PCA)
6. Let $\tilde{x} = P^\top \sqrt{\tilde{\Lambda}}$



Vary Step 2 to generate Isomap heuristics

Why it works

- G represented by weighted adjacency matrix D'
- do not know D, approximate to $\tilde{D}$ not sqEDM
- $\Rightarrow$ get $\tilde{B}$, not generally Gram
- $\leq K$ largest nonnegative eigenvalues
  $\Rightarrow$ "closest PSD matrix" B' to $\tilde{B}$ having rank $\leq K$
- Factor it to get $\tilde{x} \in \mathbb{R}^{Kn}$

Variants for Step 2

A. **Floyd-Warshall all-shortest-paths algorithm on G**
(classic Isomap)

B. **Find a spanning tree (SPT) of G and compute a random realization in $\bar{x} \in \mathbb{R}^K$, use its sqEDM**

C. **Solve a push-and-pull SDP relaxation to find a realization $\bar{x} \in \mathbb{R}^n$, use its sqEDM**

D. Solve an SDP relaxation with Barvinok objective to find $\bar{x} \in \mathbb{R}^r$ (with $r \leq \lfloor(\sqrt{8|E|+1}-1)/2\rfloor$), use its sqEDM
haven't really talked about this, sorry

Post-processing: $\tilde{x}$ *as starting point for NLP descent in GO formulation*

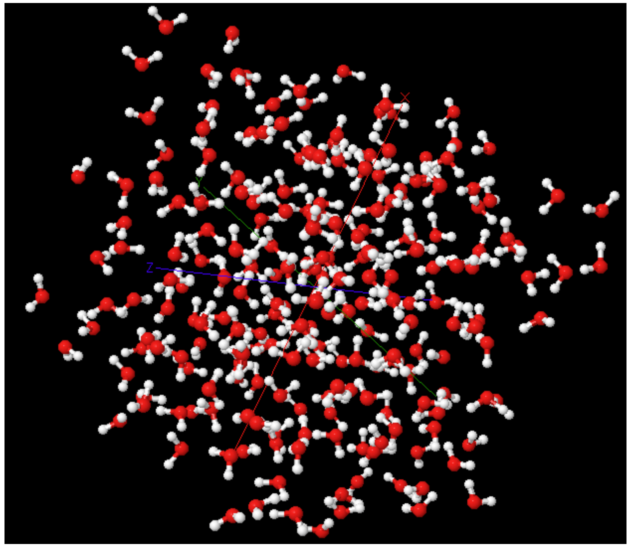
Results

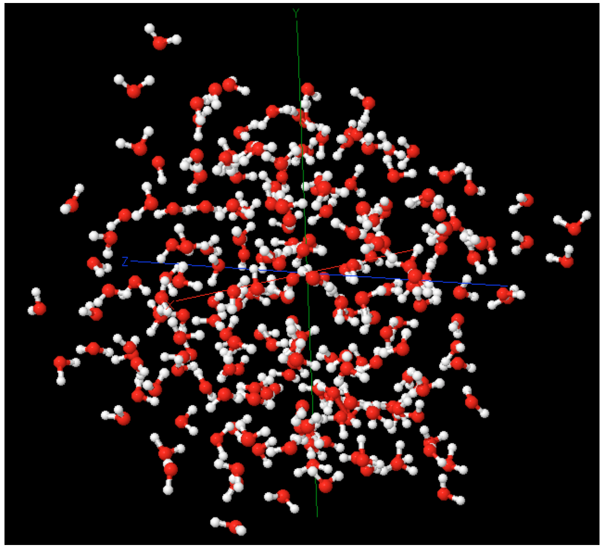
Comparison with `dgsol` [Moré, Wu 1997]

Instance			mde						lde						CPU					
Name	n	$\|E\|$	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol	Isomap	IsoNLP	SPT	SDP	Barvinok	DGSol
C0700odd.1	15	39	0.585	0.001	0.190	0.068	**0.000**	0.135	0.989	0.004	0.896	0.389	**0.001**	0.634	**0.002**	1.456	1.589	0.906	1.305	1.747
C0700odd.2	15	39	0.599	**0.000**	0.187	0.086	**0.000**	0.128	0.985	**0.002**	0.956	0.389	0.009	1.000	**0.003**	1.376	1.226	1.002	1.063	0.887
C0700odd.3	15	39	0.599	**0.000**	0.060	0.086	**0.000**	0.128	0.985	**0.002**	0.326	0.389	0.009	1.000	**0.003**	1.259	1.256	0.861	1.167	0.877
C0700odd.4	15	39	0.599	**0.000**	0.283	0.086	0.001	0.128	0.985	**0.002**	2.449	0.389	0.008	1.000	**0.003**	1.347	1.222	0.976	1.063	1.033
C0700odd.5	15	39	0.599	**0.000**	0.225	0.086	0.000	0.128	0.985	**0.002**	0.867	0.389	0.007	1.000	**0.003**	1.284	1.157	0.987	1.100	0.700
C0700odd.6	15	39	0.599	**0.000**	0.283	0.086	0.000	0.128	0.985	**0.002**	1.520	0.389	**0.002**	1.000	**0.002**	1.372	1.196	0.998	1.305	0.909
C0700odd.7	15	39	0.585	0.001	0.080	0.068	**0.000**	0.135	0.989	0.004	0.361	0.389	**0.001**	0.634	**0.003**	1.469	1.322	0.894	1.093	1.719
C0700odd.8	15	39	0.585	0.001	0.056	0.068	**0.000**	0.135	0.989	0.004	0.275	0.389	**0.003**	0.634	**0.003**	1.408	1.306	0.692	1.079	1.744
C0700odd.9	15	39	0.585	0.001	0.057	0.068	**0.000**	0.135	0.989	0.004	0.301	0.389	**0.002**	0.634	**0.002**	1.430	1.172	0.791	1.093	1.745
C0700odd.A	15	39	0.585	0.001	0.043	0.068	**0.000**	0.135	0.989	**0.004**	0.316	0.389	**0.004**	0.634	**0.002**	1.294	1.269	0.722	1.220	1.523
C0700odd.B	15	39	0.585	0.001	0.151	0.068	**0.000**	0.135	0.989	**0.004**	1.022	0.389	**0.004**	0.634	**0.002**	1.297	1.279	0.871	1.111	1.747
C0700odd.C	15	39	0.835	**0.022**	0.033	0.039	0.031	0.025	1.012	**0.147**	0.393	0.211	0.294	0.167	**0.004**	6.803	6.369	7.371	7.030	7.000
C0700odd.D	36	242	0.835	**0.022**	0.041	0.039	0.042	0.025	1.012	**0.147**	0.423	0.211	0.268	0.167	**0.006**	6.806	6.575	7.422	7.603	7.095
C0700odd.E	36	242	0.835	**0.022**	0.064	0.039	0.031	0.025	1.012	**0.147**	0.894	0.211	0.260	0.167	**0.006**	6.911	6.638	7.365	6.979	7.008
C0700odd.F	36	242	0.599	**0.000**	0.047	0.086	**0.000**	0.128	0.985	**0.002**	0.308	0.389	0.005	1.000	**0.002**	1.299	1.310	1.008	1.100	1.040
C0150alter.1	37	335	0.786	0.058	0.066	0.014	0.015	**0.010**	0.992	0.571	0.693	0.256	0.285	**0.253**	**0.004**	9.492	9.456	10.276	10.120	9.272
C0080create.1	60	681	0.887	0.053	0.083	**0.024**	**0.024**	0.054	1.967	0.949	0.789	**0.511**	0.516	0.718	**0.012**	18.835	19.720	21.247	20.906	19.962
C0080create.2	60	681	0.887	0.053	0.047	**0.024**	**0.024**	0.054	1.967	0.949	0.585	**0.511**	0.512	0.718	**0.008**	18.791	20.009	21.728	20.885	19.740
C0020pdb	107	999	0.939	0.110	0.119	**0.059**	0.060	0.103	1.242	1.113	1.349	1.082	1.138	**0.798**	**0.035**	29.024	27.772	35.273	35.486	32.479
1guu	150	955	0.986	0.068	0.069	**0.057**	**0.057**	0.061	0.999	0.854	0.830	**0.735**	0.751	0.768	**0.048**	30.869	28.784	41.488	41.852	37.848
1guu-1	150	959	0.986	0.061	0.063	0.058	**0.057**	0.060	1.000	**0.711**	0.855	0.805	0.829	0.778	**0.053**	31.322	31.442	42.308	41.590	37.218
1guu-4000	150	968	0.974	0.081	0.080	0.072	**0.065**	0.079	1.000	0.901	**0.728**	0.760	0.961	0.826	**0.050**	30.352	29.856	42.330	39.832	42.015
C0030pkl	198	3247	0.961	0.112	0.160	**0.076**	0.077	0.137	**1.197**	1.354	2.230	1.995	2.054	1.401	**0.091**	105.175	104.775	149.192	146.360	111.859
1PPT	302	3102	0.984	**0.121**	0.129	0.128	0.129	0.123	**1.000**	1.519	1.219	1.944	1.956	1.224	**0.356**	112.448	110.345	185.815	187.182	118.681
100d	488	5741	0.987	0.146	0.146	0.155	0.157	**0.137**	**1.000**	1.577	1.397	1.764	1.749	1.358	**0.828**	229.809	213.136	659.638	659.280	233.115
GeoMean			0.74	**0.00**	0.09	0.06	**0.00**	0.08	1.07	**0.04**	0.73	0.50	0.06	0.66	**0.01**	6.30	6.04	5.93	6.63	6.30
Avg			0.76	0.04	0.11	0.07	**0.03**	0.10	1.09	**0.44**	0.88	0.63	0.47	0.77	**0.06**	26.12	25.21	49.69	49.55	27.96
StDev			0.17	0.05	0.07	**0.03**	0.04	0.04	**0.27**	0.55	0.57	0.52	0.65	0.34	**0.18**	51.69	48.82	135.08	134.97	53.26

Large instances

Instance			mde		lde		CPU	
Name	$\|V\|$	$\|E\|$	IsoNLP	dgsol	IsoNLP	dgsol	IsoNLP	dgsol
water	648	11939	**0.005**	0.15	**0.557**	0.81	26.98	**15.16**
3al1	678	17417	0.036	**0.007**	0.884	**0.810**	**170.91**	210.25
1hpv	1629	18512	**0.074**	0.078	0.936	**0.932**	374.01	**60.28**
il2	2084	45251	**0.012**	0.035	**0.910**	0.932	465.10	**139.77**
1tii	5684	69800	0.078	**0.077**	0.950	**0.897**	7400.48	**454.375**





THE END