# Évaluation expérimentale de librairies polyhédrales

## Projet 2004

Duong Nguyen, Corinne Ancourt, François Irigoin

École des Mines de Paris - Centre de Recherche en Informatique

8 novembre 2006

# Les questions posées

- Choix des meilleurs algorithmes
- Choix des meilleurs implantations
- Impact des exceptions sur la précision : temps, espace, magnitude

# Les questions posées

- Choix des meilleurs algorithmes
- Choix des meilleurs implantations
- Impact des exceptions sur la précision : temps, espace, magnitude

# Les questions posées

- Choix des meilleurs algorithmes
- Choix des meilleurs implantations
- Impact des exceptions sur la précision : temps, espace, magnitude

# Exemple de dérive en magnitude

```
I = 1, J = 0, K = 0
DO WHILE(K.LT.100)
   K = K + J
   J = J + I
   I = I + 1
ENDDO
PRINT *, I, J, K
IF(X.GT.0.) THEN
   READ *, J, K
   PRINT *, I, J, K
ELSEIF(Y.GT.0.) THEN
   READ *, I, K
   PRINT *, I, J, K
ELSE
   READ *, I, J
   PRINT *, I, J, K
ENDIF
```

# Exemple de dérive en magnitude (suite)

```
      IF (X.GT.0.) THEN
         READ *, J, K
         PRINT *, I, J, K
      ELSE
         IF (Y.GT.0.) THEN
            READ *, I, K
C  first iteration:
C  P(I,J,K) {2<=J}
C  second iteration:
C  P(I,J,K) {3<=J}
C  third iteration:
C  P(I,J,K) {6<=J, J<=300}
C  fourth iteration:
C  P(I,J,K) {}

            PRINT *, I, J, K
         ELSE
            READ *, I, J
            PRINT *, I, J, K
         ENDIF
```

# Exemple de dérive en magnitude (suite 2)

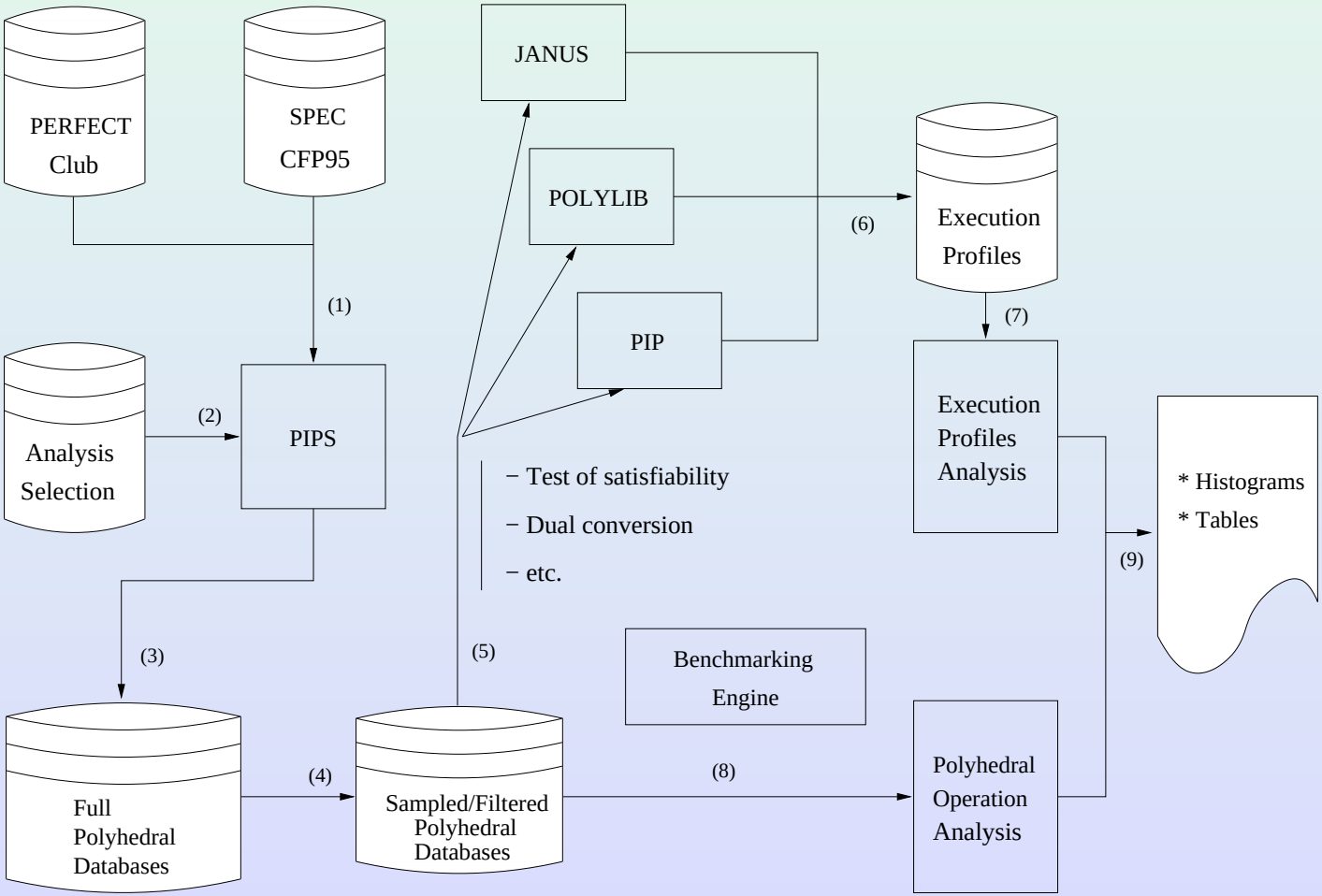
```
      I = 1,  J = 0, K = 0
C  first iteration:
C  T(I,J,K) {I#init<=I}
C  second iteration:
C  T(I,J,K) {I#init<=I, I+J#init<=I#init+J}
C  third iteration:
C  T(I,J,K) {I#init<=I, I+J#init<=I#init+J,
C    6I#init+3J+K#init<=6I+3J#init+K, I#init+J+K#init<=I+J#init+K}
C  fourth iteration:
C  T(I,J,K)
C    {1379460I#init+895055J+454903K#init<=1379460I+895055J#init+454903
C    K, 1063137I#init+639920J+364213K#init<=1063137I+639920J#init+
C    364213K, 6748I#init+1469J#init+479K<=6748I+1469J+479K#init,
C    1802I#init+899J+299K#init<=1802I+899J#init+299K,
C    287I#init+10J#init+7K<=287I+10J+7K#init,
C    41I#init+5J#init+2K<=41I+5J+2K#init,
C    10I#init+4J+K#init<=10I+4J#init+K,
C    2622I+2622J#init+263K<=2622I#init+2622J+263K#init,
C    28497I+28497J#init+109K#init<=28497I#init+28497J+109K,
C    30061I+30061J#init+673K<=30061I#init+30061J+673K#init}

      DO WHILE (K.LT.100)
```

# Enveloppe convexe problématique

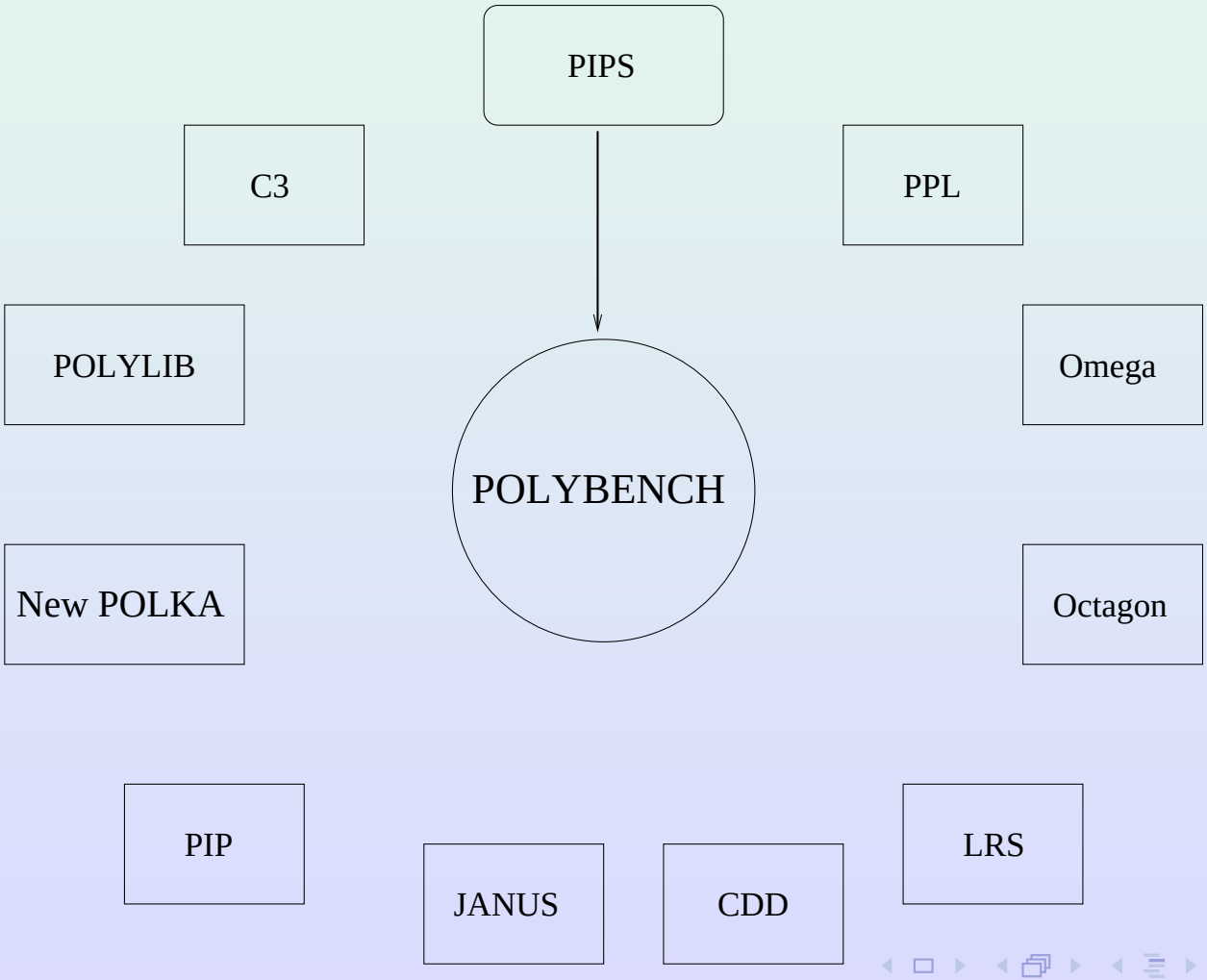
```
#DIMENSION: (69)  INEGALITES(72)  EGALITES (13)
VAR N2M#new, LPN#new, LZN#new, N2P#new, N1H#new, N2#new, MEMSIZ#new,
NWH#new, NWEIG#new, LEIG#new, NW#new, NWQ#new, LZX#new, LPY#new,
LZY#new, LZO#new, NUMBER#new, NXXXIN#new, NXXXIN#init, NPTS#new,
NPTS#init, NSKIP#new, NSKIP#init, MTRN#new, MTRN#init, MSKIP#new,
MSKIP#init, ISIGN#new, ISIGN#init, NXLOG2#new, NXLOG2#init,
NFTVMT#new, NFTVMT#init, NXACAC#new, NXACAC#init, NXXOUT#new,
NXXOUT#init, KZN#new, KZN#init, KZO#new, KZO#init, K#new, K#init,
NUSHUF#new, NUSHUF#init, NXXCSR#new, NXXCSR#init, NXSCSC#new,
NXSCSC#init, NXPRNT#new, NXPRNT#init, ZETAPH:NCALL#new,
ZETAPH:NCALL#init, KPN#new, KPN#init, NXXRCS#new, NXXRCS#init,
TEMPHY:NCALL#new, TEMPHY:NCALL#init, LVECT#new, LVECT#init, LSKIP#new,
LSKIP#init, NVECT#new, NVECT#init, NVSKIP#new, NVSKIP#init,
NSTEPS#new, NSTEPS#init
  {
- NSTEPS#new - TEMPHY:NCALL#init + NSTEPS#init + TEMPHY:NCALL#new  <= 0 ,
- NSTEPS#new + NSTEPS#init  <= -1 ,
...
  }

#DIMENSION: (69)  INEGALITES (0)  EGALITES (26)
  {
- NXXXIN#new + NXXXIN#init  == 0 ,
- NPTS#new + NPTS#init  == 0 ,
...
  }
```

# Première partie
# Environnement de benchmarking

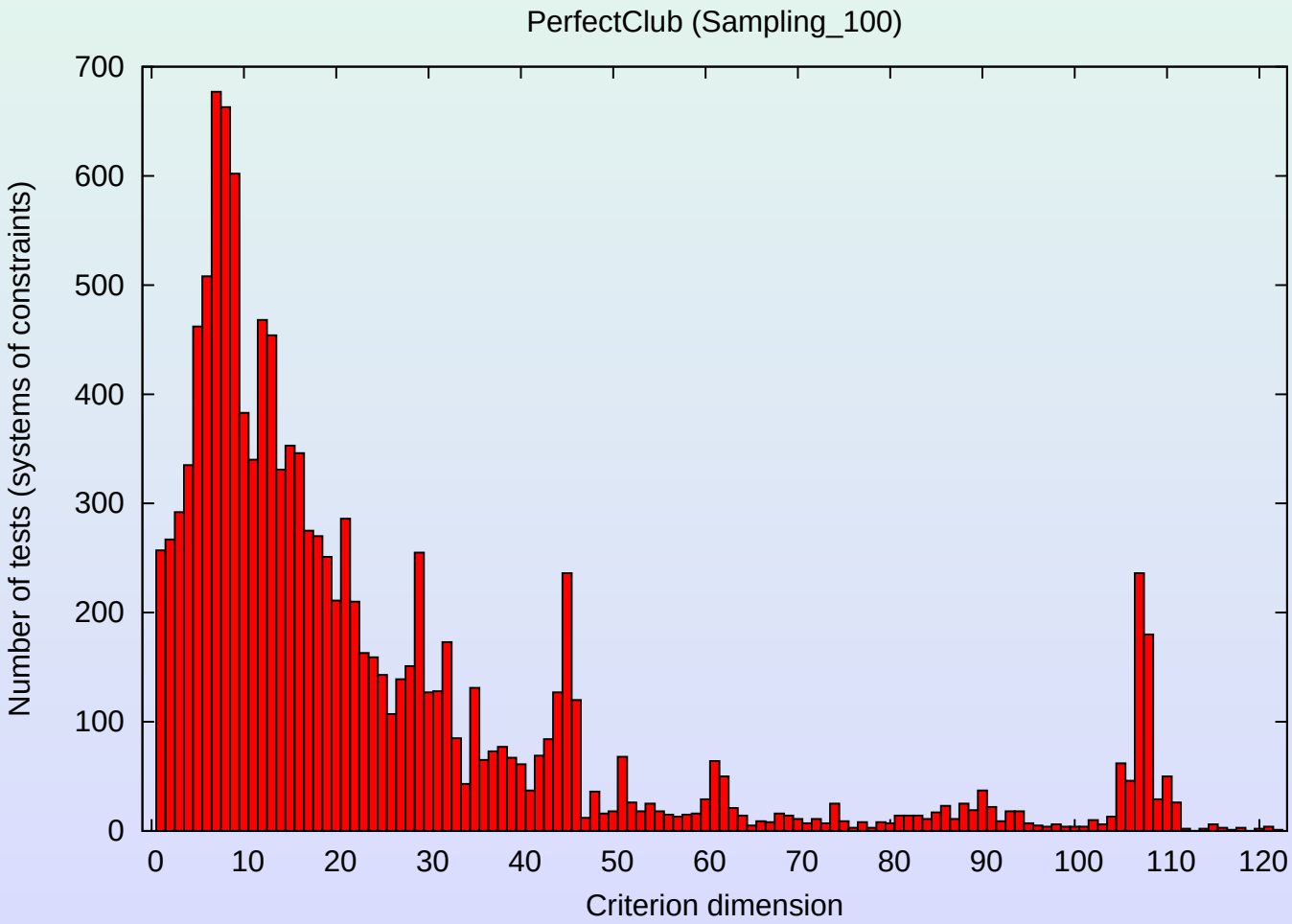
# Polybench


PERFECT Club
SPEC CFP95
JANUS
POLYLIB
PIP
(6)
Execution Profiles
(7)
(1)
Analysis Selection
(2)
PIPS
Execution Profiles Analysis
* Histograms
* Tables
(9)
− Test of satisfiability
− Dual conversion
− etc.
(3)
(5)
Benchmarking Engine
Full Polyhedral Databases
(4)
Sampled/Filtered Polyhedral Databases
(8)
Polyhedral Operation Analysis

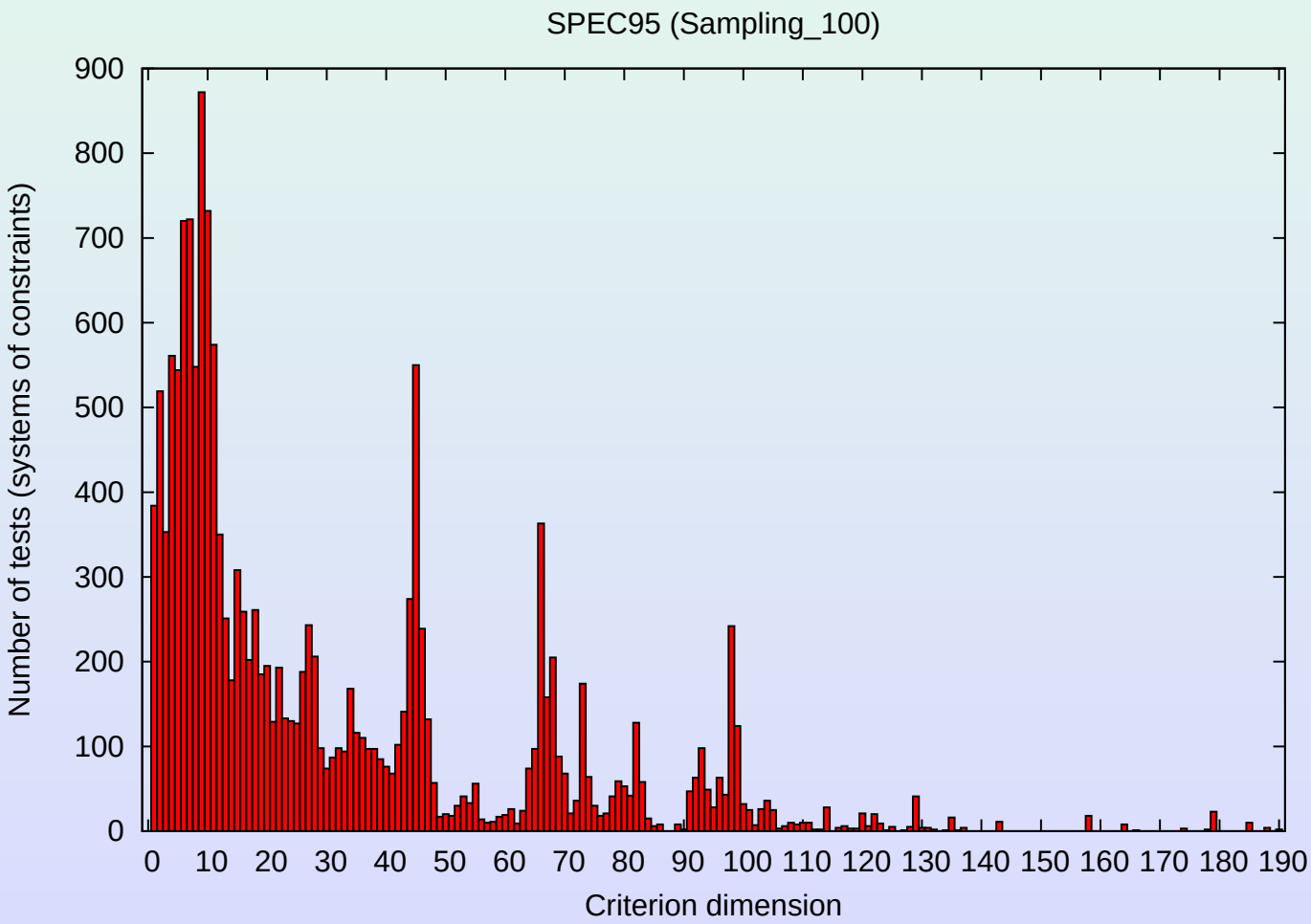
# Bibliothèques évaluées



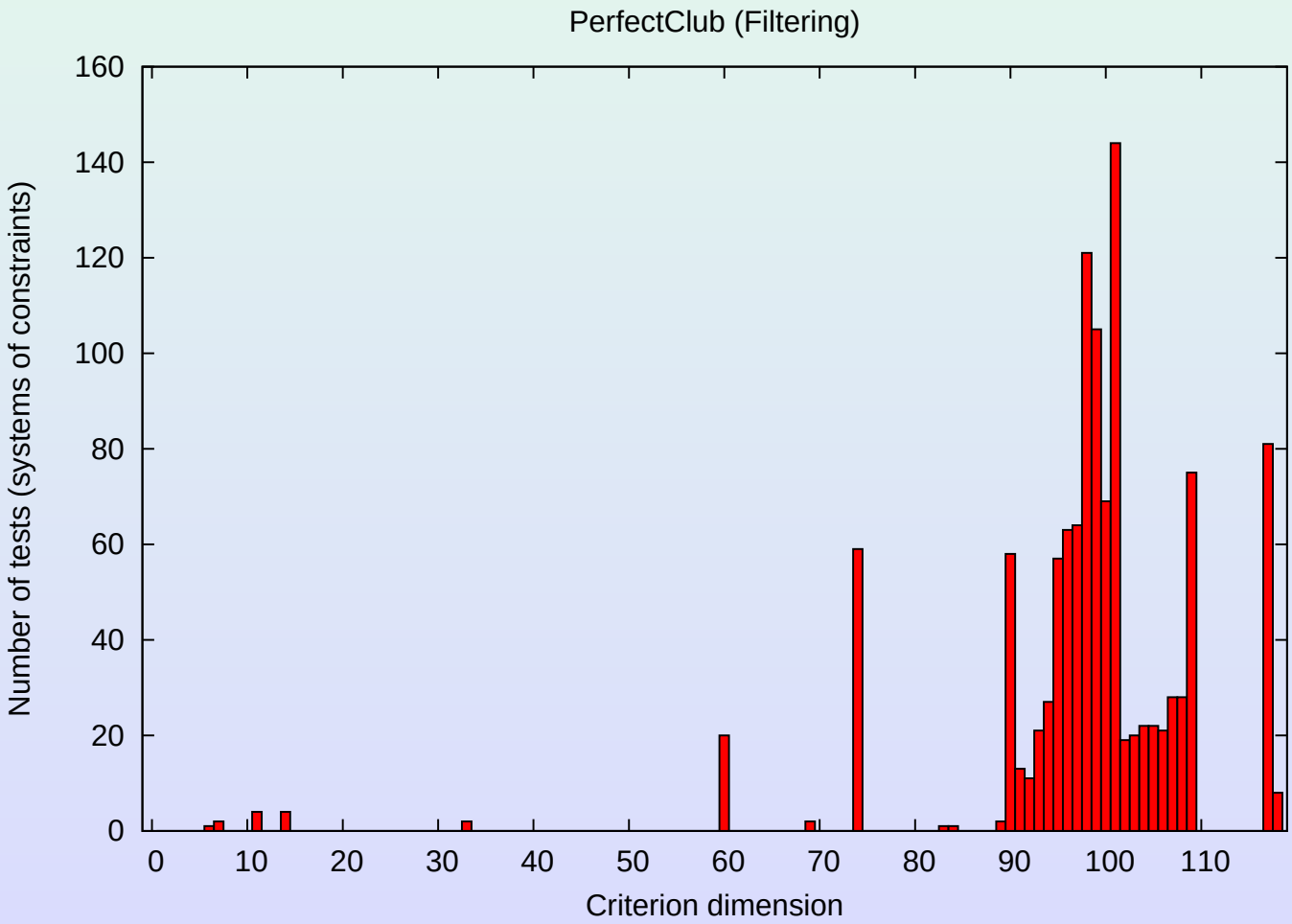
# Premier benchmark : PerfectClub (satisfiabilité)



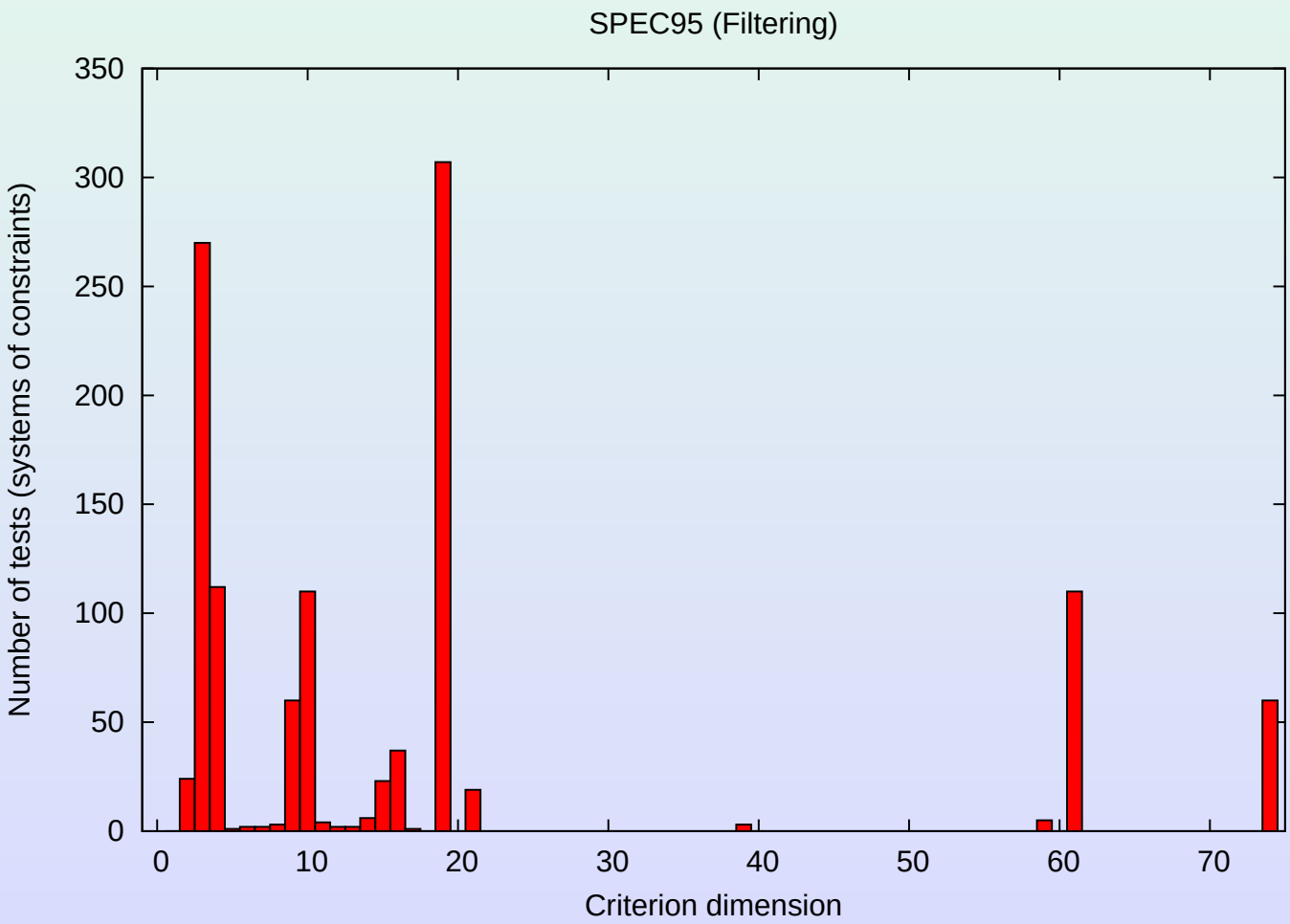
# Deuxième benchmark : SPEC95 (satisfiabilité)



# Troisième espèce de benchmarks : ensembles filtrés



# Troisième espèce de benchmarks : ensembles filtrés (suite)



# Deuxième partie
# Satisfiabilité

# Les algorithmes de satisfiabilité

- JANUS 64 bits (JV64, INRIA, Jean-Claude Sogno) : prétraitement, simplexe
- Simplexe (LS64, Mines)
- Fourier-Motzkin (FM64, Mines)
- Passage au dual (FDD64, ???)
- Exécution en parallèle des quatre algorithmes (PA)

# Les algorithmes de satisfiabilité

- JANUS 64 bits (JV64, INRIA, Jean-Claude Sogno) : prétraitement, simplexe
- Simplexe (LS64, Mines)
- Fourier-Motzkin (FM64, Mines)
- Passage au dual (FDD64, ? ? ?)
- Exécution en parallèle des quatre algorithmes (PA)

# Les algorithmes de satisfiabilité

- JANUS 64 bits (JV64, INRIA, Jean-Claude Sogno) : prétraitement, simplexe
- Simplexe (LS64, Mines)
- Fourier-Motzkin (FM64, Mines)
- Passage au dual (FDD64, ? ? ?)
- Exécution en parallèle des quatre algorithmes (PA)

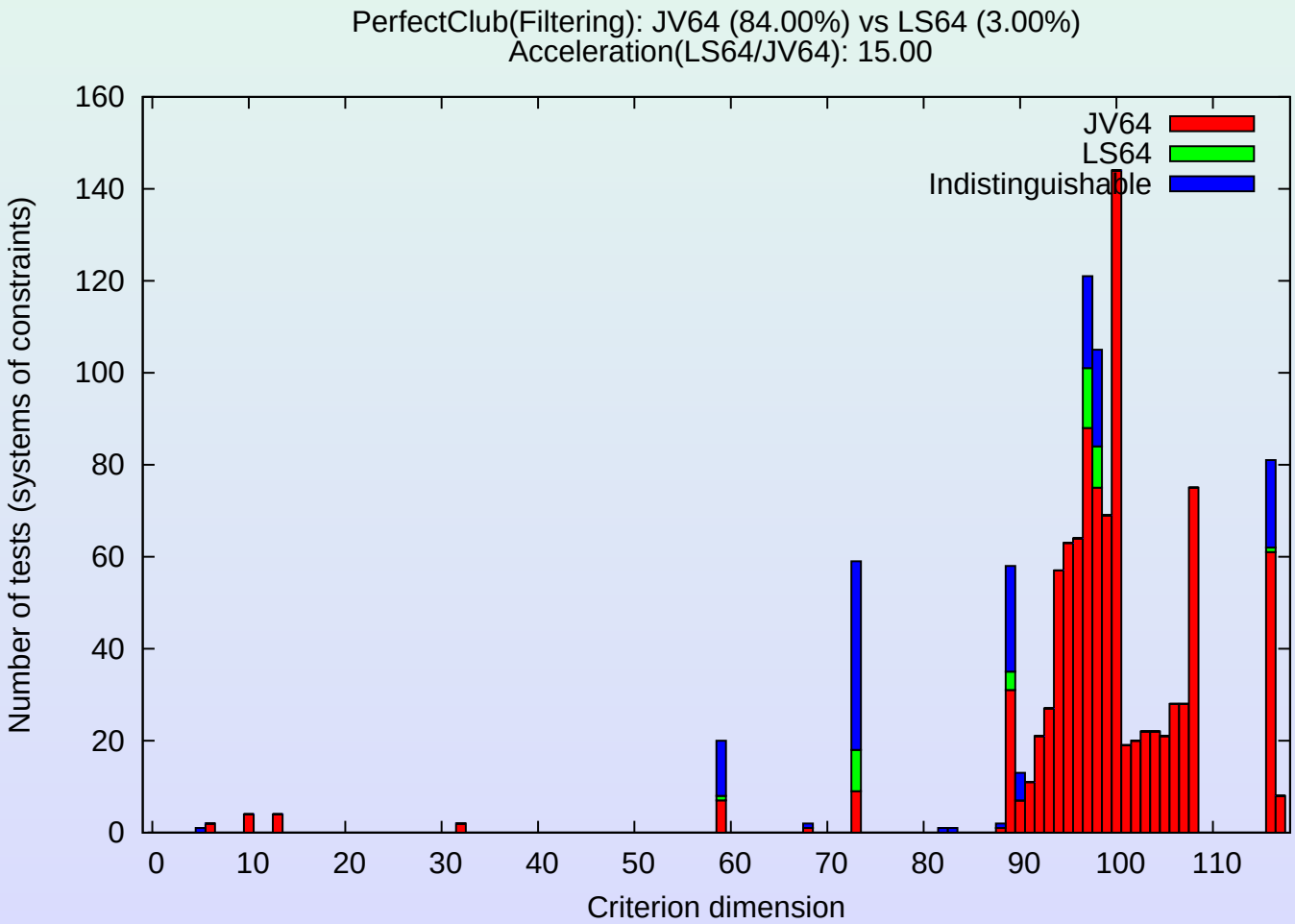
# Les algorithmes de satisfiabilité

- JANUS 64 bits (JV64, INRIA, Jean-Claude Sogno) : prétraitement, simplexe
- Simplexe (LS64, Mines)
- Fourier-Motzkin (FM64, Mines)
- Passage au dual (FDD64, ? ? ?)
- Exécution en parallèle des quatre algorithmes (PA)

# Les algorithmes de satisfiabilité

- JANUS 64 bits (JV64, INRIA, Jean-Claude Sogno) : prétraitement, simplexe
- Simplexe (LS64, Mines)
- Fourier-Motzkin (FM64, Mines)
- Passage au dual (FDD64, ? ? ?)
- Exécution en parallèle des quatre algorithmes (PA)

# Exemple de résultat partiel : Janus contre Simplexe



# Temps d'exécution (PerfectClub)

| 64-bit | **JANUS/PA** | **LS/PA** | **FM/PA** | **FDD/PA** |
|---|---|---|---|---|
| *Sampled* | 1/1 | 22/1 | 46/1 | 404/1 |
| *Filtered* | 1/1 | 15/1 | 32/1 | 520/1 |

# Exceptions (échantillonage aléatoire à 1%)

| | PerfectClub | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#overflows** | **#timeouts** | **#operations** |
| *JANUS 64-bit* | 5 | 0 | 12668 |
| *$C^3$ Simplex 64-bit* | 9 | 0 | 12668 |
| *$C^3$ Fourier-Motzkin* | 0 | 2 | 12668 |
| *$C^3$ Double Description* | 2 | 7 | 12668 |

| | SPEC95 | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#overflows** | **#timeouts** | **#operations** |
| *JANUS 64-bit* | 52 | 0 | 16303 |
| *$C^3$ Simplex 64-bit* | 51 | 0 | 16303 |
| *$C^3$ Fourier-Motzkin* | 1 | 14 | 16303 |
| *$C^3$ Double Description* | 717 | 0 | 16303 |

# Exceptions (cas difficiles)

| | PerfectClub | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#overflows** | **#timeouts** | **#operations** |
| *JANUS 64-bit* | 0 | 0 | 1310 |
| *$C^3$ Simplex 64-bit* | 2 | 0 | 1310 |
| *$C^3$ Fourier-Motzkin* | 0 | 0 | 1310 |
| *$C^3$ Double Description* | 0 | 407 | 1310 |

| | SPEC95 | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#overflows** | **#timeouts** | **#operations** |
| *JANUS 64-bit* | 2987 | 0 | 4676 |
| *$C^3$ Simplex 64-bit* | 3714 | 0 | 4676 |
| *$C^3$ Fourier-Motzkin* | 8 | 626 | 4676 |
| *$C^3$ Double Description* | 22 | 0 | 4676 |

# Différences entre benchmarks sur les cas difficiles

| Filtered databases | **PerfectClub** | **SPEC95** |
|---|---|---|
| *ASCII file size* | 5596.73 | 2496.27 |
| Average #Dimensions | 46.97 | 10.28 |
| Average #Equations | 17.40 | 2.35 |
| Average #Inequalities | 38.89 | 52.56 |
| Average #Constraints | 56.30 | 54.91 |
| Average #Sparsity index | 3.90 | 3.86 |
| Average #Vertices | 1118.76 | 18.66 |
| Average #Rays | 69.10 | 14.07 |
| Average #Lines | 6.78 | 0.53 |

# Précision des algorithmes (échantillonage à 1%)

| | PerfectClub | |
|---|---|---|
| **integer vs rational** | **declared feasible** | **#operations** |
| *JANUS 64-bit* | 0 | 12668 |
| *$C^3$ Simplex 64-bit* | 5 | 12668 |
| *$C^3$ Fourier-Motzkin* | 1 | 12668 |
| *$C^3$ Double Description* | 8 | 12668 |

| | SPEC95 | |
|---|---|---|
| **integer vs rational** | **declared feasible** | **#operations** |
| *JANUS 64-bit* | 0 | 16303 |
| *$C^3$ Simplex 64-bit* | 4 | 16303 |
| *$C^3$ Fourier-Motzkin* | 0 | 16303 |
| *$C^3$ Double Description* | 3 | 16303 |

# Entiers sur 32 ou 64 bits (SPEC95, cas difficiles)

| | SPEC95 | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#ovfls** | **#timeouts** | **#ops** |
| $C^3$ *JANUS 64-bit* | 2987 | 0 | 4676 |
| $C^3$ *JANUS 32-bit* | 3109 | 0 | 4676 |
| $C^3$ *Simplex 64-bit* | 3174 | 0 | 4676 |
| $C^3$ *Simplex 32-bit* | 3972 | 0 | 4676 |
| $C^3$ *Fourier-Motzkin 64-bit* | 8 | 625 | 4676 |
| $C^3$ *Fourier-Motzkin 32-bit* | 181 | 495 | 4676 |
| $C^3$ *Double Description 64-bit* | 22 | 0 | 4676 |
| $C^3$ *Double Description 32-bit* | 939 | 0 | 4676 |

# Conclusions sur les algorithmes de satisfiabilité

- JANUS 64 bits est nettement meilleur que tous les autres algorithmes testés
- utilisation du passage au dual : problématique
- stockage des entiers sur 64 bits : un peu plus lent, mais moins d'exceptions
- algorithme parallèle : réduction du nombre d'exceptions

# Conclusions sur les algorithmes de satisfiabilité

- JANUS 64 bits est nettement meilleur que tous les autres algorithmes testés
- utilisation du passage au dual : problématique
- stockage des entiers sur 64 bits : un peu plus lent, mais moins d'exceptions
- algorithme parallèle : réduction du nombre d'exceptions

# Conclusions sur les algorithmes de satisfiabilité

- JANUS 64 bits est nettement meilleur que tous les autres algorithmes testés
- utilisation du passage au dual : problématique
- stockage des entiers sur 64 bits : un peu plus lent, mais moins d'exceptions
- algorithme parallèle : réduction du nombre d'exceptions

# Conclusions sur les algorithmes de satisfiabilité

- JANUS 64 bits est nettement meilleur que tous les autres algorithmes testés
- utilisation du passage au dual : problématique
- stockage des entiers sur 64 bits : un peu plus lent, mais moins d'exceptions
- algorithme parallèle : réduction du nombre d'exceptions

# Troisième partie
# Autres opérateurs

# Algorithmes de projection d'un système de contraintes

- élimination de Fourier-Motzkin (P64)
- projection dans le dual (PDD64)

# Performances des algorithmes de projection

- Fourier-Motzkin est 450 fois plus rapide que la projection par passage au dual pour le PerfectClub et pour SPEC95
- Fourier-Motzkin ne génère pas d'exceptions :

| | PerfectClub | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#ovfl** | **#timeouts** | **#op** |
| *P64* | 0 | 0 | 1789 |
| *PDD64* | 19 | 15 | 1789 |

# Algorithmes de minimisation d'un système de contraintes

- élimination de contraintes redondantes par satisfiabilité (N64)
- recalcul des contraintes par passage au dual (NDD64)

# Performances des algorithmes de minimisation

- la minimisation par satisfiabilité est 450 fois plus rapide que la minimisation par passage au dual pour le PerfectClub et pour SPEC95
- l'utilisation de la satifiabilité ne génère pas d'exceptions :

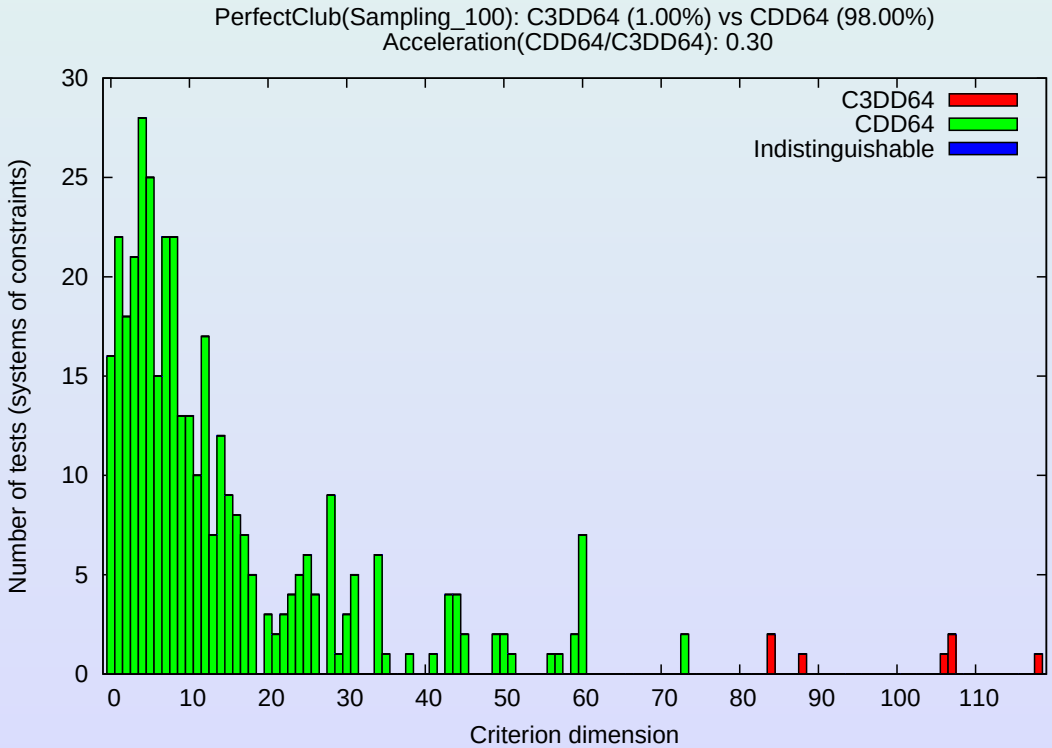
| | PerfectClub | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#ovfl** | **#timeouts** | **#op** |
| *N64-bit* | 0 | 0 | 3894 |
| *NDD64-bit* | 14 | 9 | 3894 |

# Algorithme de passage au dual : deux implantations de Chernikova

- Polylib (IRISA) : C3DD64 (integer/rational)
- CDD64 (floating point ou GNU multiprécision : pas de détection d'overflow)

# Performances des deux implantations

- CDD64 est environ 7 fois plus rapide que C3DD64 sur 1% de la base SPEC95, mais ce n'est pas uniforme :

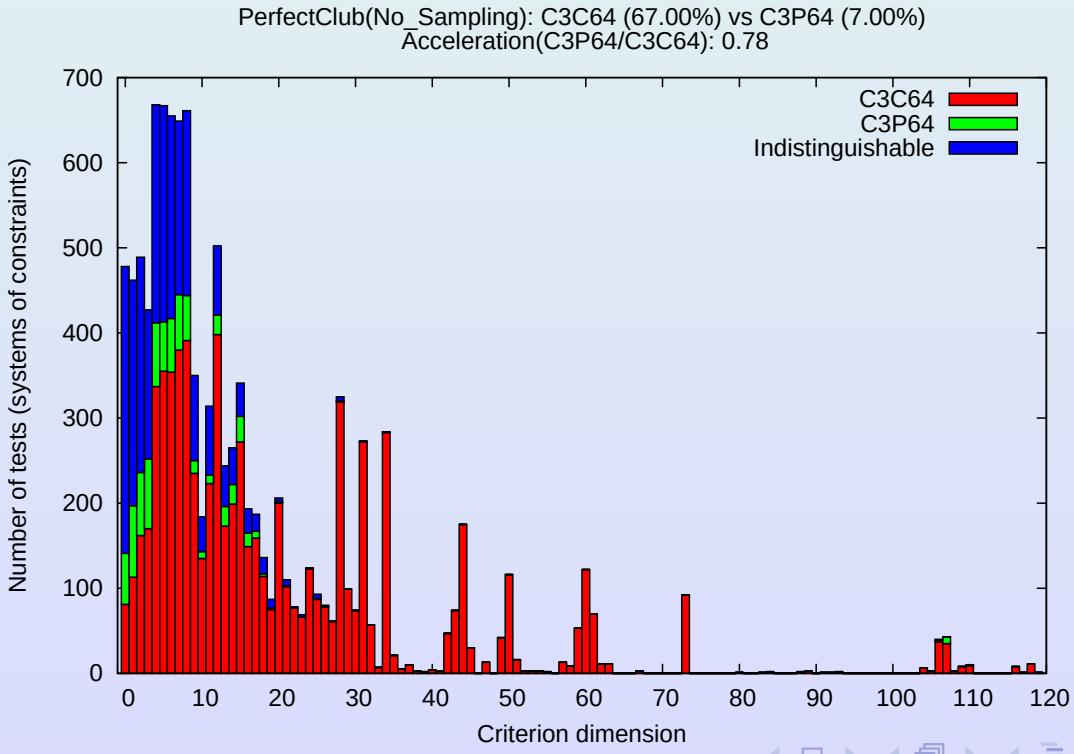


# Trois implantations 64 bits de l'enveloppe convexe

- $C^3$ avec factorisation partielle (Mines) : C3C64
- Polylib (IRISA) : C3P64
- NewPolka (VERIMAG/IRISA) : PK64

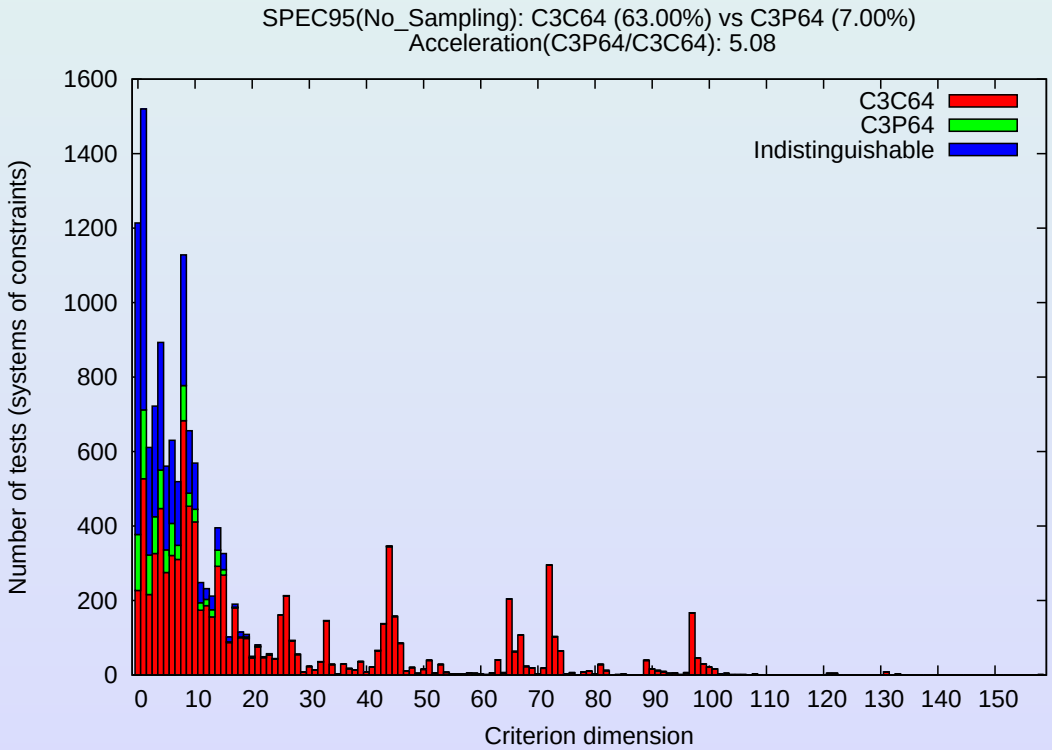
# Performances des deux premières implantations

- la factorisation C3C64 n'est perdante que dans 7% des cas pour le PerfectClub et pour SPEC95
- le gain est de 1.25 sur le PerfectClub intégral



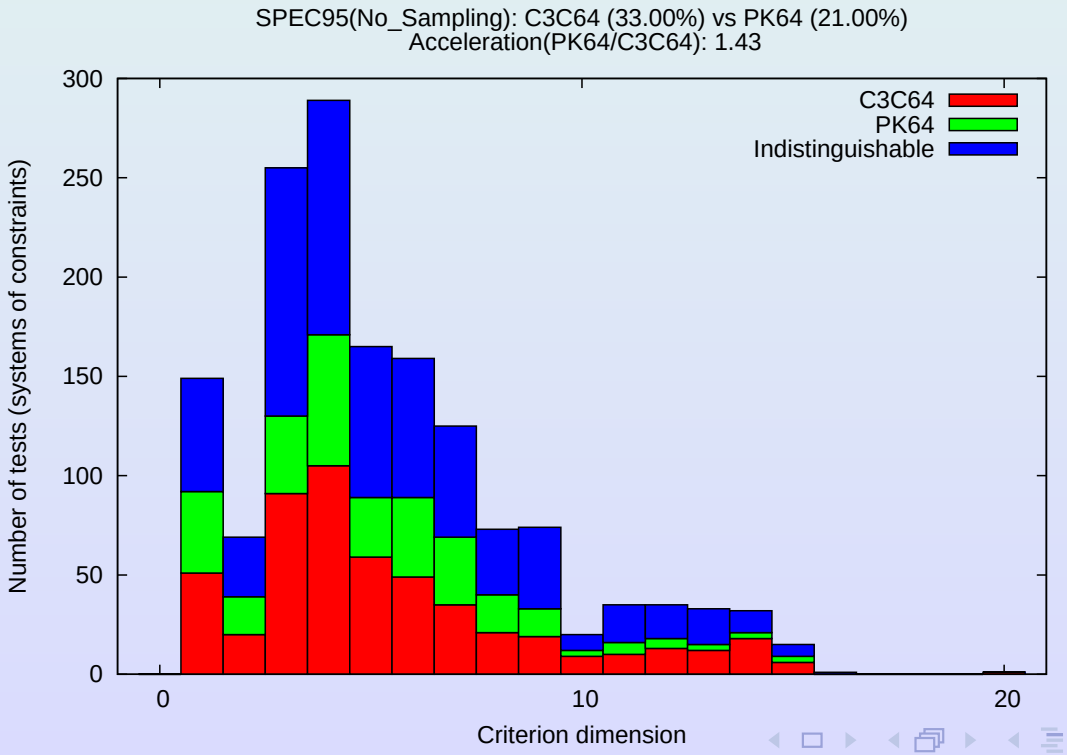
# Performances des deux premières implantations (suite)

- mais le gain est de 5 sur SPEC95 intégral (pb cohérence entre 0.78 et 5.08)



# Performances des deux dernières implantations (C3C64 et PK64)

- La version $C^3$ avec factorisation partielle est 1.48 fois plus rapide que NewPolka sur PerfectClub et 1.43 sur SPEC95



# Exceptions

- la factorisation permet d'éviter overflows et timeouts :

| | SPEC95 | | |
|---|---|---|---|
| **timeout = 2 minutes** | **#ovfl** | **#timeouts** | **#op** |
| *C3C64-bit* | 0 | 0 | 24151 |
| *C3P64-bit* | 512 | 0 | 24151 |
| *PK64-bit* | 0 | 1270 | 24151 |

# Quatrième partie
# Conclusion

# Conclusions (résultats)

- filtrage d'ensembles de benchmark : une mauvaise idée ?
- réduction forte du nombre des exceptions
- meilleur algorithme pour la satisfiabilité : Janus (Jean-Claude Sogno, INRIA), étendu à 64 bits (Duong Nguyen)
- utilisation de deux benchmarks différents : non redondants
- meilleur critère prédictif de complexité : dimension de l'espace
- n'utiliser le passage au dual qu'en cas de nécessité
- validation de l'utilisation des entiers sur 64 bits

# Conclusions (problèmes rencontrés)

- sensitivité aux jeux de données (voir satisfiabilité et exceptions)
- objectif instable : implantation de la détection des exceptions en parallèle avec les expérimentations

# Conclusions (travaux à venir)

- décomposition d'espace (D. Merchat)
- intégration validée dans PIPS