Towards Efficient Optimization in Package Management Systems

Alexey Ignatiev[1], Mikoláš Janota[1], and Joao Marques-Silva[1,2]

[1] INESC-ID/IST, Lisbon, Portugal
[2] CASL/CSI, University College Dublin, Ireland

36th International Conference on Software Engineering

Hyderabad, India
June 5, 2014

Motivation

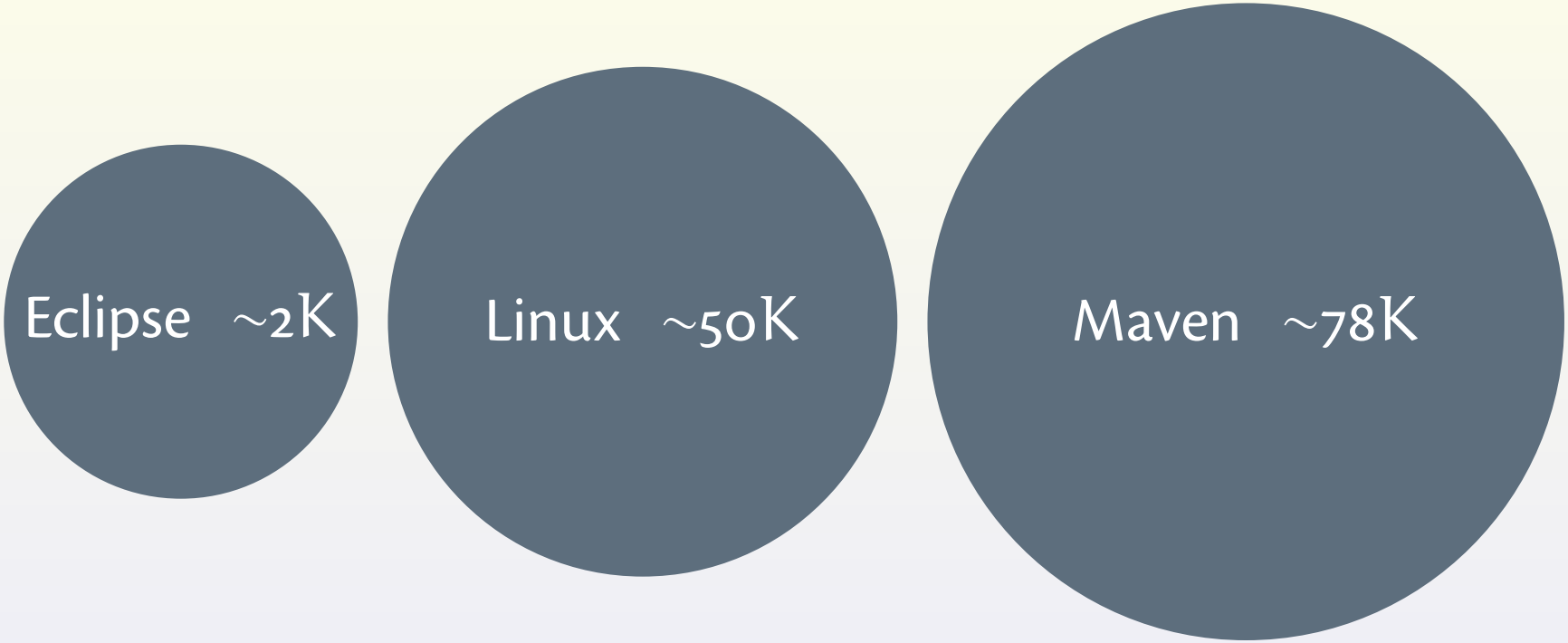


Figure : Number of packages in modern package management systems

Motivation

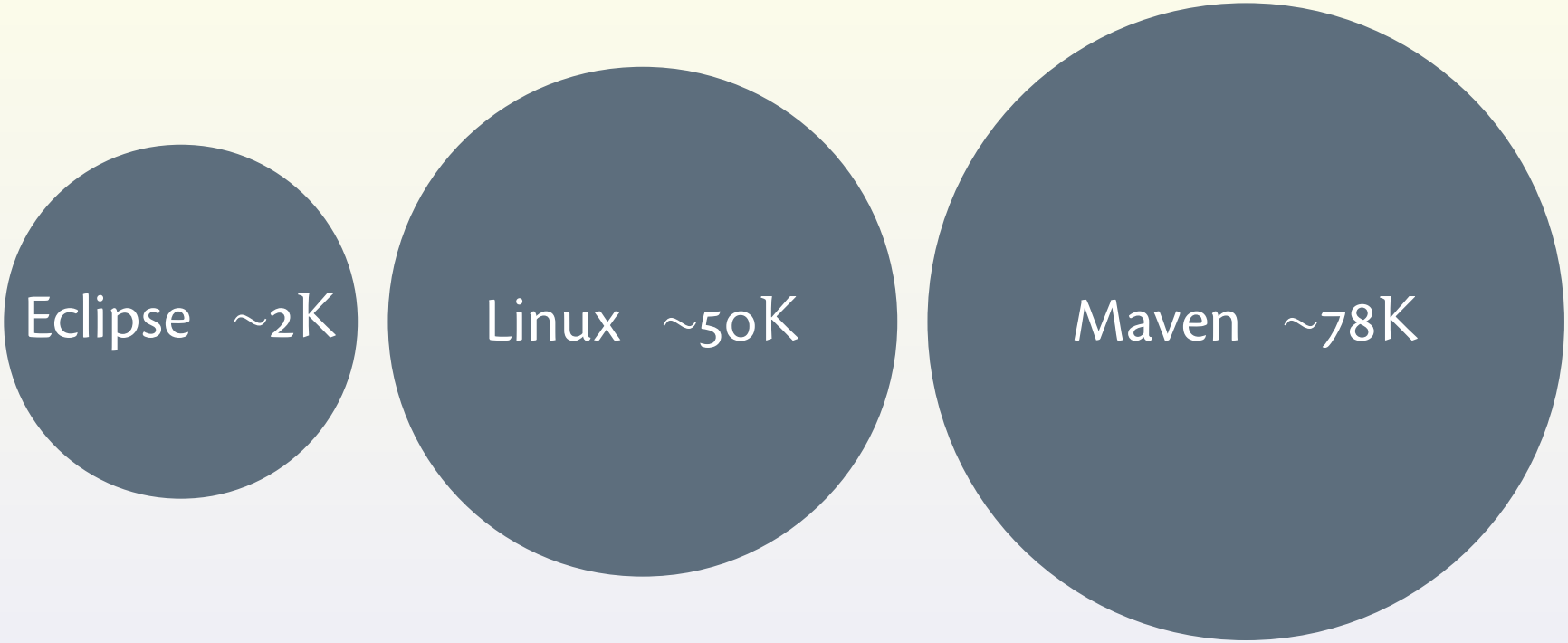


Figure : Number of packages in modern package management systems

Package installability problem

Checking whether a single package P can be installed, given a repository R, is NP-complete.

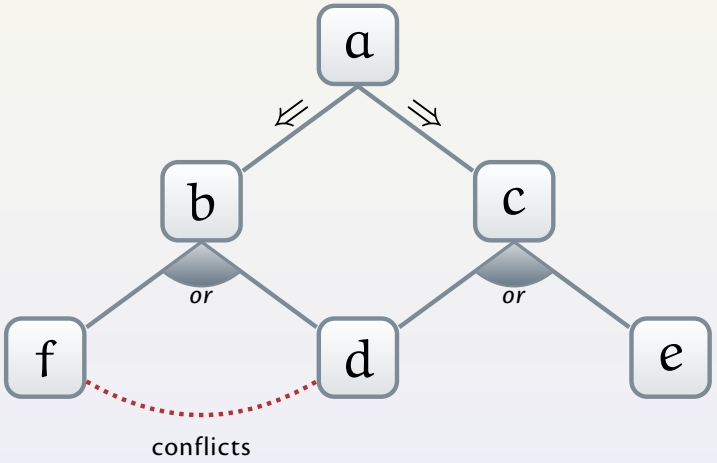
Motivation

Currently used package management systems (e.g. *APT*, *yum*, *MacPorts*)

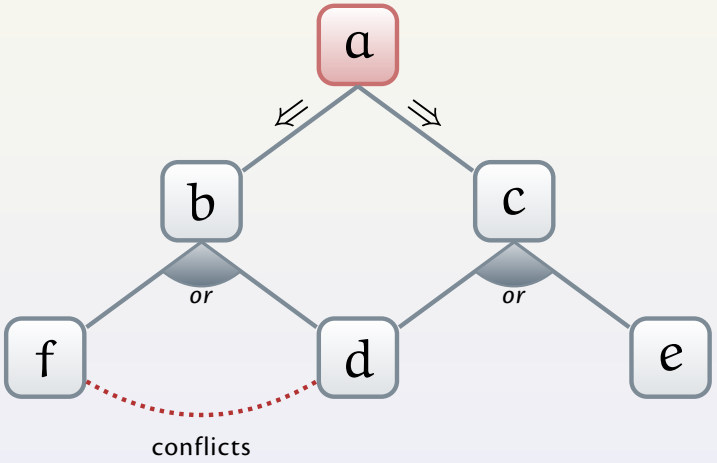
- are *incomplete*[1]
- don't support *"user preferences"*

[1] Chris Tucker, David Shuffelton, Ranjit Jhala, Sorin Lerner. OPIUM: Optimal Package Install/Uninstall Manager. ICSE 2007. pp. 178–188

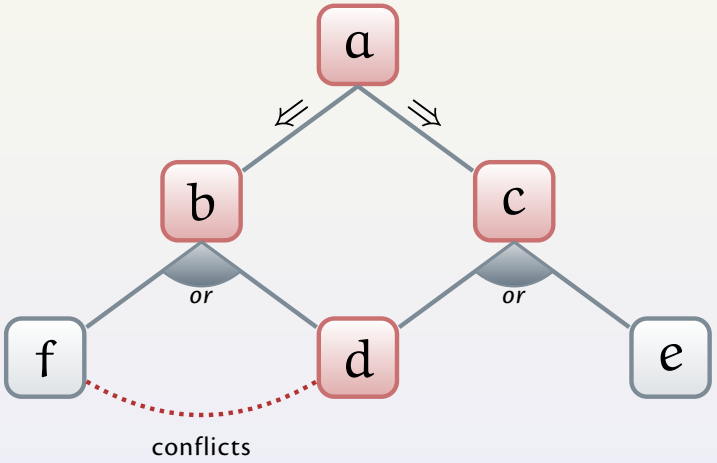
Example: dependencies and preferences over packages



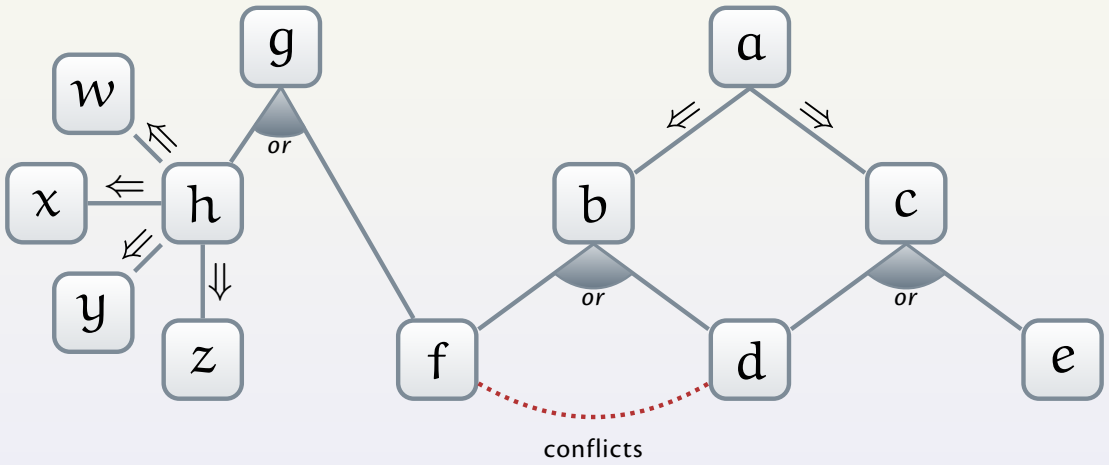
Example: dependencies and preferences over packages



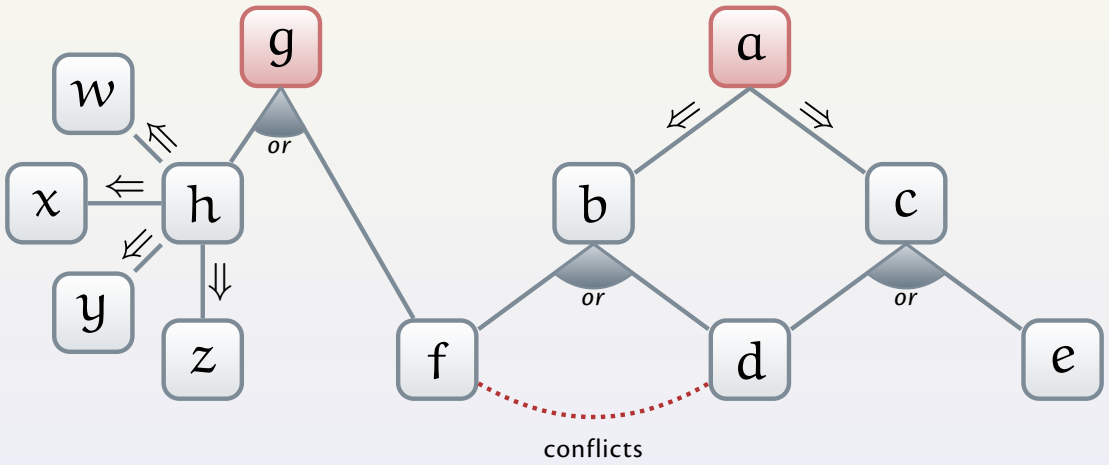
Example: dependencies and preferences over packages



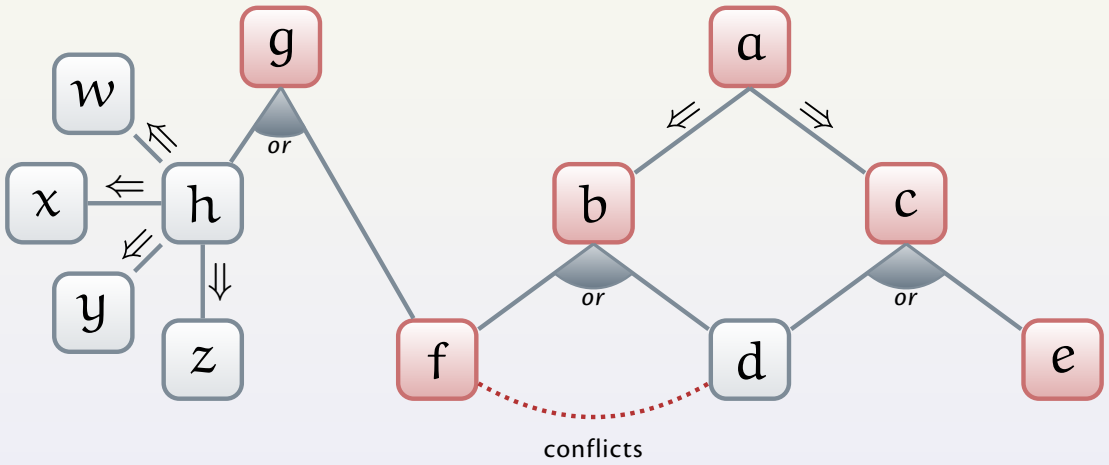
Example: dependencies and preferences over packages



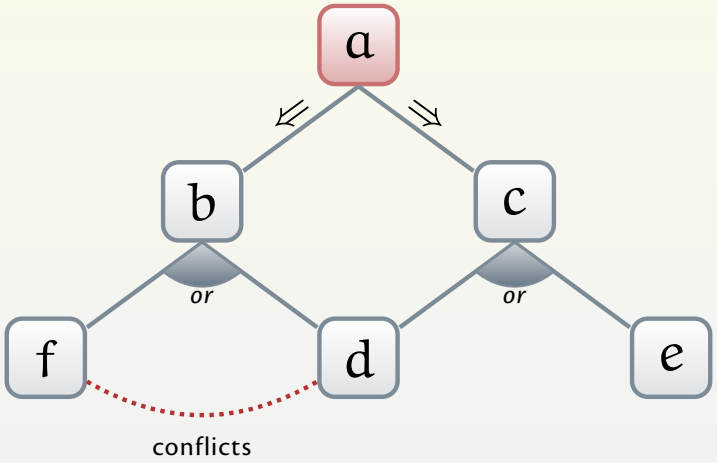
Example: dependencies and preferences over packages



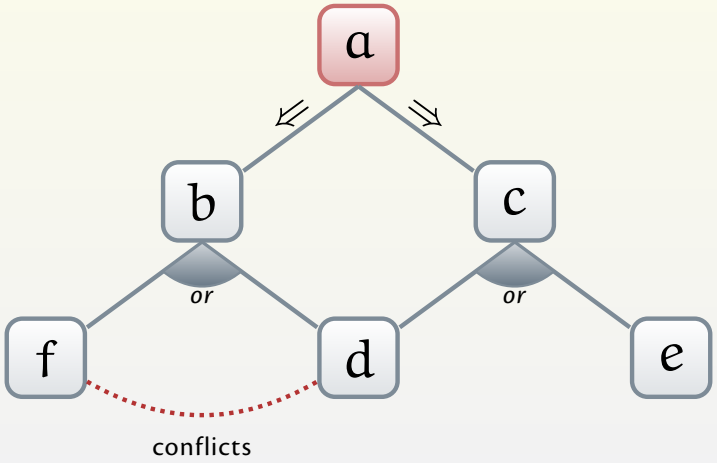
Example: dependencies and preferences over packages



Encoding to SAT and MaxSAT



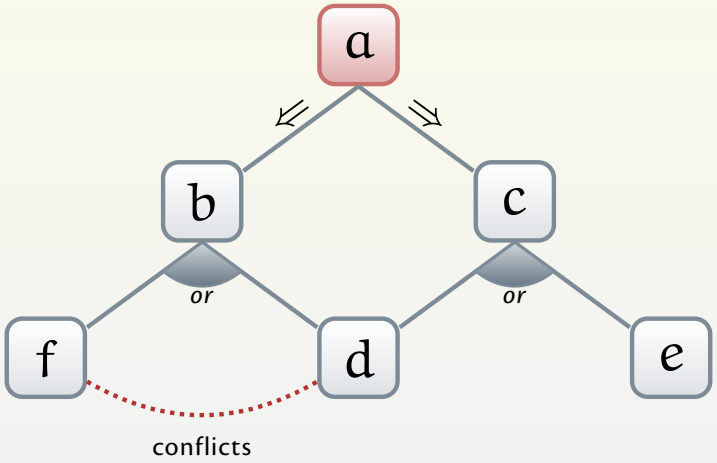
Encoding to SAT and MaxSAT



Find any solution — SAT

$$(\neg a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee f \vee d) \wedge (\neg c \vee d \vee e) \wedge (\neg f \vee \neg d) \wedge (a)$$

Encoding to SAT and MaxSAT



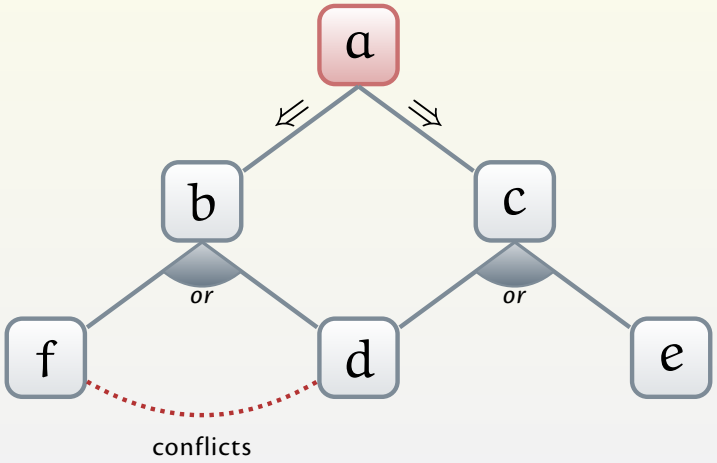
Find any solution — SAT

$$(\neg a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee f \vee d) \wedge (\neg c \vee d \vee e) \wedge (\neg f \vee \neg d) \wedge (a)$$

Find best solution — MaxSAT

$$(\neg a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee f \vee d) \wedge (\neg c \vee d \vee e) \wedge (\neg f \vee \neg d) \wedge (a)$$

Encoding to SAT and MaxSAT



Find any solution — SAT

$$(\neg a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee f \vee d) \wedge (\neg c \vee d \vee e) \wedge (\neg f \vee \neg d) \wedge (a)$$

Find best solution — MaxSAT

$$(\neg a \vee b) \wedge (\neg a \vee c) \wedge (\neg b \vee f \vee d) \wedge (\neg c \vee d \vee e) \wedge (\neg f \vee \neg d) \wedge (a)$$
$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

BLO

A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one[2].

[2] `www.mancoosi.org/misc-2012/` — 2–5 criteria in each category of MISC-2012 competition.

BLO

A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one[2].

$\Downarrow$

Boolean lexicographic optimization

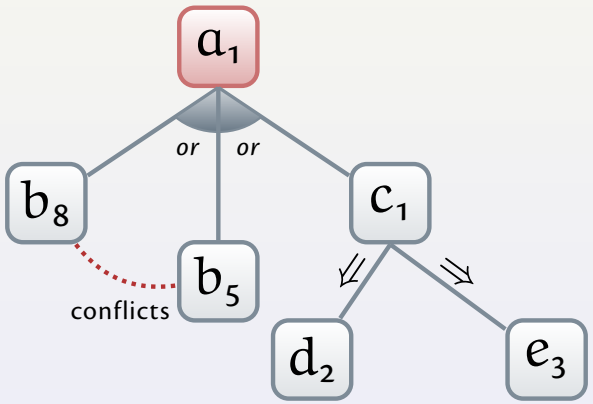
[2] `www.mancoosi.org/misc-2012/` — 2–5 criteria in each category of MISC-2012 competition.

BLO

A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one[2].

$$\Downarrow$$

Boolean lexicographic optimization



$$(\neg a_1 \vee b_8 \vee b_5 \vee c_1) \wedge (\neg b_8 \vee \neg b_5) \wedge (\neg c_1 \vee d_2) \wedge (\neg c_1 \vee e_3) \wedge (a_1)$$

$$(\neg a_1) \wedge (\neg b_8) \wedge (\neg b_5) \wedge (\neg c_1) \wedge (\neg d_2) \wedge (\neg e_3) \quad (1)$$

$$(\neg a_1, 1) \wedge (\neg b_8, 8) \wedge (\neg b_5, 5) \wedge (\neg c_1, 1) \wedge (\neg d_2, 2) \wedge (\neg e_3, 3) \quad (2)$$

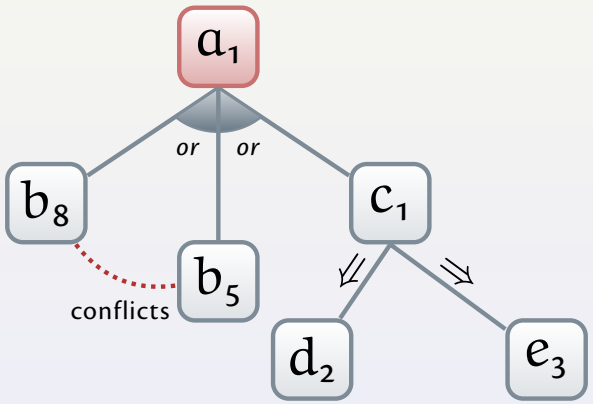
[2] `www.mancoosi.org/misc-2012/` — 2–5 criteria in each category of MISC-2012 competition.

BLO

A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one[2].

$$\Downarrow$$

Boolean lexicographic optimization



$$(\neg a_1 \vee b_8 \vee b_5 \vee c_1) \wedge (\neg b_8 \vee \neg b_5) \wedge (\neg c_1 \vee d_2) \wedge (\neg c_1 \vee e_3) \wedge (a_1)$$

$$\neg a_1 + \neg b_8 + \neg b_5 + \neg c_1 + \neg d_2 + \neg e_3 = 4 \quad (1)$$

$$(\neg a_1, 1) \wedge (\neg b_8, 8) \wedge (\neg b_5, 5) \wedge (\neg c_1, 1) \wedge (\neg d_2, 2) \wedge (\neg e_3, 3) \quad (2)$$

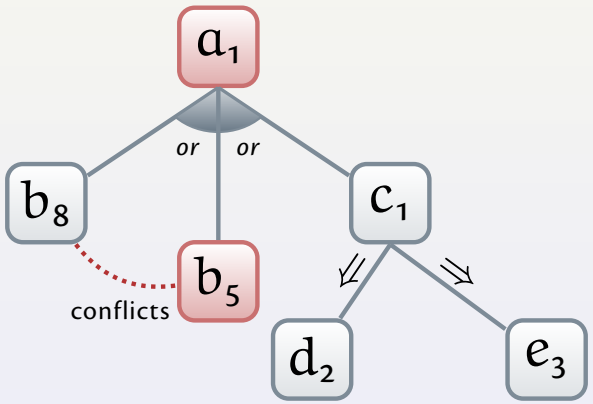
[2] `www.mancoosi.org/misc-2012/` — 2–5 criteria in each category of MISC-2012 competition.

BLO

A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one[2].

$\Downarrow$

Boolean lexicographic optimization



$$(\neg a_1 \vee b_8 \vee b_5 \vee c_1) \wedge (\neg b_8 \vee \neg b_5) \wedge (\neg c_1 \vee d_2) \wedge (\neg c_1 \vee e_3) \wedge (a_1)$$

$$\neg a_1 + \neg b_8 + \neg b_5 + \neg c_1 + \neg d_2 + \neg e_3 = 4 \quad (1)$$

$$(\neg a_1, 1) \wedge (\neg b_8, 8) \wedge (\neg b_5, 5) \wedge (\neg c_1, 1) \wedge (\neg d_2, 2) \wedge (\neg e_3, 3) \quad (2)$$

[2] `www.mancoosi.org/misc-2012/` — 2–5 criteria in each category of MISC-2012 competition.

Hybrid approach: MSS-based approximation

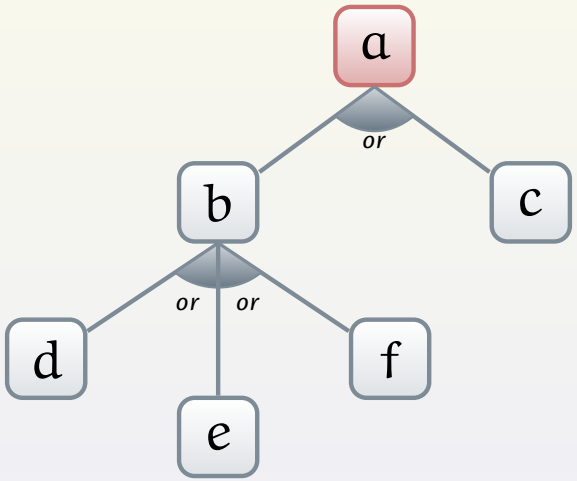
Timeout for some categories of MISC-2012 benchmarks is **300** seconds.

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

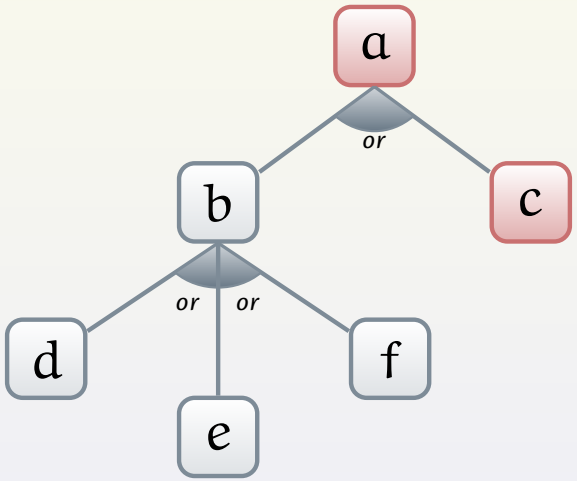


$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

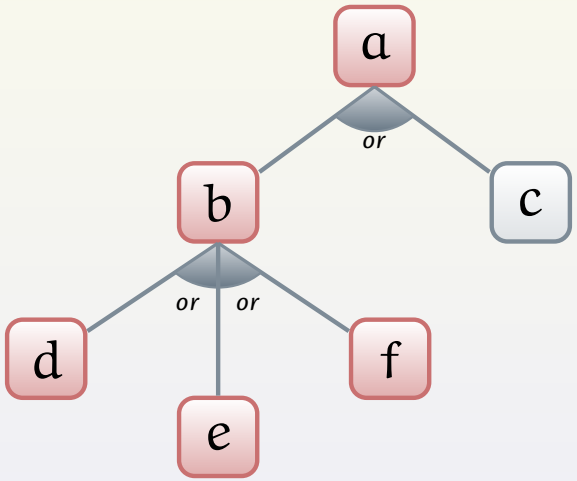


$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

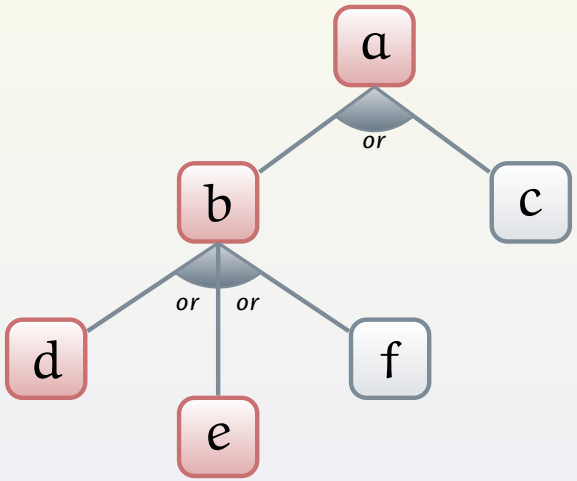


$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

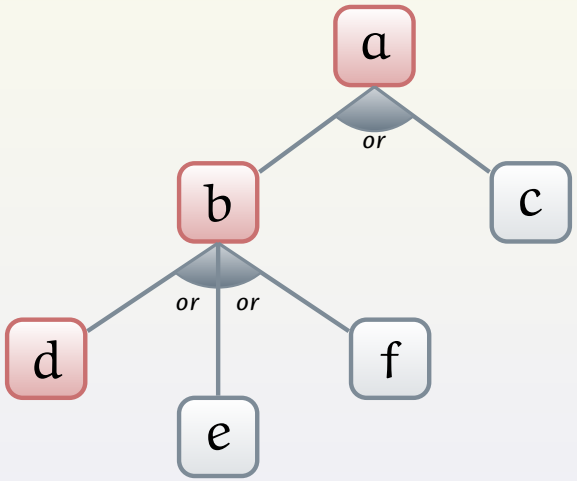


$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!

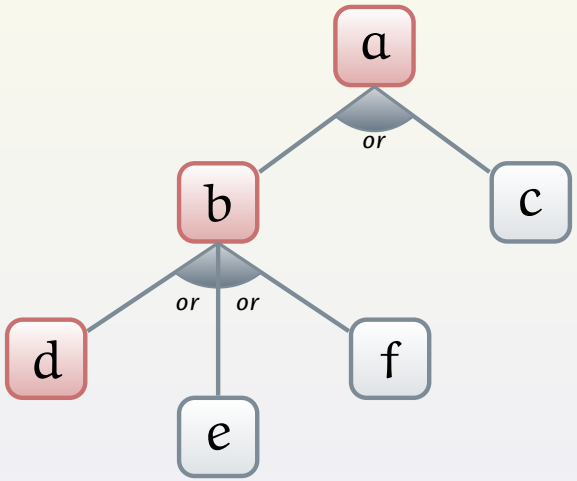


$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Hybrid approach: MSS-based approximation

Timeout for some categories of MISC-2012 benchmarks is **300** seconds.
Approximation is **much faster** than computing the exact MaxSAT solution!



$$(\neg a \vee b \vee c) \wedge (\neg b \vee d \vee e \vee f) \wedge (a)$$

$$(\neg a) \wedge (\neg b) \wedge (\neg c) \wedge (\neg d) \wedge (\neg e) \wedge (\neg f)$$

Each MSS can be seen as a **"local optimum"** of the optimization function, while the MaxSAT solution is the **"global optimum"**.

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*
2 **optimize** criterion f_i

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*
2 **optimize** criterion f_i
3 **if** Δ_E *is exceeded*: *there is no more time*
4 **break**

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*
2 **optimize** criterion f_i
3 **if** Δ_E *is exceeded*: *there is no more time*
4 **break**

5 **while** $i \leqslant n$**:** *approx. phase — BLO with MSSes*

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*
2 **optimize** criterion f_i
3 **if** Δ_E *is exceeded*: *there is no more time*
4 **break**

5 **while** $i \leqslant n$**:** *approx. phase — BLO with MSSes*
6 **approximate** criterion f_i

Hybrid approach: idea

input : n optimization criteria $f_1, f_2, \ldots, f_n$,
2 timeouts — Δ_E (exact phase) and Δ_A (approximation)

1 **foreach** $i \in \{1, \ldots, n\}$**:** *exact phase — BLO with MaxSAT*
2 **optimize** criterion f_i
3 **if** Δ_E *is exceeded*: *there is no more time*
4 **break**

5 **while** $i \leqslant n$**:** *approx. phase — BLO with MSSes*
6 **approximate** criterion f_i
7 **if** Δ_A *is exceeded*: *there is no more time*
8 **break**
9 $i \leftarrow i + 1$

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)
    3. P2 emulation mode — EclipseP2 solver (Sat4j library, 10 and 800 seconds)

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)
    3. P2 emulation mode — EclipseP2 solver (Sat4j library, 10 and 800 seconds)
- Machine configuration:

Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)
    3. P2 emulation mode — EclipseP2 solver (Sat4j library, 10 and 800 seconds)
- Machine configuration:
  - Intel Xeon 5160@3GHz with 4GB RAM

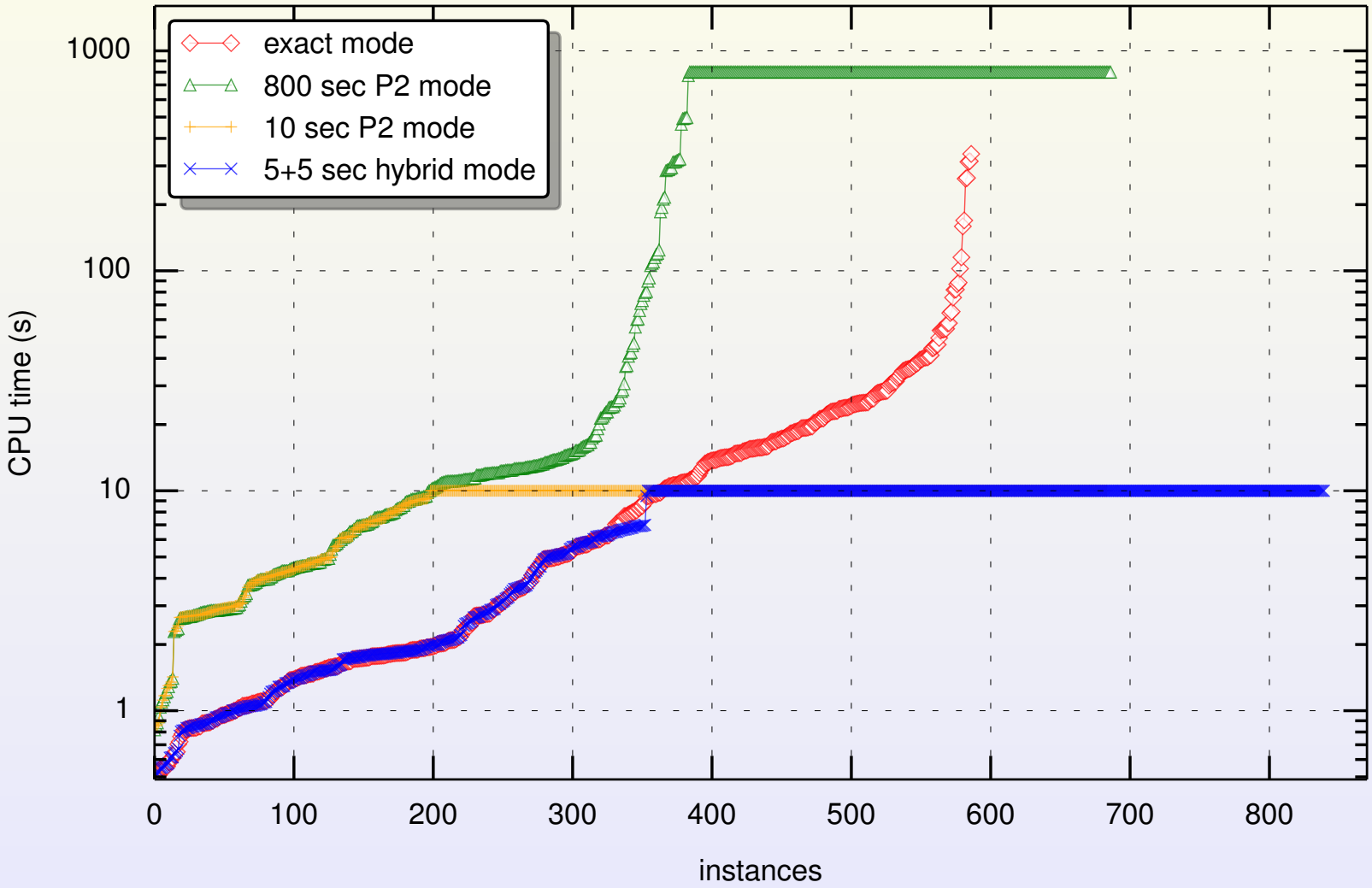
Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)
    3. P2 emulation mode — EclipseP2 solver (Sat4j library, 10 and 800 seconds)
- Machine configuration:
  - Intel Xeon 5160@3GHz with 4GB RAM
  - running Fedora Linux

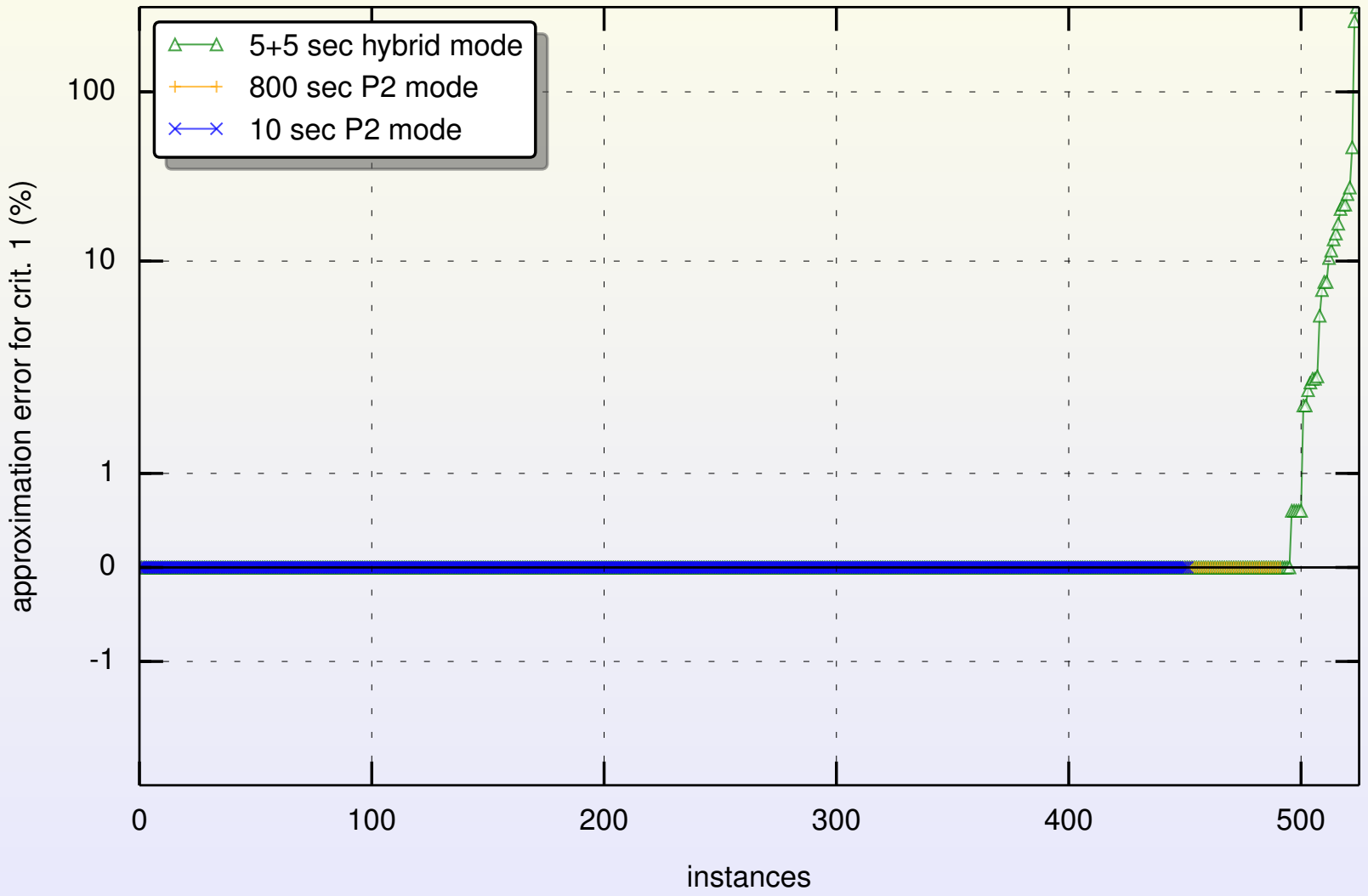
Experimental evaluation

- MANCOOSI International Solver Competition 2012 (MISC):
  - Package universe — from 27710 to 59094 packages (35276 in average)
- PackUpHyb tool
  - based on PackUp (participated in MISC-2012, open source)
  - uses MiniSAT 2.2
  - 3 modes of operation:
    1. exact mode — MaxSAT (800 seconds)
    2. hybrid mode — MaxSAT + MSS enumeration (5+5 seconds)
    3. P2 emulation mode — EclipseP2 solver (Sat4j library, 10 and 800 seconds)
- Machine configuration:
  - Intel Xeon 5160@3GHz with 4GB RAM
  - running Fedora Linux
  - 2GB memout

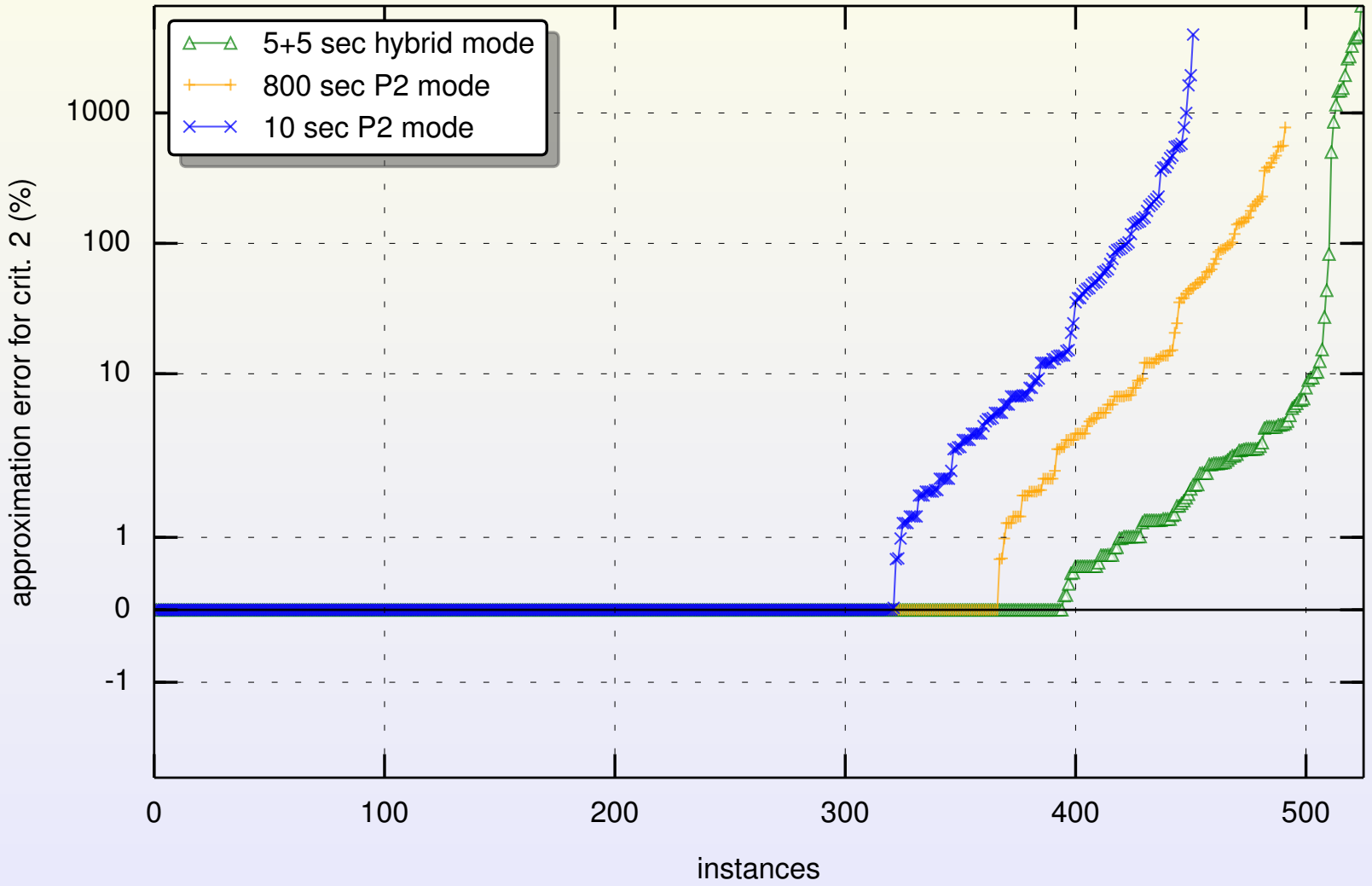
Performance of the approach



Approximation quality (level 1)



Approximation quality (level 2)



Summary and future work

- hybrid approach to *Package Upgradability*:

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT
- other optimization criteria (e.g. *leximin*)

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT
- other optimization criteria (e.g. *leximin*)
- comparison with: OPIUM, Aspcud, etc.

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT
- other optimization criteria (e.g. *leximin*)
- comparison with: OPIUM, Aspcud, etc.
- comparison with: APT, ZYpp, DNF, etc.

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT
- other optimization criteria (e.g. *leximin*)
- comparison with: OPIUM, Aspcud, etc.
- comparison with: APT, ZYpp, DNF, etc.
- integrate with a widely used tool (APT, ZYpp, DNF, etc.)

Summary and future work

- hybrid approach to *Package Upgradability*:
  - exact phase — MaxSAT approach
  - approximate phase — MSS enumeration
  - solution (exact or approximate) within 10 seconds
  - good approximation quality (guarantee of local optimality)

- improvement of MaxSAT
- can be applied to most exact solvers – not only MaxSAT
- other optimization criteria (e.g. *leximin*)
- comparison with: OPIUM, Aspcud, etc.
- comparison with: APT, ZYpp, DNF, etc.
- integrate with a widely used tool (APT, ZYpp, DNF, etc.)
- deploy in Linux distributions

Thank you for your attention!