Towards Efficient Optimization in Package Management Systems

Alexey Ignatiev¹, Mikoláš Janota¹, and Joao Marques-Silva¹,²

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

- **Eclipse**: ~2K
- **Linux**: ~50K
- **Maven**: ~78K

**Figure**: Number of packages in modern package management systems
Motivation

Eclipse  ~2K  Linux  ~50K  Maven  ~78K

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”

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

\[
(-a \lor b) \land (-a \lor c) \land (-b \lor f \lor d) \land (-c \lor d \lor e) \land (-f \lor -d) \land (a)
\]
Encoding to SAT and MaxSAT

Find any solution — SAT
\((\neg a \lor b) \land (\neg a \lor c) \land (\neg b \lor f \lor d) \land (\neg c \lor d \lor e) \land (\neg f \lor \neg d) \land (a)\)

Find best solution — MaxSAT
\((\neg a \lor b) \land (\neg a \lor c) \land (\neg b \lor f \lor d) \land (\neg c \lor d \lor e) \land (\neg f \lor \neg d) \land (a)\)
Encoding to SAT and MaxSAT

Find any solution — SAT

\((\neg a \lor b) \land (\neg a \lor c) \land (\neg b \lor f \lor d) \land (\neg c \lor d \lor e) \land (\neg f \lor \neg d) \land (a)\)

Find best solution — MaxSAT

\((\neg a \lor b) \land (\neg a \lor c) \land (\neg b \lor f \lor d) \land (\neg c \lor d \lor e) \land (\neg f \lor \neg d) \land (a)\)

\((\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)\)
A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one$^2$.

---

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

\[\downarrow\]

**Boolean lexicographic optimization**

---

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

\[ \downarrow \]

*Boolean lexicographic optimization*

\[ (\neg a_1 \lor b_8 \lor b_5 \lor c_1) \land (\neg b_8 \lor \neg b_5) \land (\neg c_1 \lor d_2) \land (\neg c_1 \lor e_3) \land (a_1) \]

\[ (\neg a_1) \land (\neg b_8) \land (\neg b_5) \land (\neg c_1) \land (\neg d_2) \land (\neg e_3) \] (1)

\[ (\neg a_1, 1) \land (\neg b_8, 8) \land (\neg b_5, 5) \land (\neg c_1, 1) \land (\neg d_2, 2) \land (\neg e_3, 3) \] (2)

---

$^2$www.mancoosi.org/misc-2012/ — 2–5 criteria in each category of MISC-2012 competition.
A user can have *multiple* optimization criteria $f_1, f_2, \ldots, f_n$ — not just one$^2$.

\[ \downarrow \]

**Boolean lexicographic optimization**

\[
(-a_1 \lor b_8 \lor b_5 \lor c_1) \land (-b_8 \lor -b_5) \land (-c_1 \lor d_2) \land (-c_1 \lor e_3) \land (a_1)
\]

\[
-a_1 + -b_8 + -b_5 + -c_1 + -d_2 + -e_3 = 4
\]  \hspace{1cm} (1)

\[
(-a_1, 1) \land (-b_8, 8) \land (-b_5, 5) \land (-c_1, 1) \land (-d_2, 2) \land (-e_3, 3)
\]  \hspace{1cm} (2)

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

$$\Rightarrow$$

**Boolean lexicographic optimization**

\[
(-a_1 \lor b_8 \lor b_5 \lor c_1) \land (-b_8 \lor -b_5) \land (-c_1 \lor d_2) \land (-c_1 \lor e_3) \land (a_1)
\]

\[
-a_1 + -b_8 + -b_5 + -c_1 + -d_2 + -e_3 = 4
\]  

\[
(-a_1, 1) \land (-b_8, 8) \land (-b_5, 5) \land (-c_1, 1) \land (-d_2, 2) \land (-e_3, 3)
\]

$^2$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!

\[(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a)\]

\[(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)\]
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!

\[
(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a) \\
(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)
\]
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!

\[(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a)\]

\[(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)\]
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!

\[
(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a) \\
(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)
\]
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!

\[
(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a) \\
(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)
\]
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!

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

\[
(\neg a \lor b \lor c) \land (\neg b \lor d \lor e \lor f) \land (a)
\]
\[
(\neg a) \land (\neg b) \land (\neg c) \land (\neg d) \land (\neg e) \land (\neg f)
\]
Hybrid approach: idea

\text{input} : n \text{ 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 — \( \Delta_E \) (exact phase) and \( \Delta_A \) (approximation)
Hybrid approach: idea

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

1. **foreach** \( i \in \{1, \ldots, n\} \):
2. **optimize** criterion \( f_i \)
Hybrid approach: idea

**input**: \( n \) optimization criteria \( f_1, f_2, \ldots, f_n \),
2 timeouts — \( \Delta_E \) (exact phase) and \( \Delta_A \) (approximation)

1. **foreach** \( i \in \{1, \ldots, n\} \):
2. **optimize** criterion \( f_i \)
3. **if** \( \Delta_E \) is exceeded:
4. **break**
Hybrid approach: idea

**input**: \( n \) optimization criteria \( f_1, f_2, \ldots, f_n \),

2 timeouts — \( \Delta_E \) (exact phase) and \( \Delta_A \) (approximation)

```
1. foreach \( i \in \{1, \ldots, n\} \):
2.   optimize criterion \( f_i \)
3.   if \( \Delta_E \) is exceeded:
4.     break

5. while \( i \leq n \):
```

- **exact phase** — BLO with MaxSAT
- **approx. phase** — BLO with MSSes
Hybrid approach: idea

**input**: n optimization criteria f₁, f₂, …, fₙ,
2 timeouts — Δₜ (exact phase) and Δₘ (approximation)

1. **foreach** i ∈ {1, . . . , n}:
2.   **optimize** criterion fᵢ
3.   **if** Δₜ is exceeded:
4.     **break**

5. **while** i ≤ n:
6.   **approximate** criterion fᵢ

**exact phase** — BLO with MaxSAT
**there is no more time**

**approx. phase** — BLO with MSSes
Hybrid approach: idea

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

2 timeouts — $\Delta_E$ (exact phase) and $\Delta_A$ (approximation)

1. **foreach** $i \in \{1, \ldots, n\}$:
   2. **optimize** criterion $f_i$
   3. **if** $\Delta_E$ is exceeded:
      4. **break**

4. **while** $i \leq n$:
   5. **approximate** criterion $f_i$
   6. **if** $\Delta_A$ is exceeded:
      7. **break**
   8. $i \leftarrow i + 1$
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:
    - 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

![Graph showing CPU time vs. instances for different modes of operation. The graph includes exact mode, 800 sec P2 mode, 10 sec P2 mode, and 5+5 sec hybrid mode. The x-axis represents instances, and the y-axis represents CPU time in seconds. The graph illustrates the efficiency of the approach in package management systems.]
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!