Automatisches Theorembeweisen in großen Theorien

Thomas Raths

Institut für Informatik
Universität Potsdam

12. Februar 2009

Themen

- Automatisches Beweisen in großen Theorien
- QMLTP-Library:
  umfassende, standardisierte Problembibliothek und Plattform für automatisches Beweisen in modaler Prädikatenlogik
- Inkonsistente Wissensbasen:
  Belief Revision und nichtmonotones Schließen

QMLTP-Library

- Quantified Modal Logic (QML) Theorem Proving Library (T. Raths & J. Otten, 2009)
- umfassende, standardisierte Problembibliothek und Plattform für automatisches Beweisen in modaler Prädikatenlogik (QML)

QMLTP-Library

- Quantified Modal Logic (QML) Theorem Proving Library (T. Raths & J. Otten, 2009)
- umfassende, standardisierte Problembibliothek und Plattform für automatisches Beweisen in modaler Prädikatenlogik (QML)
- soll stimulieren Entwicklung neuer, effizientere Beweiser und Kalküle für QML

QMLTP-Library

- Quantified Modal Logic (QML) Theorem Proving Library (T. Raths & J. Otten, 2009)
- umfassende, standardisierte Problembibliothek und Plattform für automatisches Beweisen in modaler Prädikatenlogik (QML)
- soll stimulieren Entwicklung neuer, effizientere Beweiser und Kalküle für QML
- QML: klassische Prädikatenlogik erweitert um Modalitäten

QMLTP-Library

- Quantified Modal Logic (QML) Theorem Proving Library (T. Raths & J. Otten, 2009)
- umfassende, standardisierte Problembibliothek und Plattform für automatisches Beweisen in modaler Prädikatenlogik (QML)
- soll stimulieren Entwicklung neuer, effizientere Beweiser und Kalküle für QML
- QML: klassische Prädikatenlogik erweitert um Modalitäten
- Anwendungen: Plansystem PKS, Dialogsystem Artimis, Satzplaner SPUD, Tools zur Programmverifikation und -synthese (KeY, VSE II, KIV), Wissensbasen für Beschreibungslogik
- QML erlaubt kompakte, natürliche Repräsentation: konditionale Pläne, abstrakte Informationen, unvollständiges Wissen, Programme, Objekte, ...

QMLTP-Library (2)

- Call for Problems & Provers, erste Demo-Version, Position-Paper für Tableaux-2009

QMLTP-Library (2)

- Call for Problems & Provers, erste Demo-Version, Position-Paper für Tableaux-2009
- QMLTP v0.1: `http://www.iltp.de/qmltp`
- 100 Probleme:
  98 aus Gödel-Transformation auf Problemen der TPTP-Lib.
  2: Instanzen der Barcan bzw. umgekehrten Barcan Formel:
  $(\ \forall x.(\Box f(x))) \rightarrow (\Box(\forall x.f(x))),\quad \Box(\forall x.f(x)) \rightarrow (\forall x.(\Box f(x)))\ )$

QMLTP-Library (2)

- Call for Problems & Provers, erste Demo-Version, Position-Paper für Tableaux-2009
- QMLTP v0.1: `http://www.iltp.de/qmltp`
- 100 Probleme:
  98 aus Gödel-Transformation auf Problemen der TPTP-Lib.
  2: Instanzen der Barcan bzw. umgekehrten Barcan Formel:
  $(\ \forall x.(\Box f(x))) \rightarrow (\Box(\forall x.f(x))),\quad \Box(\forall x.f(x)) \rightarrow (\forall x.(\Box f(x)))\ )$

- Syntax: TPTP-Syntax, erweitert um
  box(i):F für $\Box_i F$, dia(i):F für $\Diamond_i F$

QMLTP-Library (2)

- Call for Problems & Provers, erste Demo-Version, Position-Paper für Tableaux-2009
- QMLTP v0.1: `http://www.iltp.de/qmltp`
- 100 Probleme:
  98 aus Gödel-Transformation auf Problemen der TPTP-Lib.
  2: Instanzen der Barcan bzw. umgekehrten Barcan Formel:
  $(\ \forall x.(\Box f(x))) \rightarrow (\Box(\forall x.f(x))), \quad \Box(\forall x.f(x)) \rightarrow (\forall x.(\Box f(x))) \ )$

- Syntax: TPTP-Syntax, erweitert um
  box(i):F für $\Box_i F$, dia(i):F für $\Diamond_i F$
- Status, Rating für S4, kumulative Domäne ermittelt
  mit zwei Beweisern: GQML-Prover v1.2 (Thion et al., 2002),
  mleanTAP v1.0 (Otten, 2009)

QMLTP-Library (2)

- Call for Problems & Provers, erste Demo-Version,
  Position-Paper für Tableaux-2009
- QMLTP v0.1: `http://www.iltp.de/qmltp`
- 100 Probleme:
  98 aus Gödel-Transformation auf Problemen der TPTP-Lib.
  2: Instanzen der Barcan bzw. umgekehrten Barcan Formel:
  $(\ \forall x.(\Box f(x))) \rightarrow (\Box(\forall x.f(x))),\quad \Box(\forall x.f(x)) \rightarrow (\forall x.(\Box f(x)))\ )$

- Syntax: TPTP-Syntax, erweitert um
  box(i):F für $\Box_i F$, dia(i):F für $\Diamond_i F$
- Status, Rating für S4, kumulative Domäne ermittelt
  mit zwei Beweisern: GQML-Prover v1.2 (Thion et al., 2002),
  mleanTAP v1.0 (Otten, 2009)
- Ausblick, to do:
  Probleme: aus Anwendungen, Papers; Beweiser
  Logiken: K, K4, D, D4, T, S4, S5; Domänen: konstant, varrier

Inkonsistente Wissensbasen

- Hintergrund: Sicherheitsanforderungen als Regeln in Expertensystem ESSLN (Le, 1989) verwalten (C. Brandt)
- mögliche Anwendung von leanCoP: Inkonsistenzen erkennen
- implementiert: Charakterisierung von Inkonsistenz-Management nach Benferhat et al. (1993):
- maximal konsistente Teilmengen einer Wissensbasis
- maximal bzgl. Mengen-Inklusion und Kardinalität
- $\Rightarrow$ minimale Kandidaten: minimale Menge von Fakten, ohne die die Wissensbasis konsistent ist
- berücksichtigt Prioritäten der Fakten
- benutzt leanCoP v2.1 (J. Otten) als Erfüllbarkeits-Checker
- Zahl der Erfüllbarkeitschecks minimal
- wie groß dürfen die Wissensbasen sein? (Leistung)

Inkonsistente Wissensbasen (2)

Beispiel:

$a : 1.0,\ \ \neg a : 1.0,\ \ a \longrightarrow b : 0.7,\ \ \neg a \longrightarrow c.0.6,\ \ \neg b : 0.4,\ \ \neg c : 0.2$

minimale Kandidaten: $[a : 1.0, \neg c : 0.2],\ [\neg a : 1.0, \neg b : 0.4]$

Inkonsistente Wissensbasen (2)

Beispiel:

$a : 1.0,\ \ \neg a : 1.0,\ \ a \longrightarrow b : 0.7,\ \ \neg a \longrightarrow c.0.6,\ \ \neg b : 0.4,\ \ \neg c : 0.2$

minimale Kandidaten: $[a : 1.0, \neg c : 0.2],\ [\neg a : 1.0, \neg b : 0.4]$

Leistung:

Wissensbasis bestehe aus prädikatenlogische Literale

($\Rightarrow$ Erfüllbarkeitstests: trivial)

	maximale Größe der Wissensbasis für minimale Kandidaten in 1 min basierend auf	
	Mengen-Inklusion	Kardinalität
ohne Priorität	11	120
mit Priorität (5 Stufen)	45	250

Automatisches Theorembeweisen in großen Theorien

- automatisches Beweisen eines Theorems in einer umfangreichen Hintergrund-Theorie ("large theory")
- Theorie: viele Axiome (bis Millionen), nur wenige relevant (1-5, max. 50)

Automatisches Theorembeweisen in großen Theorien

- automatisches Beweisen eines Theorems in einer umfangreichen Hintergrund-Theorie ("large theory")
- Theorie: viele Axiome (bis Millionen),
  nur wenige relevant (1-5, max. 50)
- Anwendug in Ontologien, Wissensbasen:
  - intelligente Anfragen bearbeiten, die das Erkennen logischer Zusammenhänge erfordern
    Ausdruckskraft möglicher Anfragen erhöhen
  - Inkonsistenzen erkennen und lokalisieren
  - Beispiele: SUMO, Cyc

Automatisches Theorembeweisen in großen Theorien

- automatisches Beweisen eines Theorems in einer umfangreichen Hintergrund-Theorie ("large theory")
- Theorie: viele Axiome (bis Millionen),
  nur wenige relevant (1-5, max. 50)
- Anwendug in Ontologien, Wissensbasen:
  - intelligente Anfragen bearbeiten, die das Erkennen logischer Zusammenhänge erfordern
    Ausdruckskraft möglicher Anfragen erhöhen
  - Inkonsistenzen erkennen und lokalisieren
  - Beispiele: SUMO, Cyc
- Anwendung in mathematische Bibliotheken:
- Beweise überprüfen
- neue Theoreme beweisen
- Inkonsistenzen erkennen und lokalisieren
- Beispiele: Mizar Mathematical Library (MML)

Entwicklung, Menschen, Beweiser

- Goeff Sutcliffe (1998-): TPTP Library:
- Josef Urban (2003): Mizar ML $\rightarrow$ TPTP ($\Rightarrow$ MPTP)
- Ramachandran, Reagan, Goolsbey (2005): OpenCyc $\rightarrow$ TPTP
- Adam Pease, G. Sutcliffe (2007): SUMO $\rightarrow$ TPTP
- CADE-21, 2007:
  - Workshop "Empirically Successful Automated Reasoning in Large Theories" (Sutcliffe, Urban, Schulz)
  - MPTP 100$ Challenge (J. Urban, G. Sutcliffe) leanCoP 2.0 (J. Otten), schlägt alle stand-alone Beweiser auf chainy-Version
  - Meta-Systeme: MaLARea (J. Urban, 2007), SRASS (G. Sutcliffe, 2007)

Entwicklung, Menschen, Beweiser (2)

IJCAR 2008, CASC-J4 LTB:

- 3000$ von Articulate Software für erfolgreichstes Beweiser-System in Kategorie SUMO: Metasystem SInE (K. Hoder, 2008)
- SUMO 100$ Challenges (Articulate Software)
- randoCoP v1.1 (T. Raths, J. Otten): 3. Platz der Beweiser mit Beweisausgabe in Kategorie LTB

Cyc

- umfangreiche Ontologie von Alltagswissen ("common sense")
- gestartet von Douglas Lenat 1984
- OpenCyc: open-source Version, 1.8 Mill. Fakten, 260000 Begriffe (2006)
- Anwendung: Sprachverstehen, spezielle Ontologie aufbauen
- OpenCyc $\longrightarrow$ TPTP-syntax (Prädikatenlogik): Ramachandran, Reagan, Goolsbey (2005), 90%
- automatisches Beweisen für Bearbeitung von Anfragen
- eingebaute Inferenzmaschine: higher-order Metaschließen, optimiert auf Cyc, modular, graphen-orientiert
- effizienter als herkömmliche Versionen von Vampire, E, SPASS

SUMO

- Suggested Upper Merged Ontology
- I. Niles, A. Pease (2000-)
- open-source
- mit multi-linguales Lexikon WordNet 1.6 verbunden
- 4000 Axiome, 1000 Begriffe, 750 Regeln
- kombiniert mit allen assoziierten Ontologien: 70000 Axiome, 20000 Begriffe
- Prdikatenlogik mit Sorten
- SUMO $\rightarrow$ TPTP (A. Pease, G. Sutcliffe, 2007)
- SigmaKEE: Wissensreprsentations-Umgebung, die SUMO untersttzt
- automatisches Beweisen: bis jetzt mit Vampire
  soll erweitert werden um Paket von automatischen Beweisern ("TPTPWorld") (S. Trac, G. Sutcliffe, A. Pease, 2008)

Mizar Mathematical Library (MML)

- größte mathematische Bibliothek von Theoremen und Definitionen
- 2694 Theoreme aus 943 Artikel formalisiert (2006)
- basiert auf Tarski-Grothendieck Mengentheorie, nah an ZFC
- MML $\longrightarrow$ TPTP: Josef Urban (2004)
  Typeninformation entfernt
- MPTP 100$ Challenges (Urban, Sutcliffe)

Probleme mit großen Theorien in TPTP-Bibliothek

- aus Cyc, SUMO, Mizar-ML
- jeweils ein Theorem, basiedend auf unterschiedlich umfangreiche Hintergrund-Theorien
- gewünscht: Theorem in einer möglichst großen Theorie beweisen
  $\Rightarrow$ Benutzer muss nicht die relevanten Axiome selbst auswählen
- 1385 Probleme-Versionen

Quelle	Probleme	Versionen	Axiome
Cyc	30	6	$60 \cdots$ 3.3 Mill.
SUMO	35	3	$10000 \cdots 60000$
MPTP	245	4	$50 \cdots 70000$
			(mit $\doteq$: 40% mehr)

ATP-Systeme

Konnektionen-Kalkül:

leanCoP 2.0,	J. Otten	beschränktes Backtracking, Regularität, Fakt.
ileanCoP 1.2	J. Otten	intuitionistische Prädikatenlogik
randoCoP 1.1	J. Otten & T. Raths	randomisierte Beweissuche, Beweisausgabe, automatisch $\doteq$-Axiome zufügen, TPTP
Setheo 3.3	R. Letz	Modell-Elimination

ATP-Systeme

Konnektionen-Kalkül:

leanCoP 2.0,	J. Otten	beschränktes Backtracking, Regularität, Fakt.
ileanCoP 1.2	J. Otten	intuitionistische Prädikatenlogik
randoCoP 1.1	J. Otten & T. Raths	randomisierte Beweissuche, Beweisausgabe, automatisch $\doteq$-Axiome zufügen, TPTP
Setheo 3.3	R. Letz	Modell-Elimination

Resolution und Paramodulation:

Vampire 10.0	A. Voronkov	Indextechniken, Compiling, Termrewriting
E 1.0	S. Schulz	reiner Gleichheits-Beweiser, Tuning
SPASS 3.0	C. Weidenbach	Superposition, Zerlegungsregel
Metis 2.1	J. Hurd	in SML, benutzt von HOL4
Otter 3.3, Prover9	W. McCune	Otter: stabile Benchmark
SNARK 08/07	M. Stickel	in Lisp, angewandt auf Wissensbasis

ATP-Systeme

Konnektionen-Kalkül:

leanCoP 2.0,	J. Otten	beschränktes Backtracking, Regularität, Fakt.
ileanCoP 1.2	J. Otten	intuitionistische Prädikatenlogik
randoCoP 1.1	J. Otten & T. Raths	randomisierte Beweissuche, Beweisausgabe, automatisch $\doteq$-Axiome zufügen, TPTP
Setheo 3.3	R. Letz	Modell-Elimination

Resolution und Paramodulation:

Vampire 10.0	A. Voronkov	Indextechniken, Compiling, Termrewriting
E 1.0	S. Schulz	reiner Gleichheits-Beweiser, Tuning
SPASS 3.0	C. Weidenbach	Superposition, Zerlegungsregel
Metis 2.1	J. Hurd	in SML, benutzt von HOL4
Otter 3.3, Prover9	W. McCune	Otter: stabile Benchmark
SNARK 08/07	M. Stickel	in Lisp, angewandt auf Wissensbasis

Instanzenbasierte Methoden:

iProver 0.5c	K. Korovin	nutzt MiniSat, E-/V-Klausifier bei FOF
Equinox 3.0	K. Claessen	nutzt MiniSat

ATP-Systeme

Konnektionen-Kalkül:

leanCoP 2.0,	J. Otten	beschränktes Backtracking, Regularität, Fakt.
ileanCoP 1.2	J. Otten	intuitionistische Prädikatenlogik
randoCoP 1.1	J. Otten & T. Raths	randomisierte Beweissuche, Beweisausgabe, automatisch $\doteq$-Axiome zufügen, TPTP
Setheo 3.3	R. Letz	Modell-Elimination

Resolution und Paramodulation:

Vampire 10.0	A. Voronkov	Indextechniken, Compiling, Termrewriting
E 1.0	S. Schulz	reiner Gleichheits-Beweiser, Tuning
SPASS 3.0	C. Weidenbach	Superposition, Zerlegungsregel
Metis 2.1	J. Hurd	in SML, benutzt von HOL4
Otter 3.3, Prover9	W. McCune	Otter: stabile Benchmark
SNARK 08/07	M. Stickel	in Lisp, angewandt auf Wissensbasis

Instanzenbasierte Methoden:

iProver 0.5c	K. Korovin	nutzt MiniSat, E-/V-Klausifier bei FOF
Equinox 3.0	K. Claessen	nutzt MiniSat

Natürliches Schliessen: Muscadet 3.0 (D. Pastre) Prolog

Anpassung der Beweiser

randoCoP 1.1:

- Beweisausgabe: gut leserlich, verständlicher Beweis, ohne zusätzliche Suche
- fügt Gleichheitsaxiome automatisch zu
  wesentlich schneller als TPTP-Tool
- liest auch TPTP-Syntax

Vampire 10.0:

- Vorverarbeitung, grobe Auswahl relevanter Axiome, mit Prolog

Zwei verschiedene Artenvon Problemen

	große Theorie	kleine Theorie
Axiome	nur wenige Axiome relevant	alle Axiome relevant
Beweis	einfach, wenn man die relevanten Axiome erstmal hat,	Beweis schwer,
	Axiome sind meist Fakten oder Regeln	verschachtelte Axiome
Heraus-forderung	relevante Axiome finden	Beweis finden

Metasysteme SInE

- Krystof Hoder
- wählt Axiome iterativ aus
- Relation von den in den Axiomen und der Behauptung vorkommenden Symbolen zu den Axiomen, die diese enthalten
- iterativ
- anfangs: nur Behauptung wird genommen
- Beweisversuch
- in nchsten Iterationschritt alle Axiome dazu genommen, die Symbole enthalten, die in bisher gewählten Axiomen vorkommen, usw.
- Vorverarbeitung: Probleme, die die selbe Axiommenge enthalten, werden zusammen gruppiert
- kann beliebigen Beweiser benutzen, z.B. E, Vampire, randoCoP, leanCoP

Meta-System MaLARea

- Josef Urban
- Lernen der Relevanz der Axiome für die Behauptung aus vorhergehenden Beweisen
- Ähnlichkeit: symbol-, term-, modellbasiert
- relevantesten Axiome werden fr die nächsten Beweisversuche benutzt
- benutzt zum Beweisen E, SPASS, Paradox, Mace
  zum Lernen: SNoW

Benchmarks

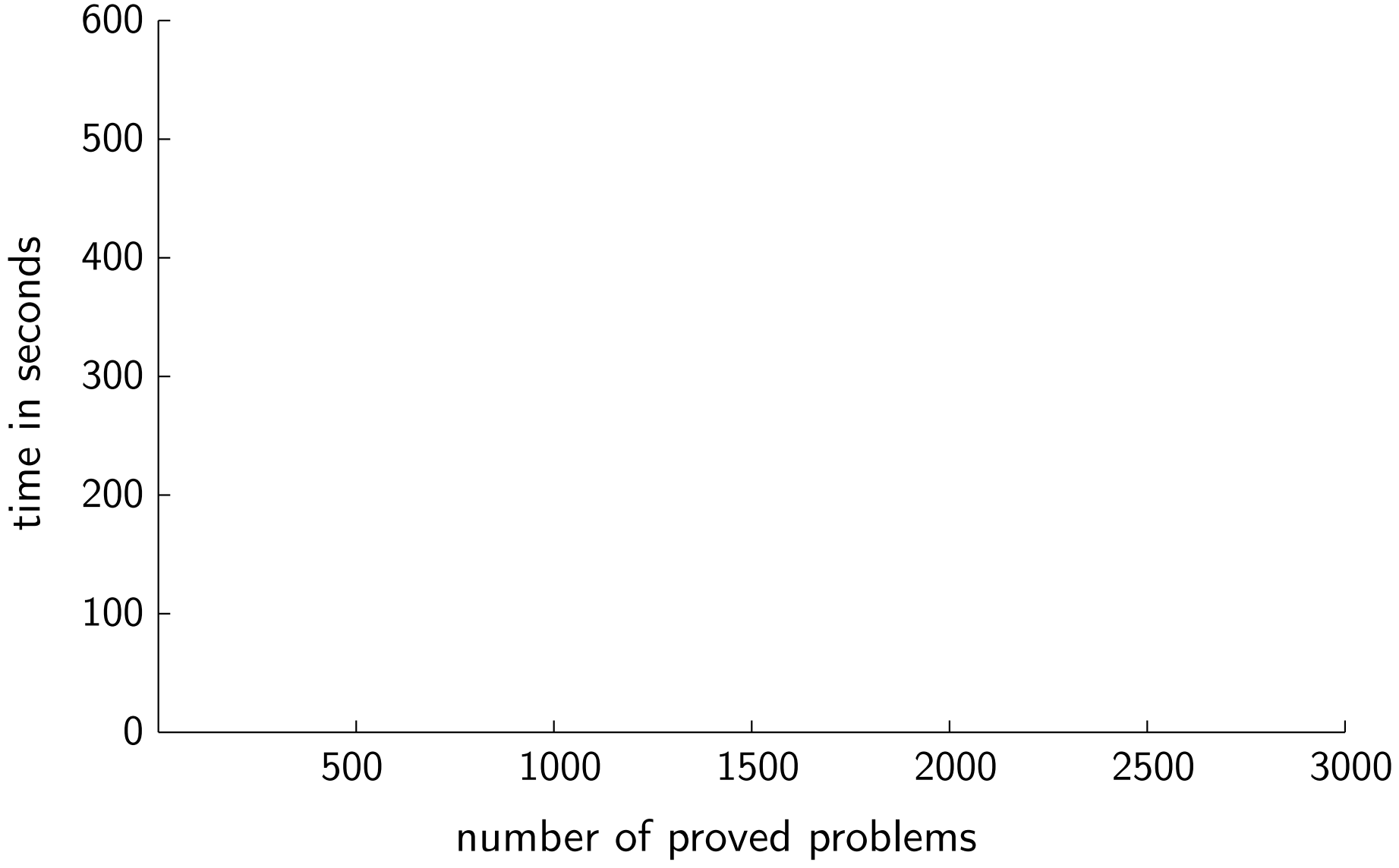
TPTP v3.6.0

- 5051 Probleme
- davon: 1385 large-theory Probleme (Kategorie: LTB)
- Zeitlimit<600 sec.
- Xeon 3.4 Ghz (Cluster)

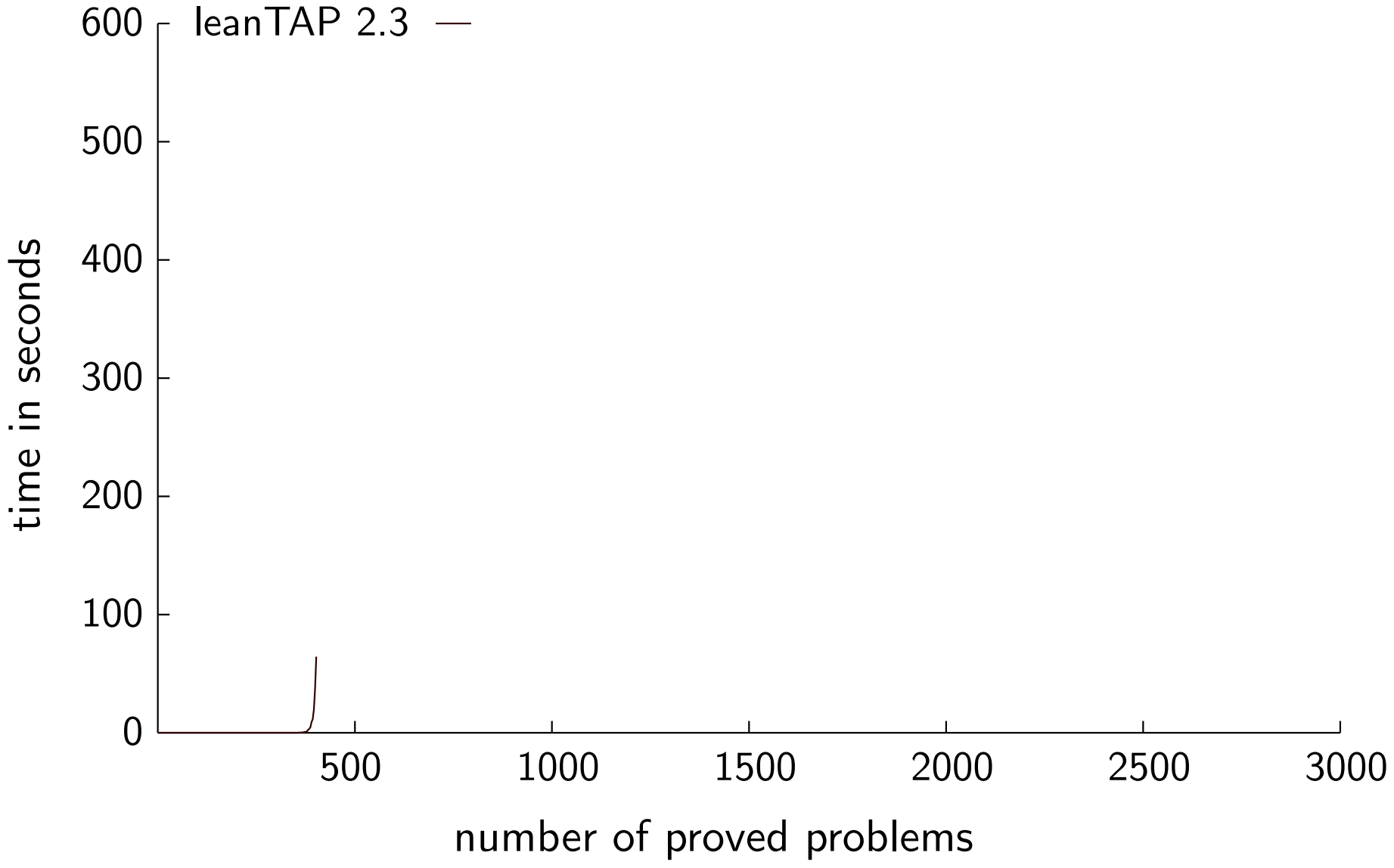
TPTP v3.6.0

Beweiser	bewiesen	wider-legt	SOTAC
Vampire 10.0	2699	-	0.16
E 1.0	2541	372	0.14
SPASS 3.0	2324	367	0.12
iProver 0.5c	2299	359	0.12
Equinox 3.0	2094	-	0.12
randoCoP1.1	1827	30	0.10
leanCoP 2.0	1792	34	0.10
SNARK 08/07	1735	113	0.09
Prover 9 11/08	1647	-	0.09
Metis 2.1	1640	330	0.08
Otter 3.3	1389	-	0.08
Muscadet 3.0	1315	-	0.14
Setheo 3.3	1296	27	0.07
ileanCoP1.2	1272	71	0.08
leanCoP1.0	1105	1	0.07
lean*TAP* 2.3	404	-	0.07

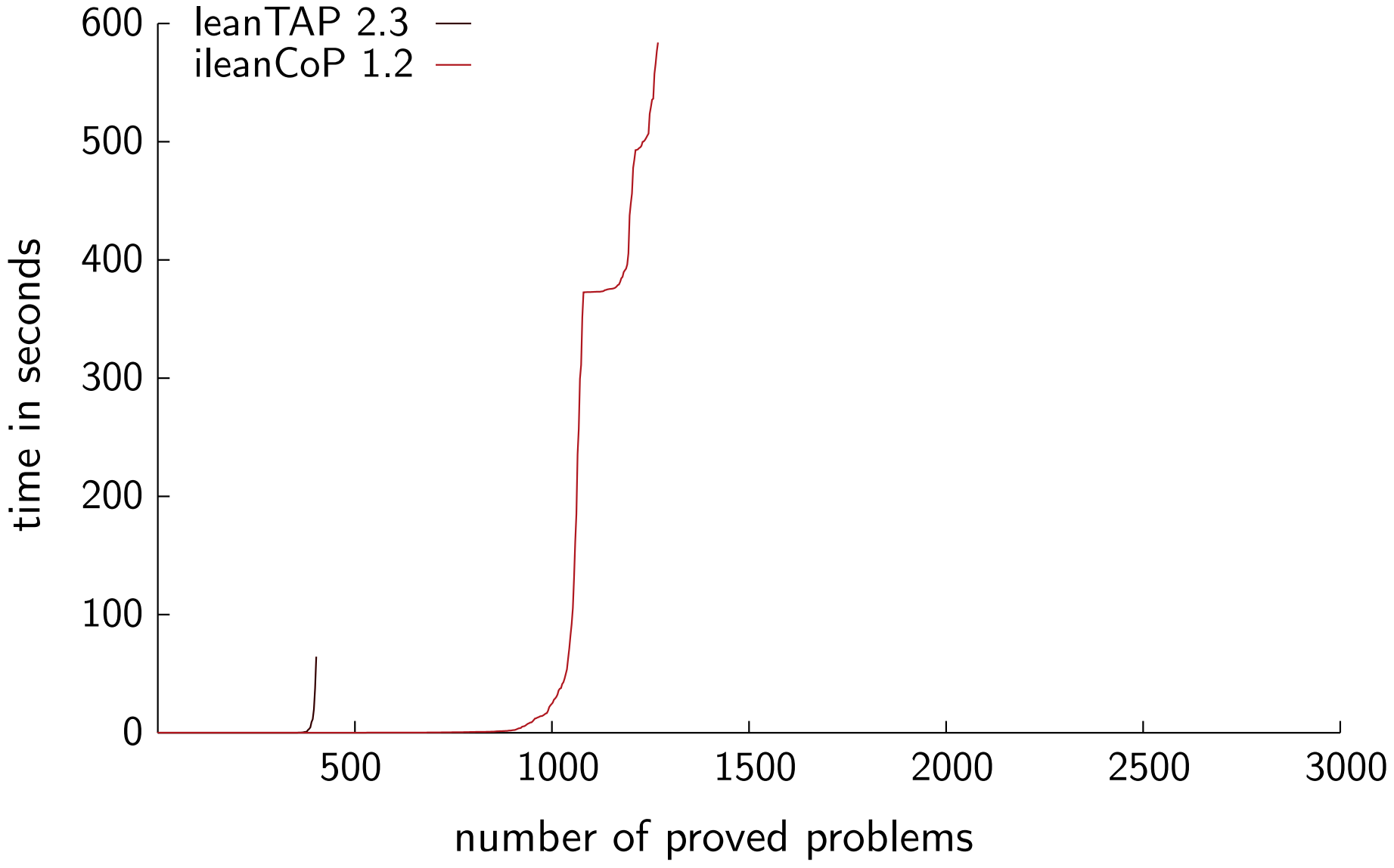
TPTP v3.6.0


600
500
400
300
200
100
0
time in seconds
500
1000
1500
2000
2500
3000
number of proved problems

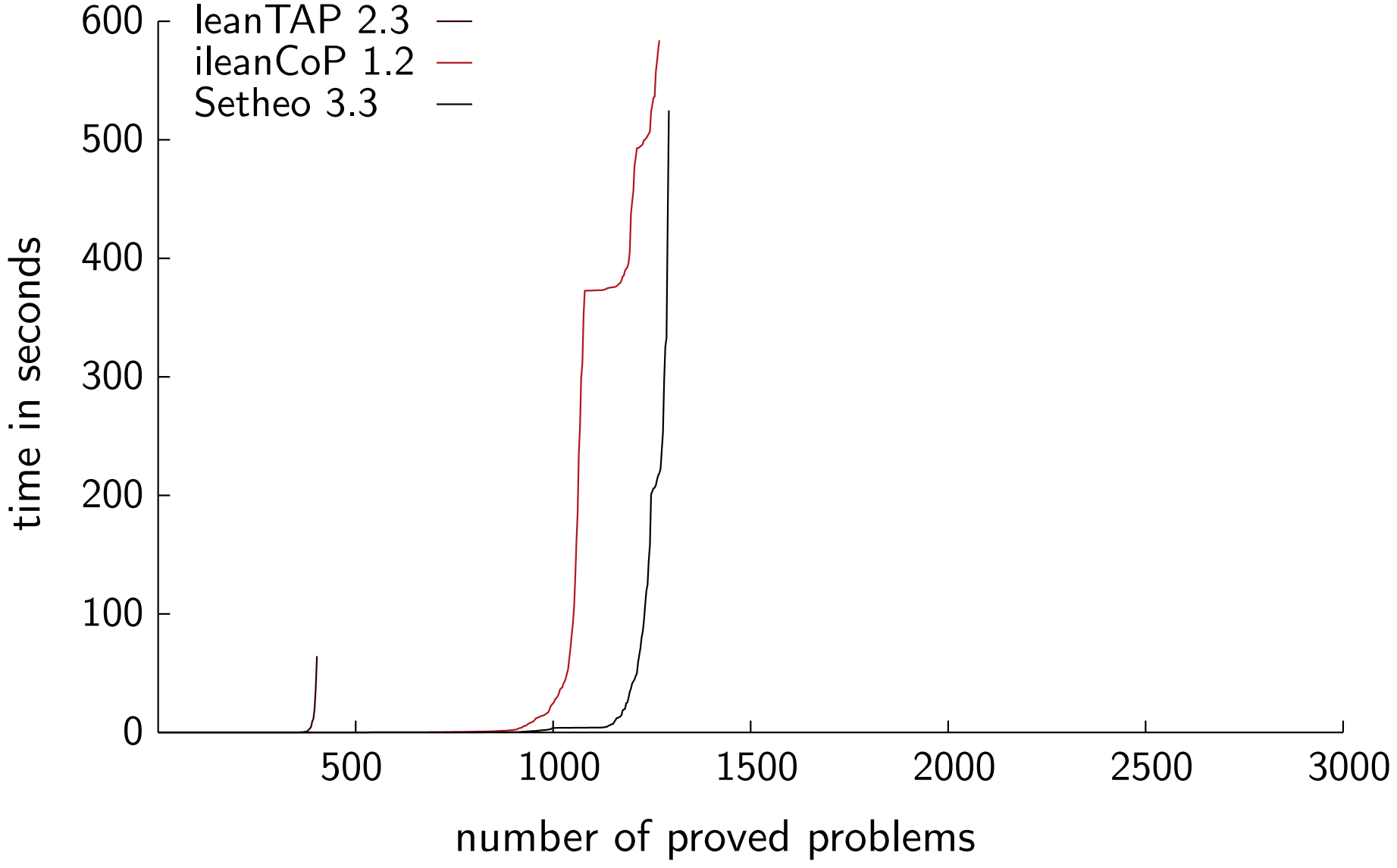
TPTP v3.6.0


leanTAP 2.3
600
500
400
300
200
100
0
time in seconds
500
1000
1500
2000
2500
3000
number of proved problems

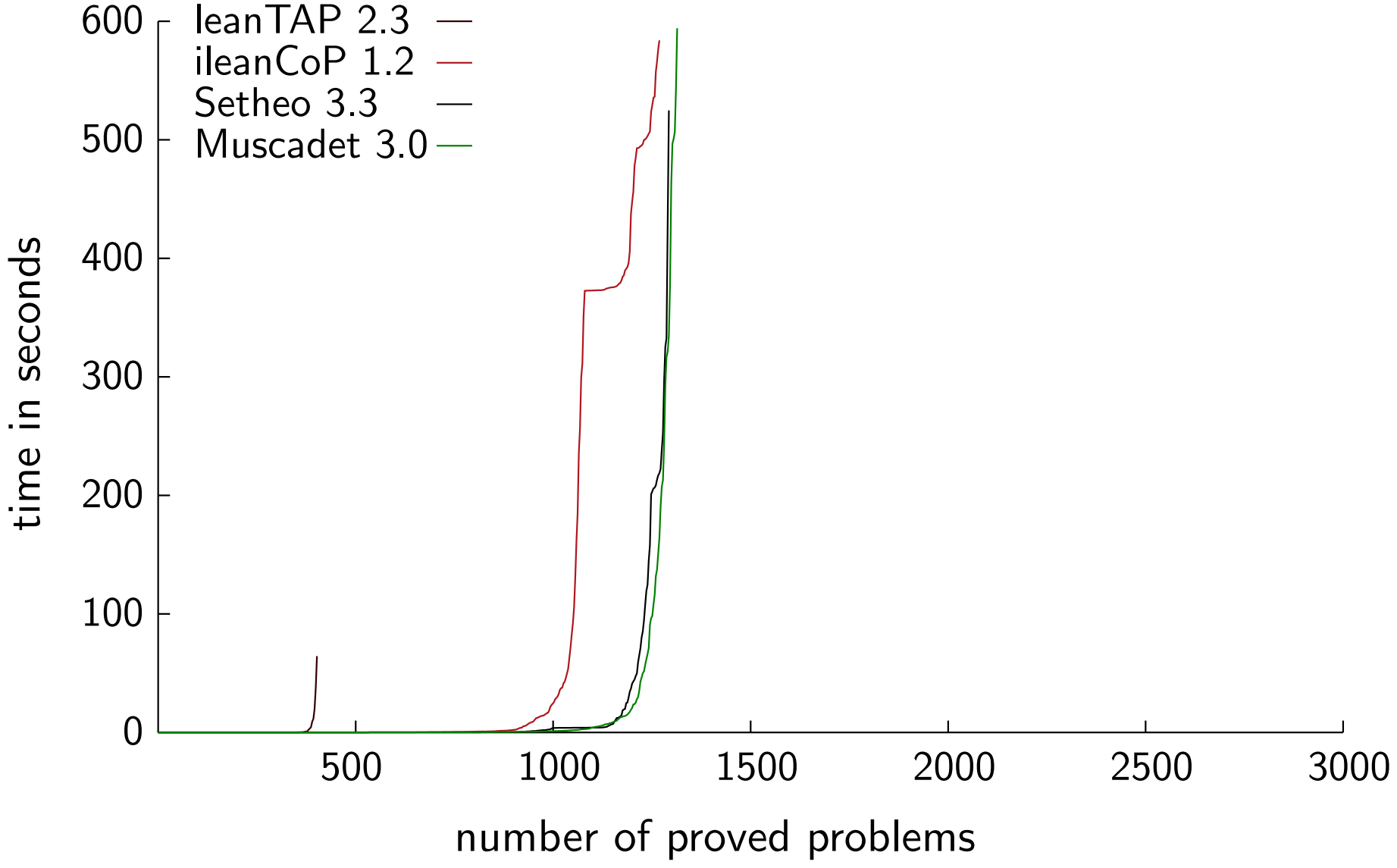
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

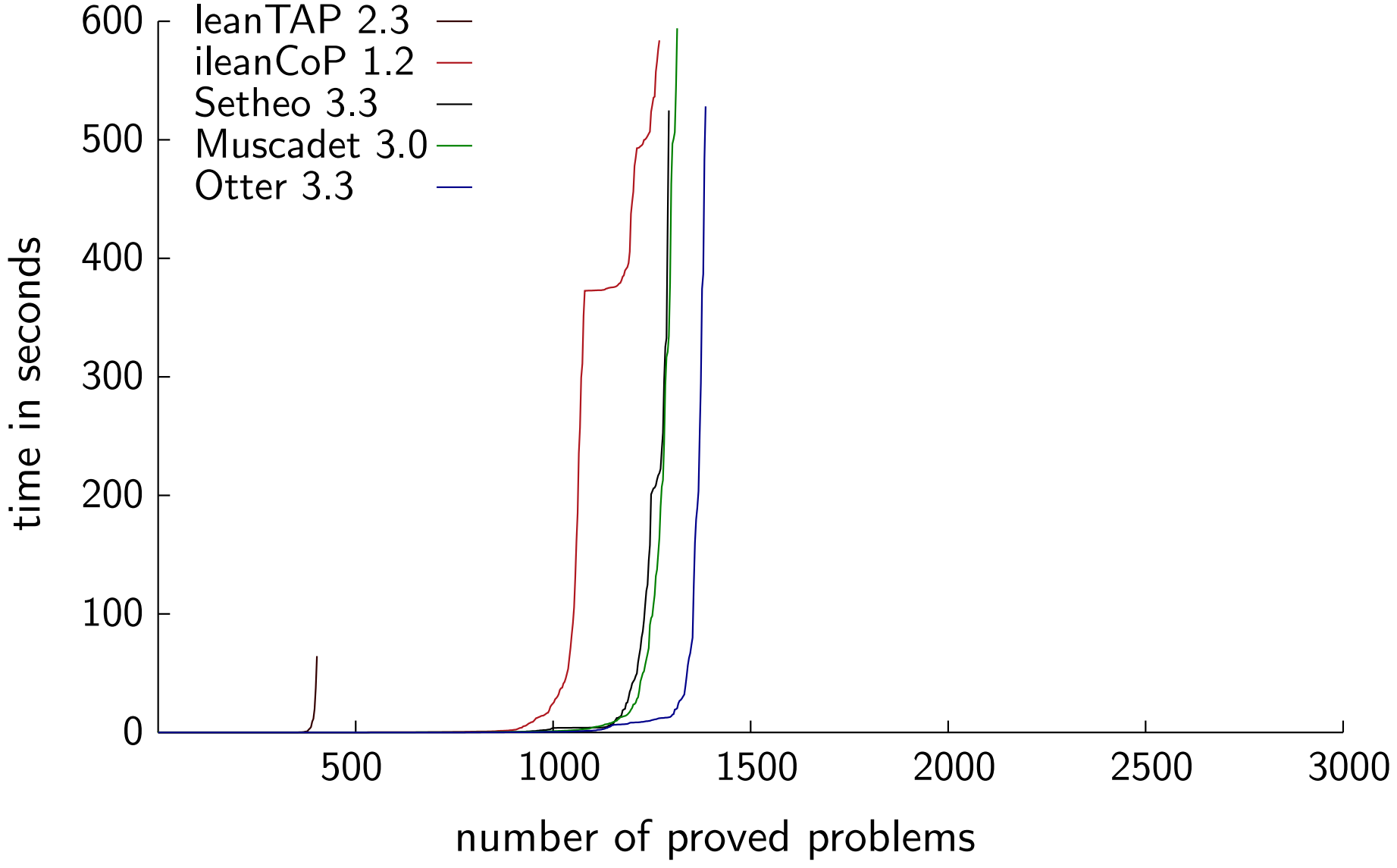
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

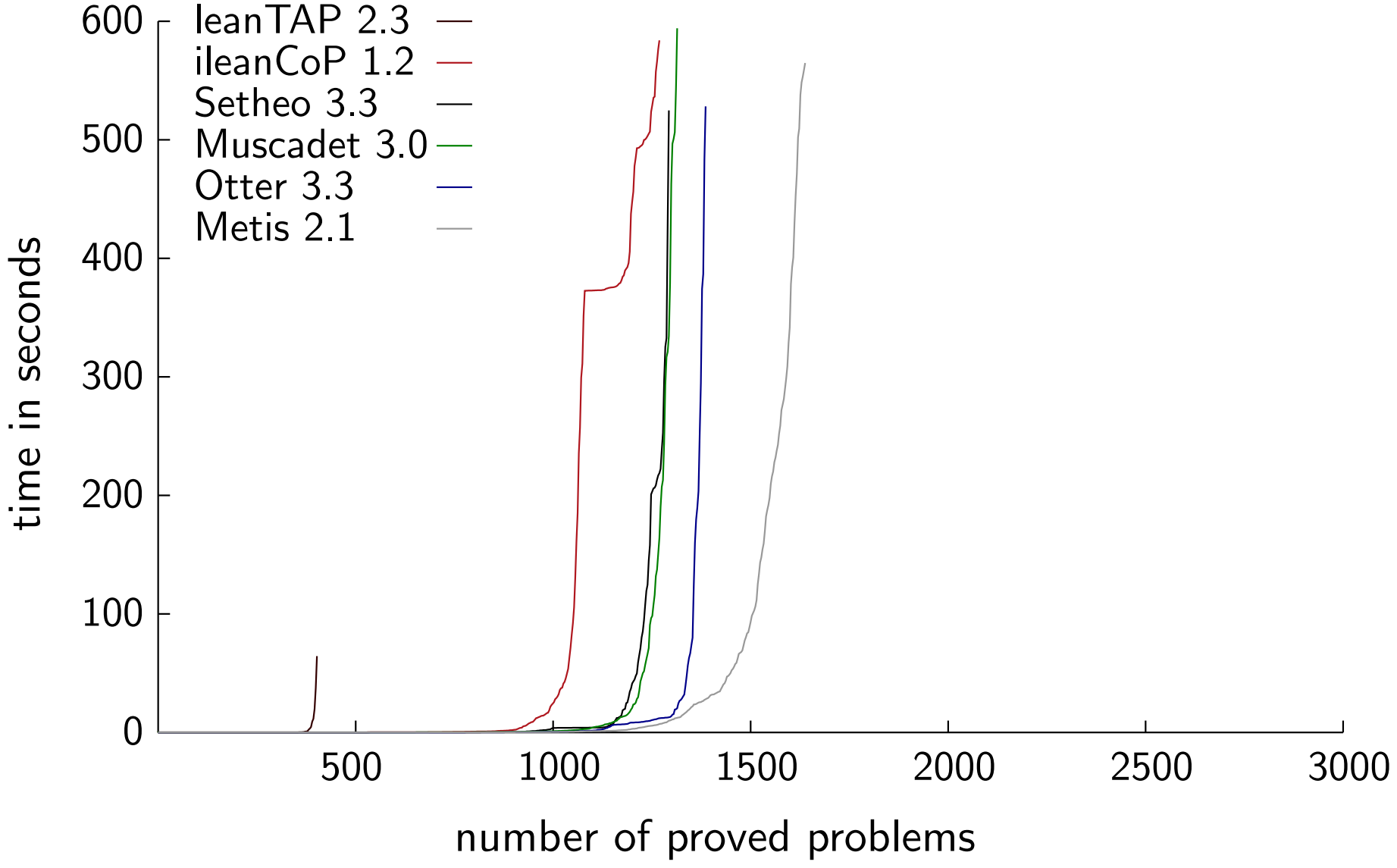
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

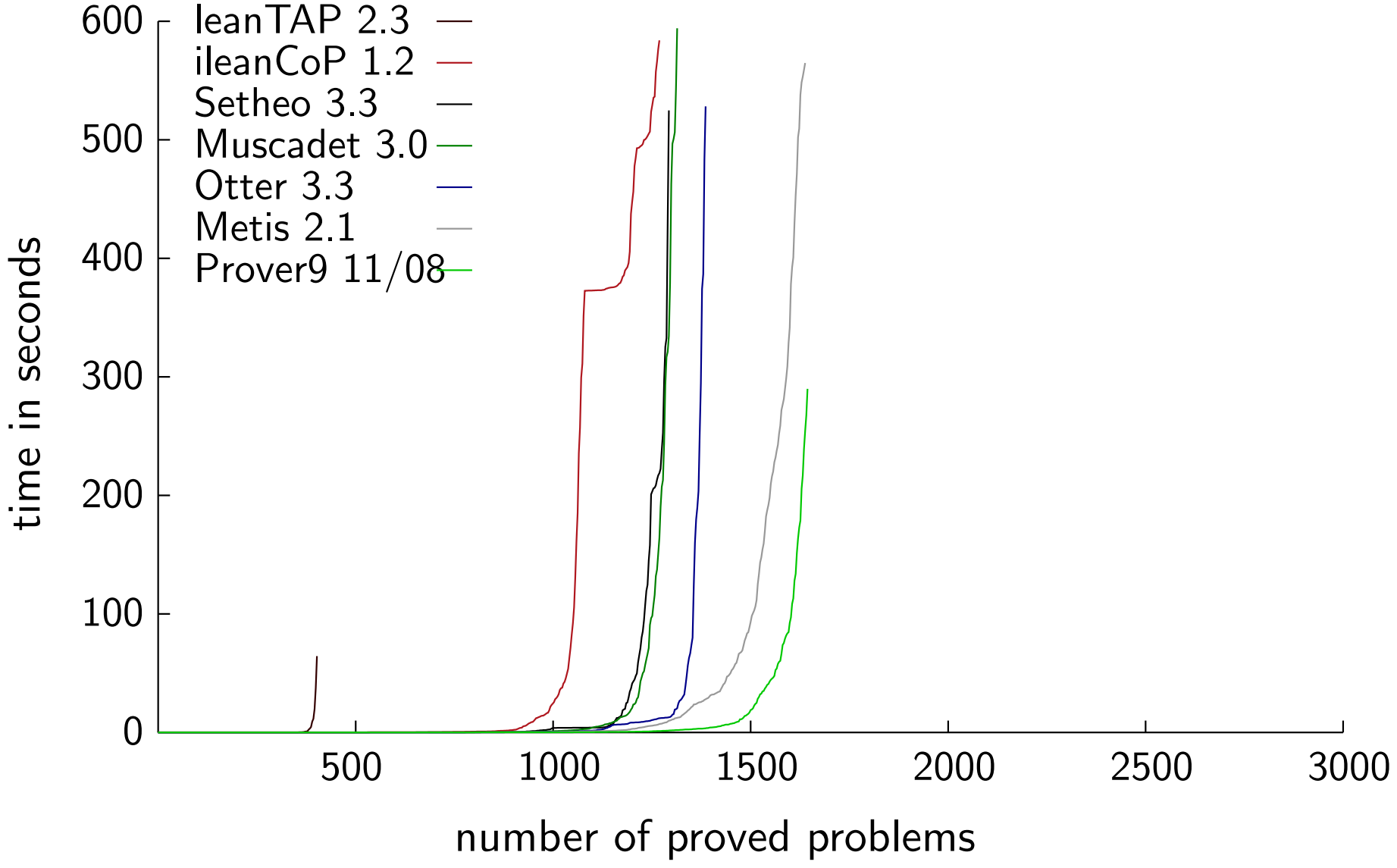
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

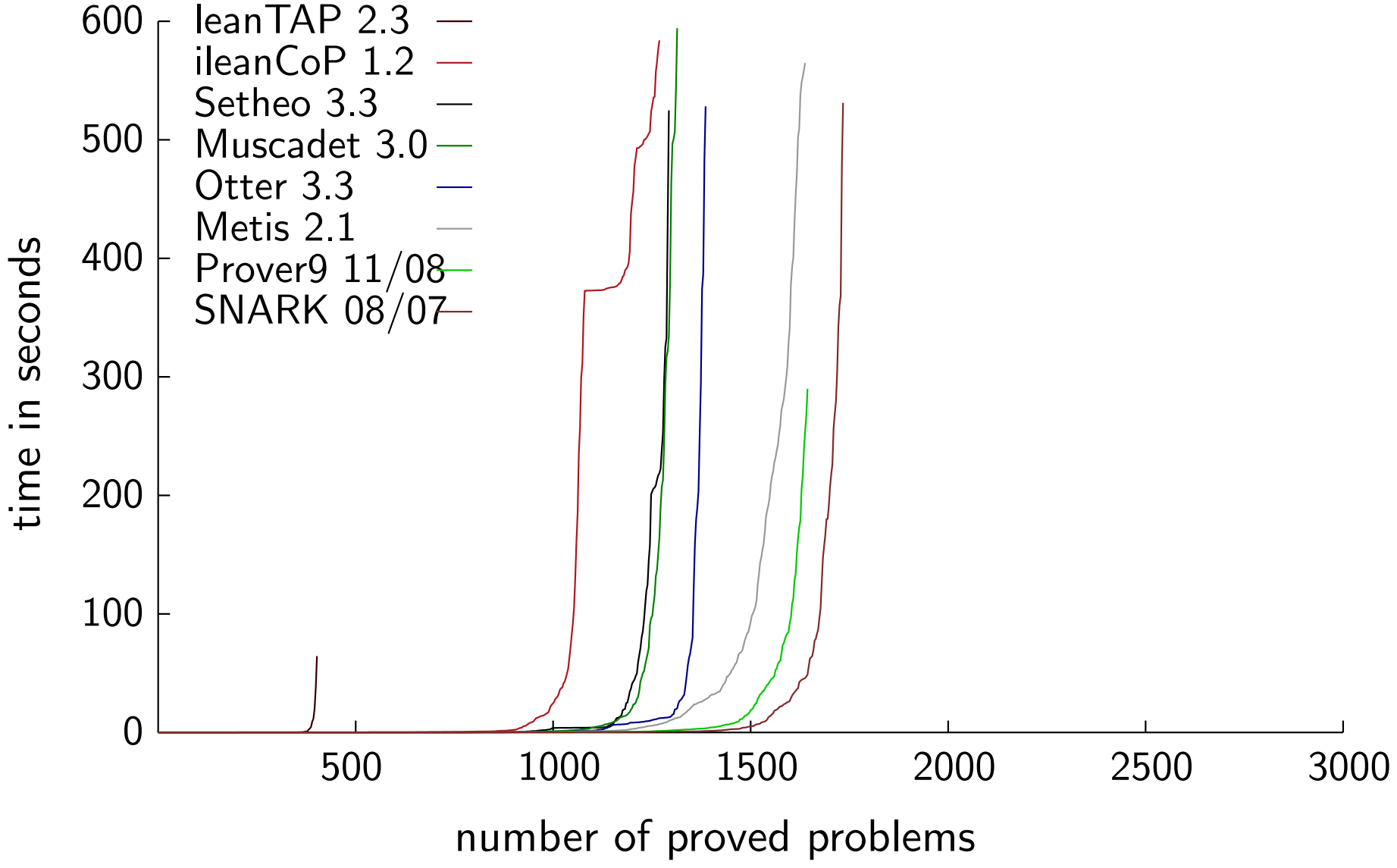
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

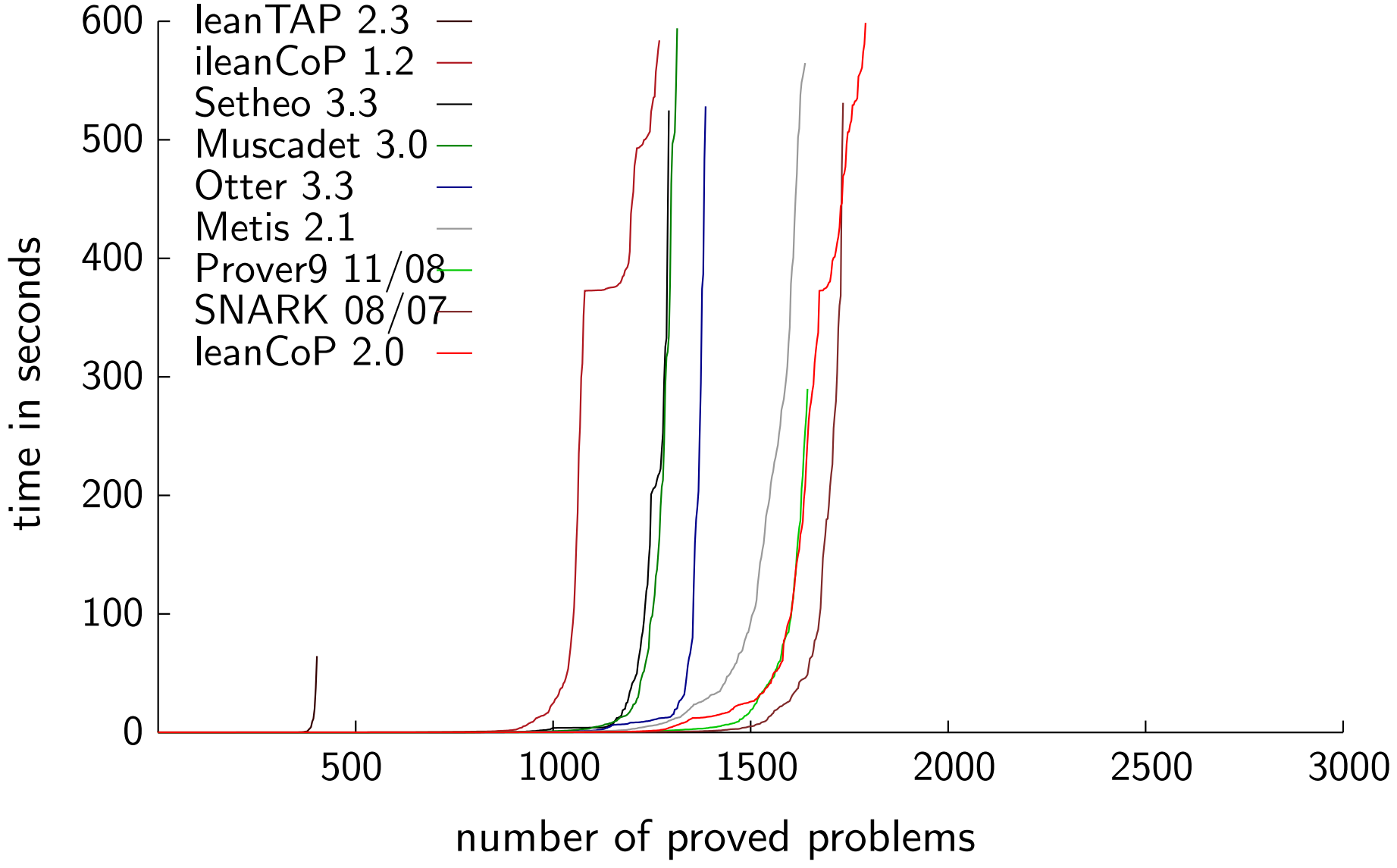
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

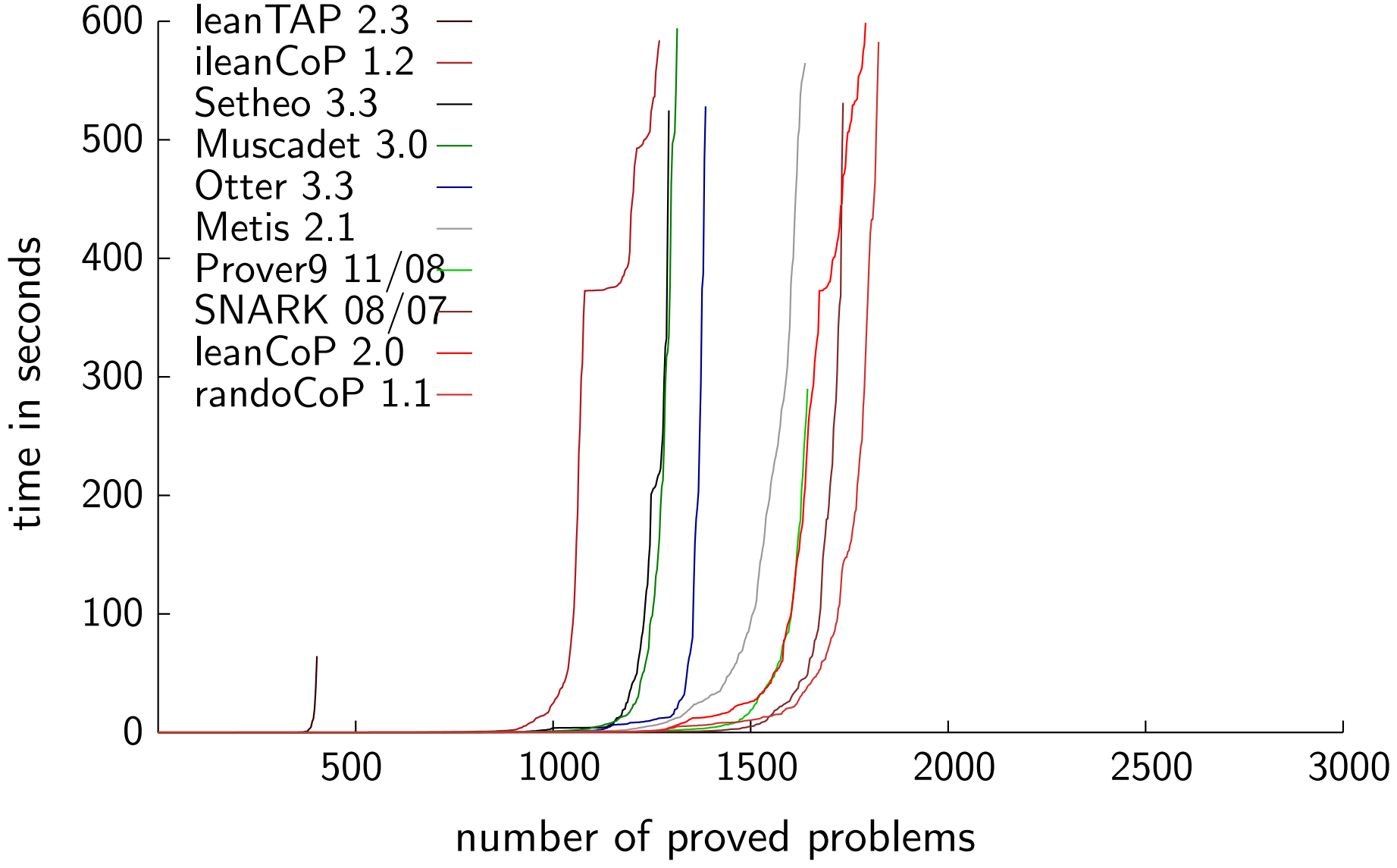
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

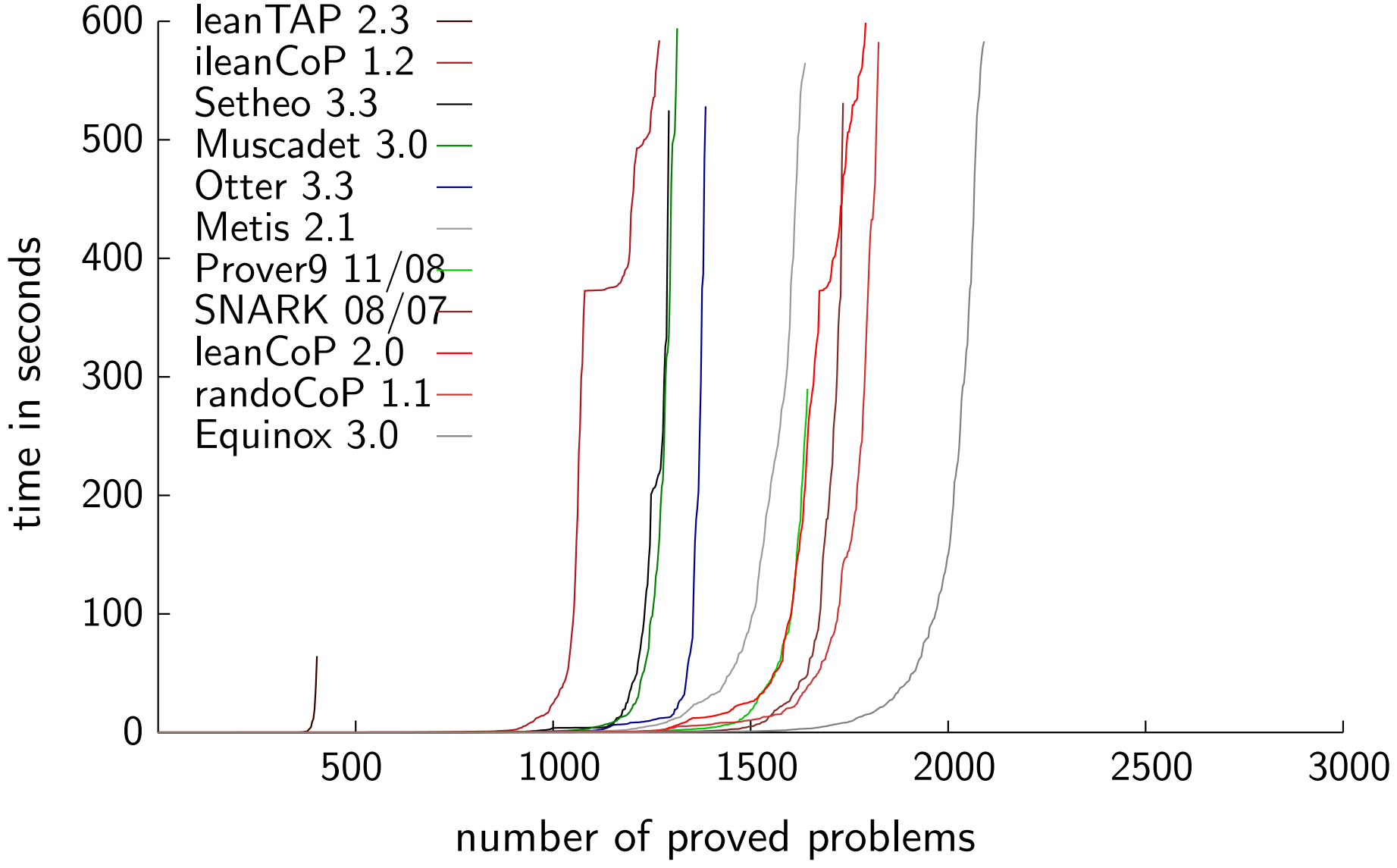
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

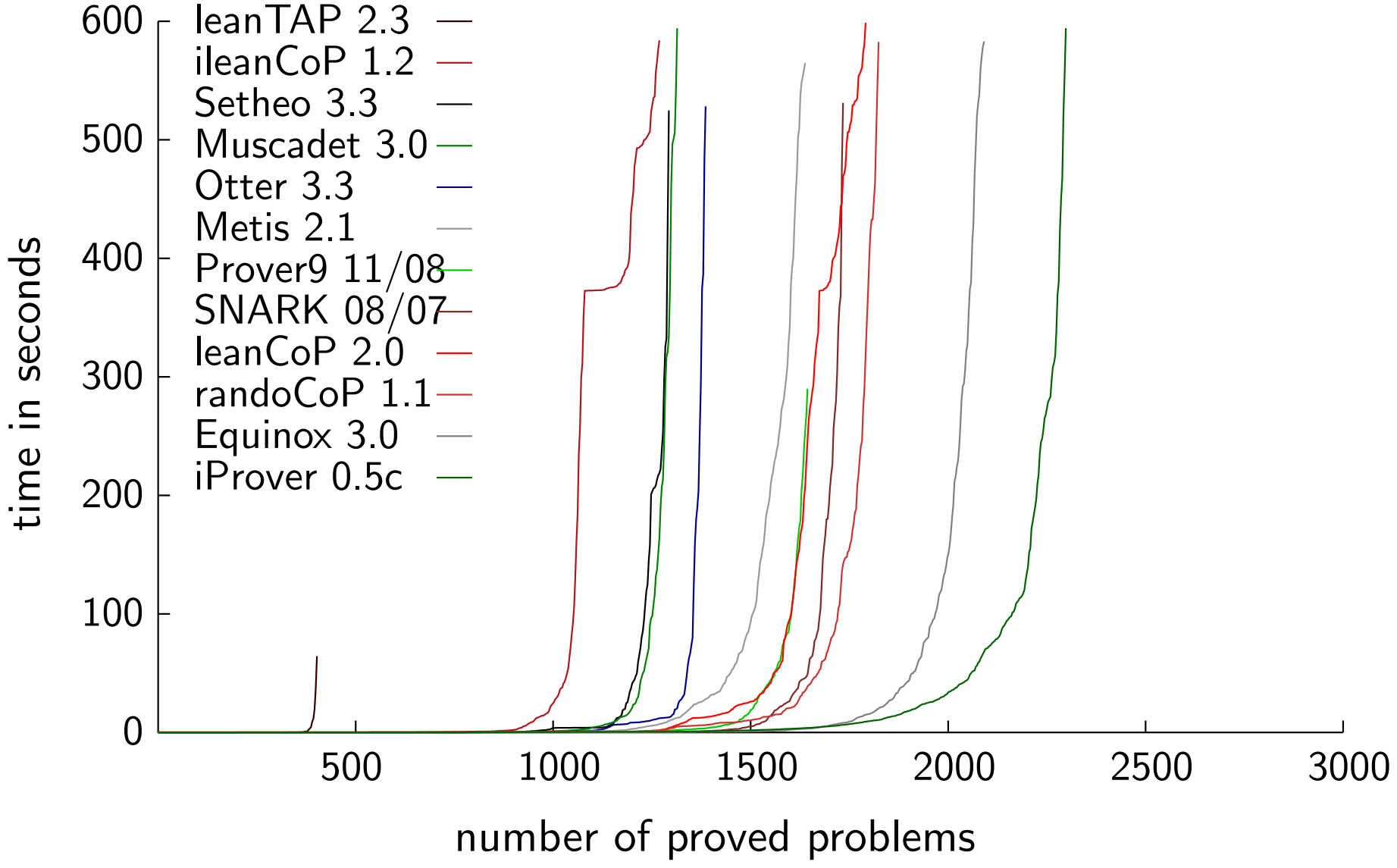
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

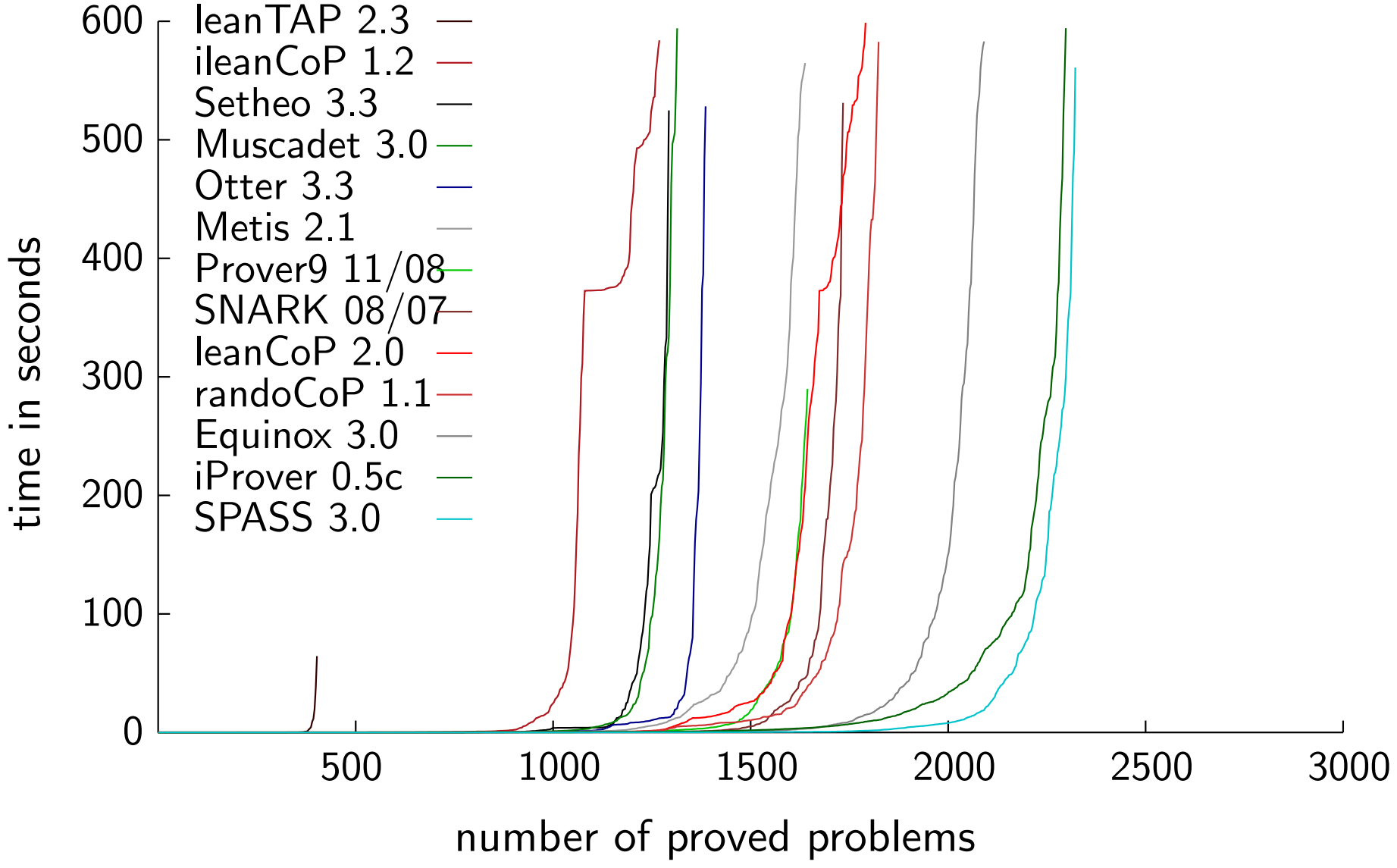
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
Equinox 3.0
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

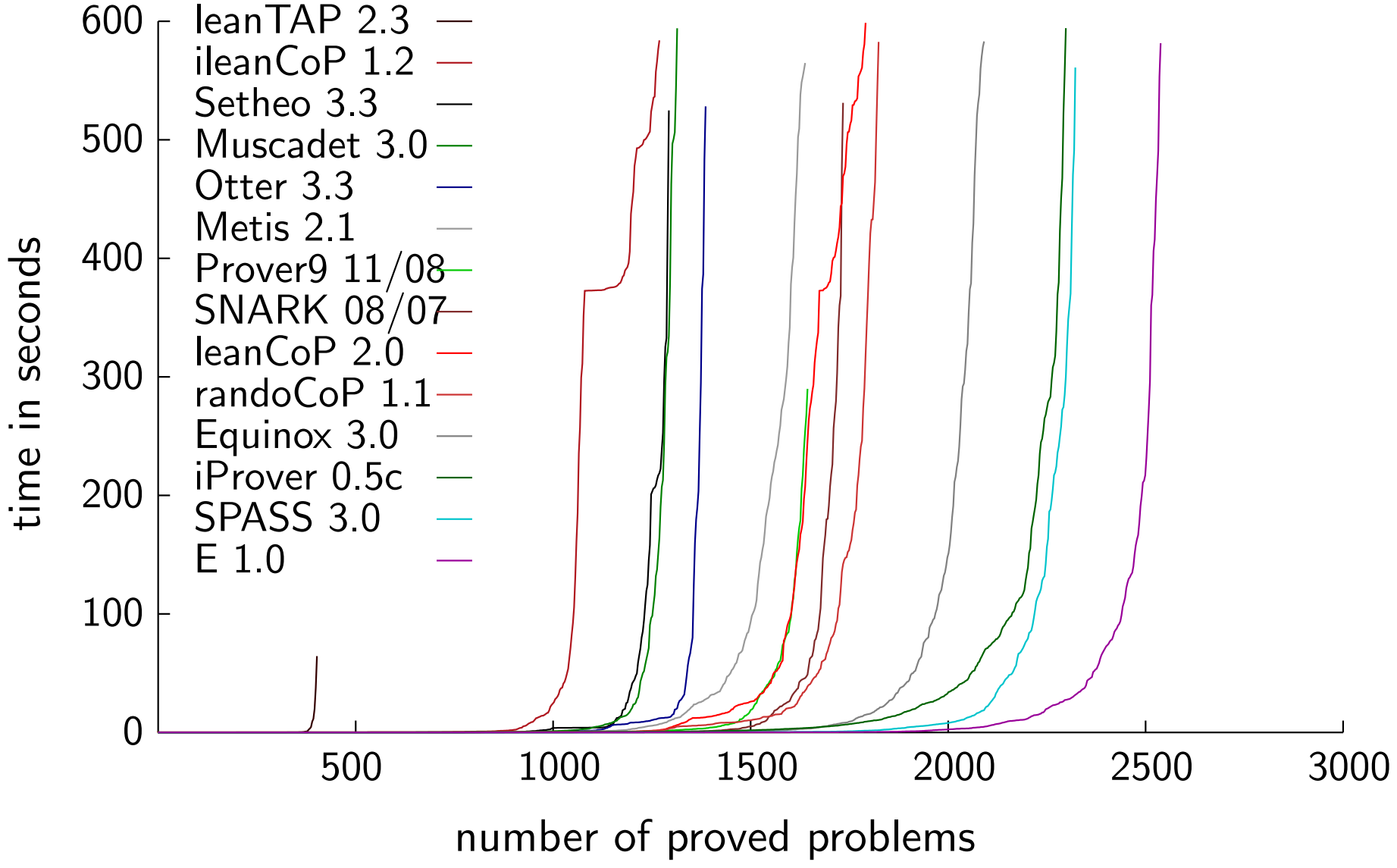
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
Equinox 3.0
iProver 0.5c
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

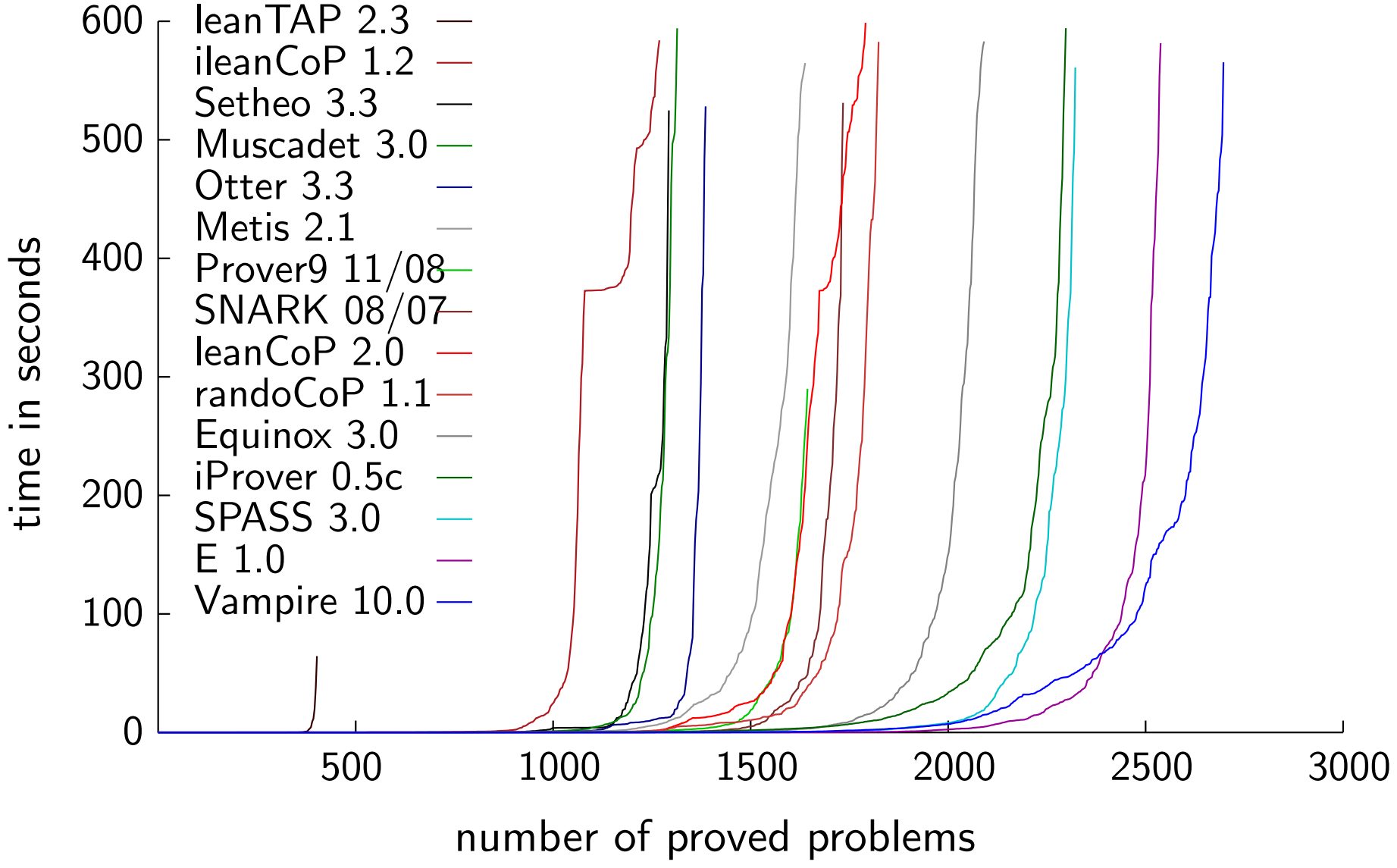
TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
Equinox 3.0
iProver 0.5c
SPASS 3.0
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
Equinox 3.0
iProver 0.5c
SPASS 3.0
E 1.0
time in seconds
number of proved problems
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000

TPTP v3.6.0


leanTAP 2.3
ileanCoP 1.2
Setheo 3.3
Muscadet 3.0
Otter 3.3
Metis 2.1
Prover9 11/08
SNARK 08/07
leanCoP 2.0
randoCoP 1.1
Equinox 3.0
iProver 0.5c
SPASS 3.0
E 1.0
Vampire 10.0
time in seconds
0
100
200
300
400
500
600
500
1000
1500
2000
2500
3000
number of proved problems

CASC-J4, LTB

- CASC-J4: Kategorie LTB:
- 150 Probleme aus den 1385 LTB-Problemen
- Zeitlimit 150 * 240 sec. = 10 h
- Xeon 3.4 Ghz (Cluster)

LTB

Beweiser	LTB	SMO	CYC	MZR
Vampire 10.0	321	34	171	116
iProver 0.5c	321	41	181	99
E 1.0	275	23	161	91
randoCoP1.1	210	7	152	51
Equinox 3.0	209	3	127	79
Metis 2.1	208	29	159	20
SPASS 3.0	193	7	92	94
leanCoP 2.0	181	10	124	47
SNARK 08/07	168	3	99	66
ileanCoP1.2	143	10	122	11
Setheo 3.3	101	0	82	19
leanCoP1.0	100	0	83	17
Muscadet 3.0	85	2	59	24
Otter 3.3	78	0	60	18
Prover 9 11/08	68	0	0	68
lean*TAP* 2.3	15	0	15	0

CASC-J4 LTB

Beweiser	bewiesen	größtes bewiesene Problem		
		Axiome	wie oft	relevant
SInE 0.3 + Vampire 9.0	89	540000	10	1
randoCoP1.1	70	540000	9	1
E 1.0	68	540000	9	1
leanCoP2.1	51	540000	10	1
MaLARea 0.3	51	63000	3	2
iProver 0.5c	43	63000	3	2
Vampire 10.0	35	63000	2	2
E 1.0	34	540000	5	1
randoCoP1.1	21	52000	11	1
Equinox 3.0	17	63000	1	1
SNARK 08/07	14			
Prover 9 11/08	13			
SPASS 3.0	12			
leanCoP 2.0	10			
Metis 2.1	9			
ileanCoP1.2	4			

Fazit und Ausblick

- Auswahl relevanter Axiome entscheidend, dann Beweis einfach
- effiziente Auswahlstrategie implementieren
- leanCoP, randoCoP für SigmaKEE
- leanCoP, randoCoP mit Klausifier von E