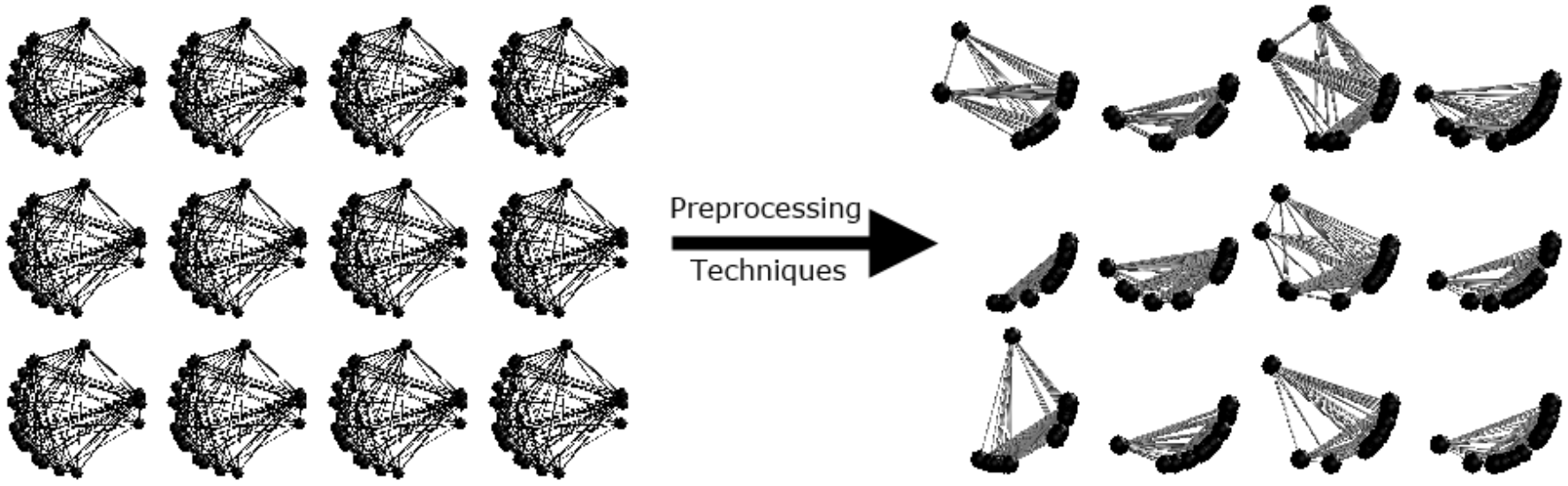


An Extended Analysis of the Utility of Preprocessing Techniques for SAT Solvers

Adrian Balint, Nobert Manthey

08.03.2013

Introduction

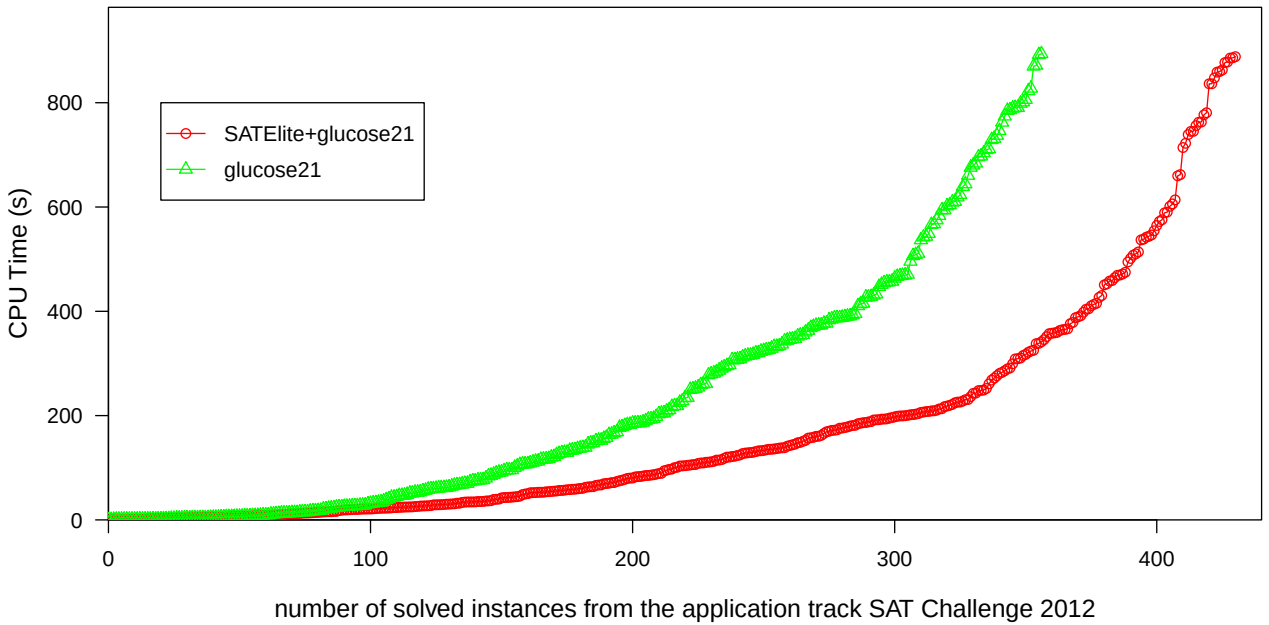
- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)



Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)
- Most widely used PP: SatElite

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)
- Most widely used PP: SatElite
- Several new PPTs presented since the introduction of SatElite

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)
- Most widely used PP: SatElite
- Several new PPTs presented since the introduction of SatElite
- Most PPTs are designed to boost CDCL solvers

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)
- Most widely used PP: SatElite
- Several new PPTs presented since the introduction of SatElite
- Most PPTs are designed to boost CDCL solvers
- There is almost no analysis of modern PPTs for SLS

Introduction

- Preprocessing techniques (PPT) are crucial for the high performance of SAT solvers
- None of the state-of-the-art CDCL solvers affords **not** to use a preprocessor (PP)
- Most widely used PP: SatElite
- Several new PPTs presented since the introduction of SatElite
- Most PPTs are designed to boost CDCL solvers
- There is almost no analysis of modern PPTs for SLS
- SLS solvers seldom applied to structured problems

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Utility of a PPT P for solver S

A PPT P is considered to be useful (or utile) for a solver S on a set of instances I if $perf(S(P(I)) > perf(S(I))$.

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Utility of a PPT P for solver S

A PPT P is considered to be useful (or utile) for a solver S on a set of instances I if $perf(S(P(I)) > perf(S(I))$.

1. *How utile is each PPT on its own?*

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Utility of a PPT P for solver S

A PPT P is considered to be useful (or utile) for a solver S on a set of instances I if $perf(S(P(I)) > perf(S(I))$.

1. *How utile is each PPT on its own?*
2. *Which combination and parametrization of PPTs is best?*

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Utility of a PPT P for solver S

A PPT P is considered to be useful (or utile) for a solver S on a set of instances I if $perf(S(P(I)) > perf(S(I))$.

1. *How utile is each PPT on its own?*
2. *Which combination and parametrization of PPTs is best?*
3. *Can we improve the best PPT with appropriate parametrization?*

Goals of our Work

- Analyze the **utility** of PPTs for CDCL and SLS solvers individually.

Utility of a PPT P for solver S

A PPT P is considered to be useful (or utile) for a solver S on a set of instances I if $perf(S(P(I)) > perf(S(I))$.

1. *How utile is each PPT on its own?*
2. *Which combination and parametrization of PPTs is best?*
3. *Can we improve the best PPT with appropriate parametrization?*
4. *How susceptible is the performance gain when we exchange the solvers S?*

Experimental Setup

- Implement currently available PPTs into a PP framework $\rightarrow$ **Coprocessor 3 (CP3)**

Experimental Setup

- Implement currently available PPTs into a PP framework $\rightarrow$ **Coprocessor 3 (CP3)**
- Use a CDCL solver (glucose 2.1) and a SLS solver (Sparrow)

Experimental Setup

- Implement currently available PPTs into a PP framework $\rightarrow$ **Coprocessor 3 (CP3)**
- Use a CDCL solver (glucose 2.1) and a SLS solver (Sparrow)
- Evaluate on application (for CDCL) and hard combinatorial (for SLS) problems from SAT Challenge 2012 (highly heterogeneous set)

Experimental Setup

- Implement currently available PPTs into a PP framework $\rightarrow$ **Coprocessor 3 (CP3)**
- Use a CDCL solver (glucose 2.1) and a SLS solver (Sparrow)
- Evaluate on application (for CDCL) and hard combinatorial (for SLS) problems from SAT Challenge 2012 (highly heterogeneous set)

Answering Question 1

Evaluate each PPT individually in combination with each solver

Experimental Setup

- Implement currently available PPTs into a PP framework → **Coprocessor 3 (CP3)**
- Use a CDCL solver (glucose 2.1) and a SLS solver (Sparrow)
- Evaluate on application (for CDCL) and hard combinatorial (for SLS) problems from SAT Challenge 2012 (highly heterogeneous set)

Answering Question 1

Evaluate each PPT individually in combination with each solver

Answering Question 2,3

Use automated algorithm configuration tools to configure/tune the PPTs optimizing $perf(S(P(I)))$

Experimental Setup

- Implement currently available PPTs into a PP framework → **Coprocessor 3 (CP3)**
- Use a CDCL solver (glucose 2.1) and a SLS solver (Sparrow)
- Evaluate on application (for CDCL) and hard combinatorial (for SLS) problems from SAT Challenge 2012 (highly heterogeneous set)

Answering Question 1

Evaluate each PPT individually in combination with each solver

Answering Question 2,3

Use automated algorithm configuration tools to configure/tune the PPTs optimizing $perf(S(P(I)))$

Answering Question 2,3

Validate best found PPTs with other solvers

Preprocessing Techniques implemented in CP3

1. Unit Propagation (**UP**)
2. Subsumption (**SUB**)
3. Strengthening (**STR**)
4. (Bounded) Variable Elimination (**BVE**)
5. (Bounded) Variable Addition (**BVA**)
6. Probing (**Probe**)
7. Covered Clause Elimination (**CCE**)
8. Hidden Tautology Elimination (**HTE**)
9. Equivalent Literal Elimination (**EE**)
10. Unhiding (**Unhide**)
11. Ternary Resolution (**3RES**)
12. Add Binary Resolvents (**ADD2**)
13. **Dense**

Preprocessing Techniques implemented in CP3 - Details

3RES - Resolving Ternary Clauses

Resolve ternay clauses with ternary or binary clauses and keep resolvent if the resulting clause is ternary or smaller.

Preprocessing Techniques implemented in CP3 - Details

3RES - Resolving Ternary Clauses

Resolve ternay clauses with ternary or binary clauses and keep resolvent if the resulting clause is ternary or smaller.

BVA - Bounded Variable Addition

Introduces Tseitin variables and reduces the number of clauses.

Preprocessing Techniques implemented in CP3 - Details

3RES - Resolving Ternary Clauses

Resolve ternay clauses with ternary or binary clauses and keep resolvent if the resulting clause is ternary or smaller.

BVA - Bounded Variable Addition

Introduces Tseitin variables and reduces the number of clauses.

ADD2 - Add Binary Resolvents - specially for SLS

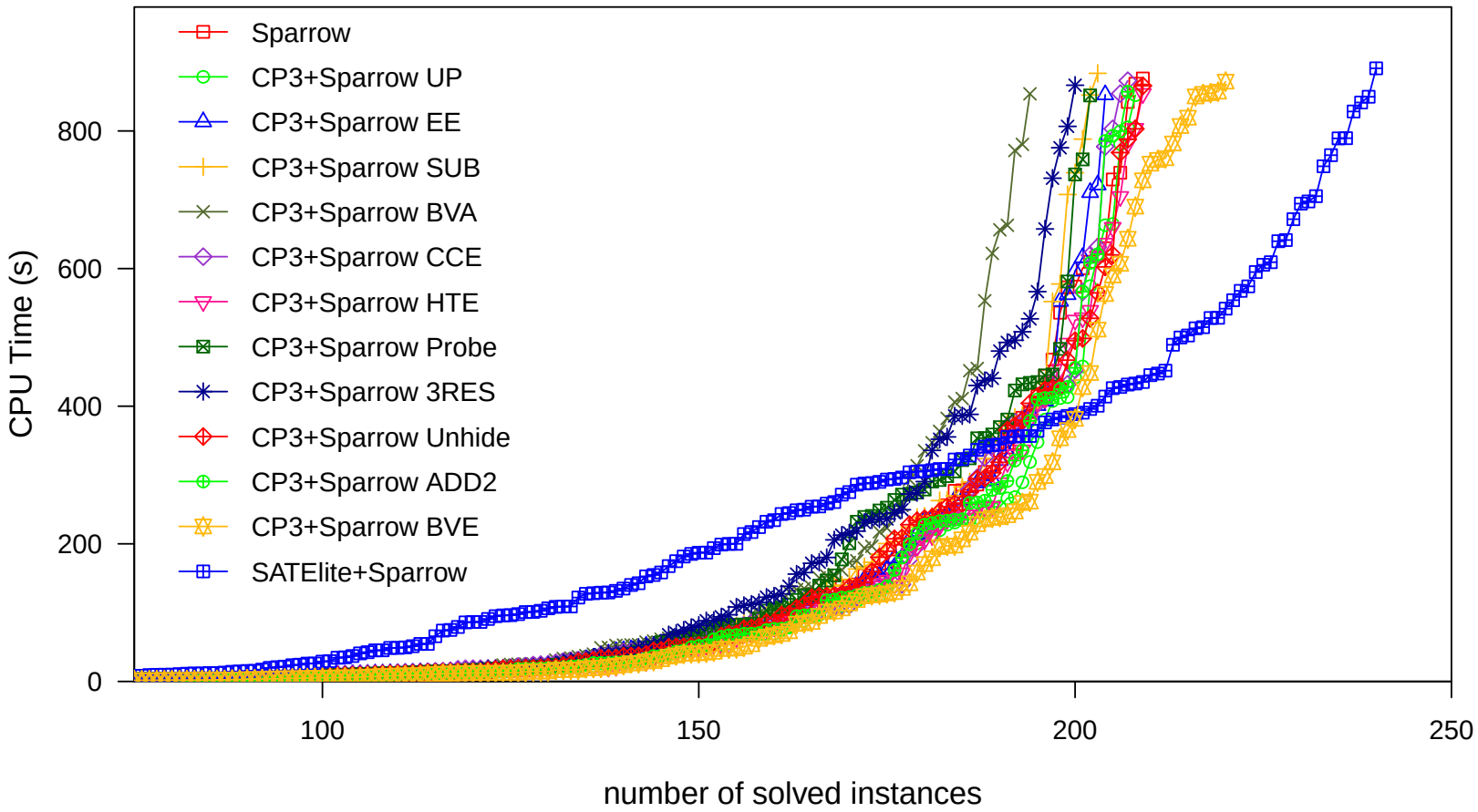
Introduces additionally binary clauses to "‘short-cut"’ implication chains.

PPT for SLS - CP3+Sparrow on HC12 - Single PPT Analysis

- Each PPT evaluated individually with standard parametrization

PPT for SLS - CP3+Sparrow on HC12 - Single PPT Analysis

- Each PPT evaluated individually with standard parametrization

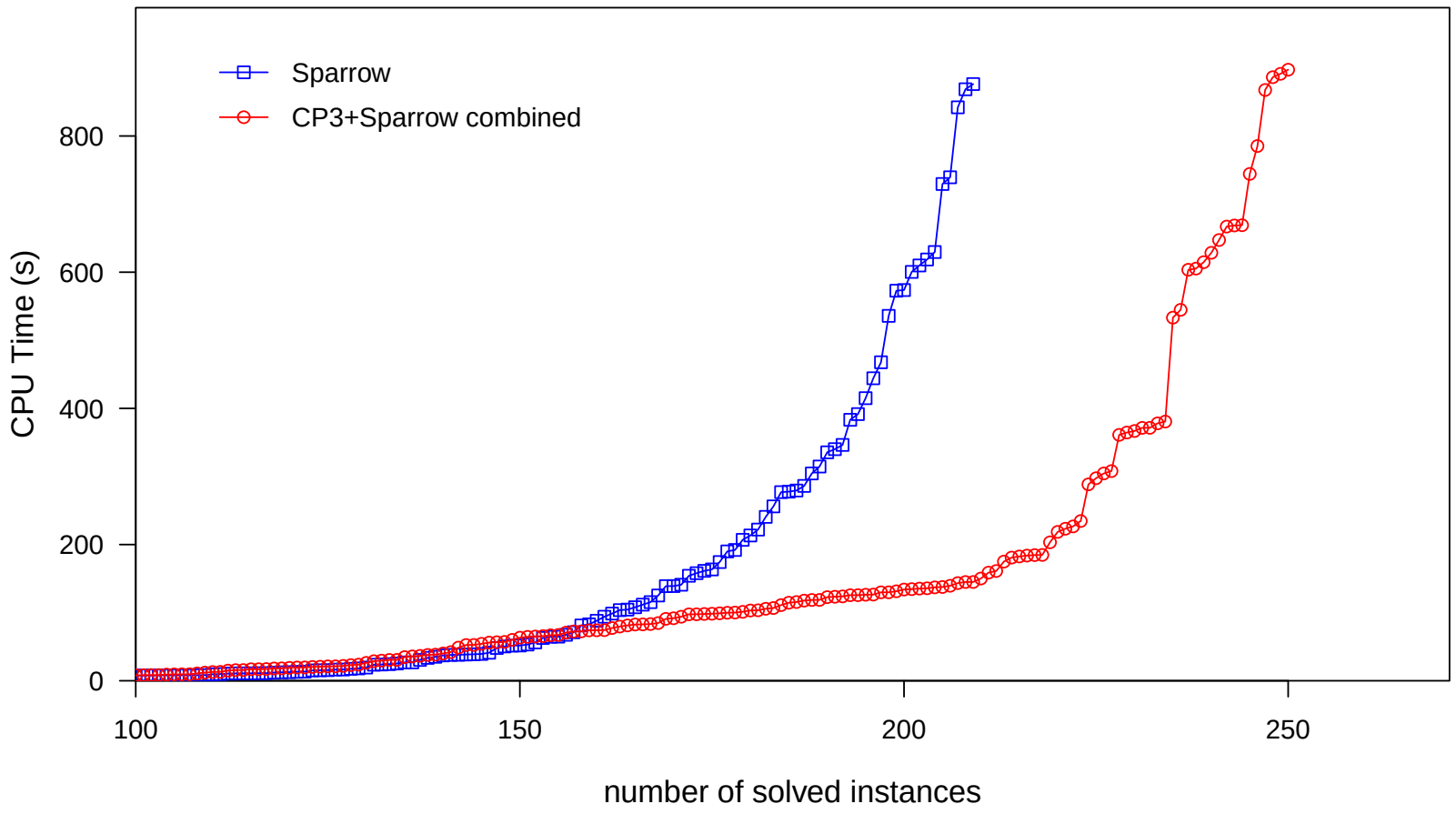


PPT for SLS - CP3+Sparrow on HC12 - Combined PPT Analysis

- Optimize the parameters of all PPTs keeping the order fixed

PPT for SLS - CP3+Sparrow on HC12 - Combined PPT Analysis

- Optimize the parameters of all PPTs keeping the order fixed



PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Best single PPT: Bounded Variable Elimination (BVE)

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Best single PPT: Bounded Variable Elimination (BVE)
- Further extend BVE with novel ideas

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Best single PPT: Bounded Variable Elimination (BVE)
- Further extend BVE with novel ideas
  - Allow an increase of the number of literals/clauses per elimination step up to an upper bound

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Best single PPT: Bounded Variable Elimination (BVE)
- Further extend BVE with novel ideas
  - Allow an increase of the number of literals/clauses per elimination step up to an upper bound
  - Define new orders for the variable selection heuristic (e.g.: maximum occurrence, ratio between negative/positive occurrences, . . .)

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Best single PPT: Bounded Variable Elimination (BVE)
- Further extend BVE with novel ideas
  - Allow an increase of the number of literals/clauses per elimination step up to an upper bound
  - Define new orders for the variable selection heuristic (e.g.: maximum occurrence, ratio between negative/positive occurrences, . . .)

Results of the optimization of BVE for SLS

- Choose variable with maximum occurrence first (totally contrary to BVE for CDCL: choose variable with minimum occurrence first)
- Allow the formula to increase up to 10 clauses per step and totally not more than 1000 clauses

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

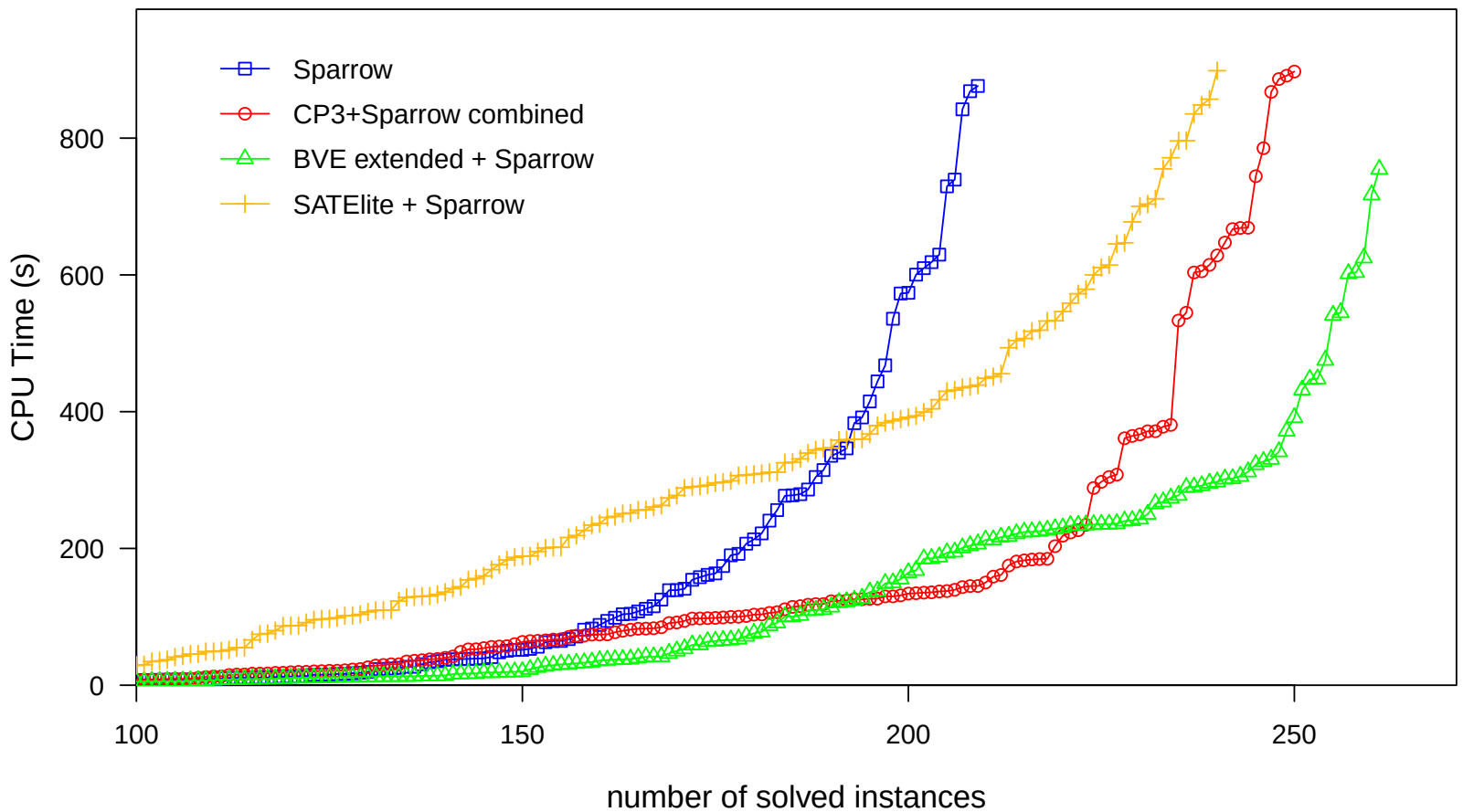
- Optimize the new parameters of BVE

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Optimize the new parameters of BVE
- Results compared to SatElite+Sparrow which is also mainly using BVE (but in a CDCL friendly way)

PPT for SLS - CP3+Sparrow on HC12 - Extended PPT Analysis

- Optimize the new parameters of BVE
- Results compared to SatElite+Sparrow which is also mainly using BVE (but in a CDCL friendly way)

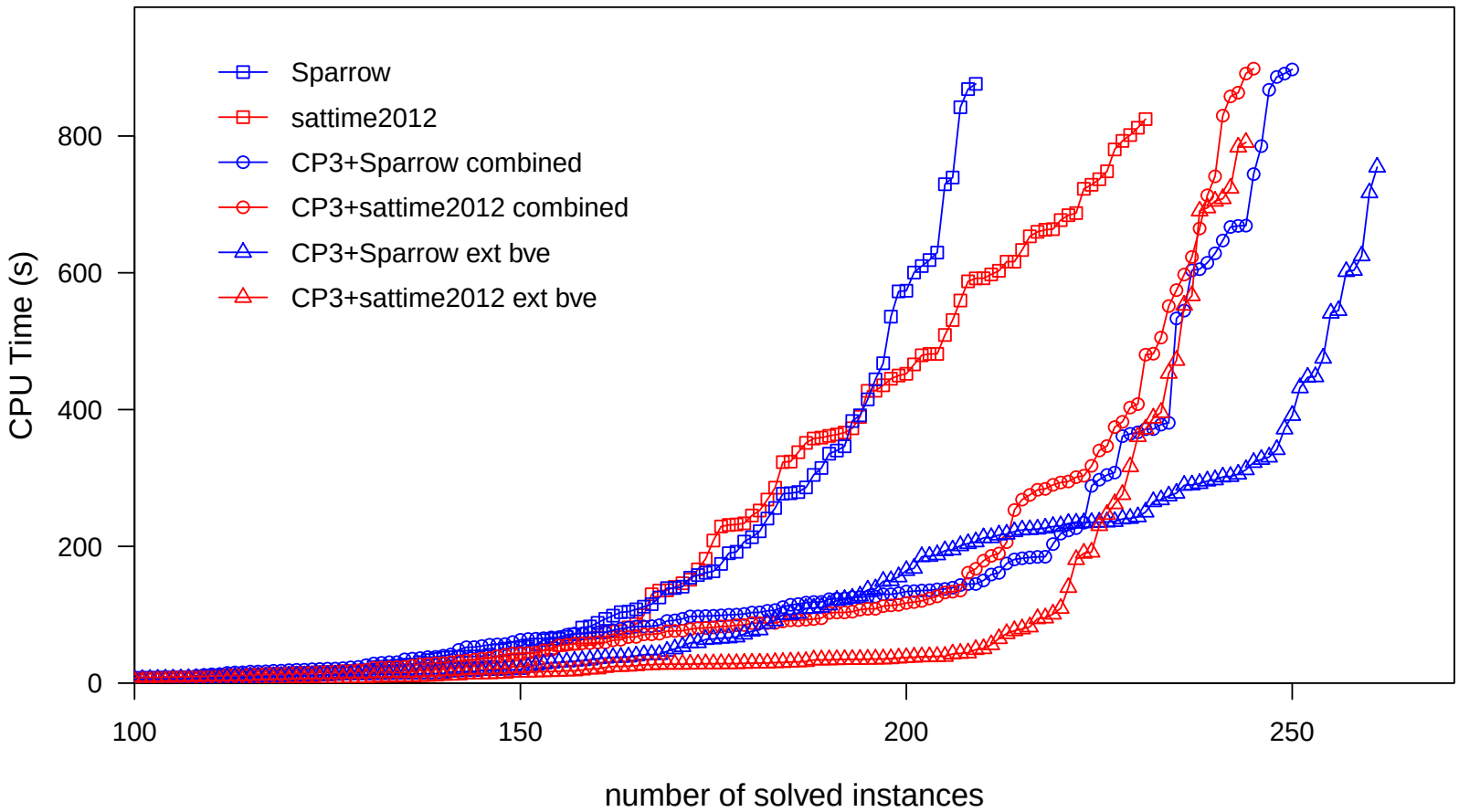


PPT for SLS - Applicability to other Solvers

- Replace Sparrow with sattime2012

PPT for SLS - Applicability to other Solvers

- Replace Sparrow with sattime2012

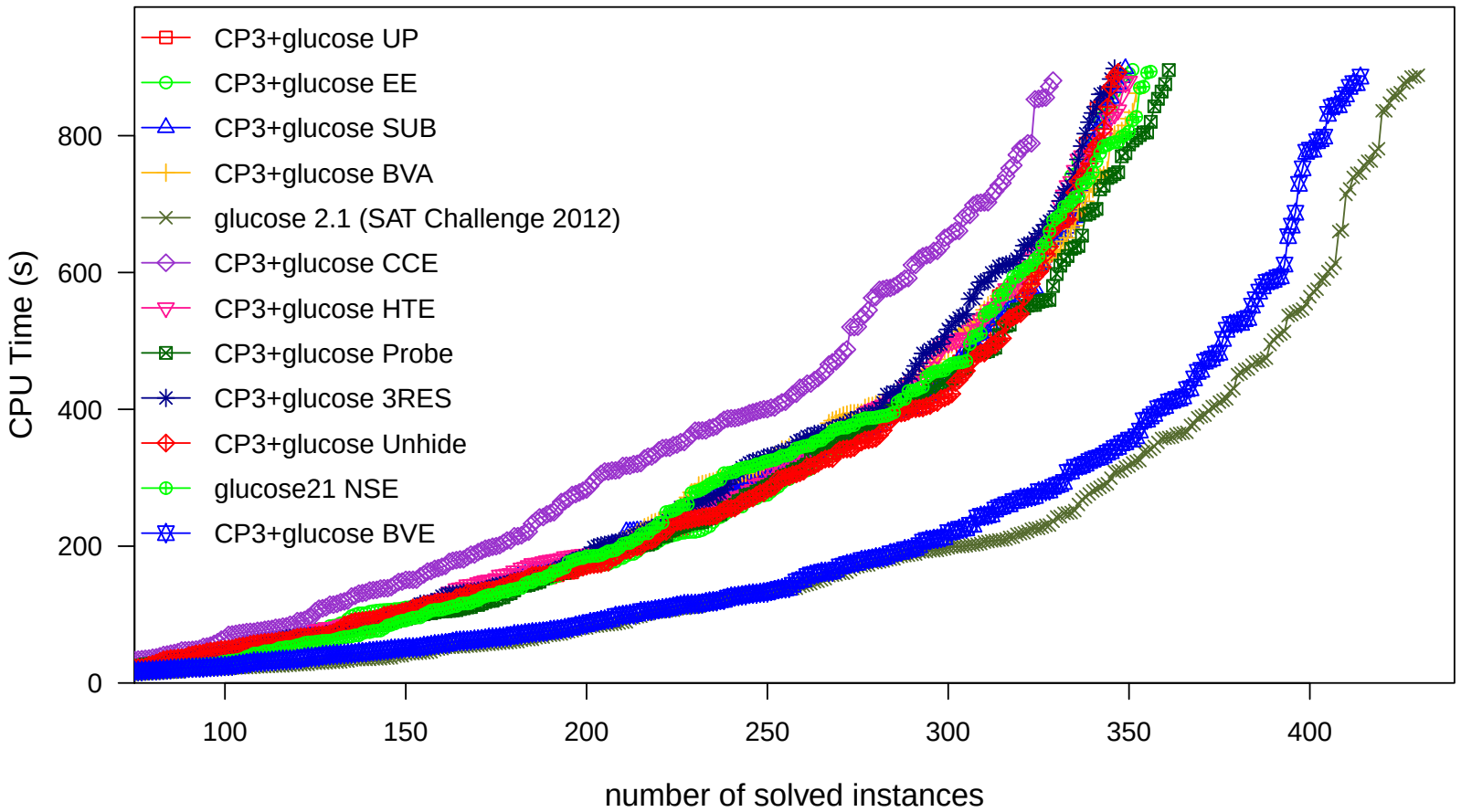


PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- Each PPT evaluated individually with standard parametrization

PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- Each PPT evaluated individually with standard parametrization



PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- The graph shows only the overall utility

PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- The graph shows only the overall utility

Unique PPT contribution (UPT)

Similar to "'unique solver contribution"': the number of instances that can be solved only when using a certain PPT.

PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- The graph shows only the overall utility

Unique PPT contribution (UPT)

Similar to "'unique solver contribution"': the number of instances that can be solved only when using a certain PPT.

	None	UP	3RES	SUB+STR	EE	Unhide	HTE	Probe	BVE	BVA	CCE
UPT	2	–	–	1	6	–	1	5	52	2	2
solved	356	347	346	349	351	347	350	361	414	352	329

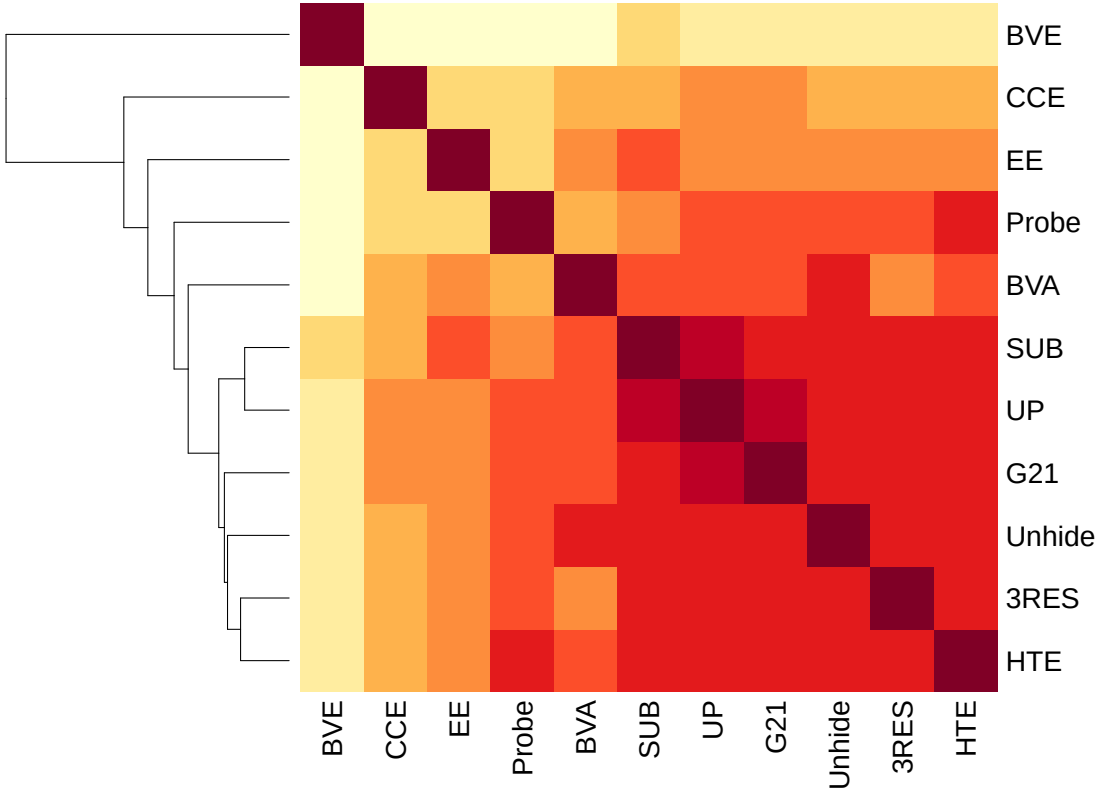
Single UPT contribution of each PPT.

PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- How similar are the individual PPTs with respect to performance?

PPT for CDCL - CP3+glucose 2.1 - Single PPT Analysis

- How similar are the individual PPTs with respect to performance?



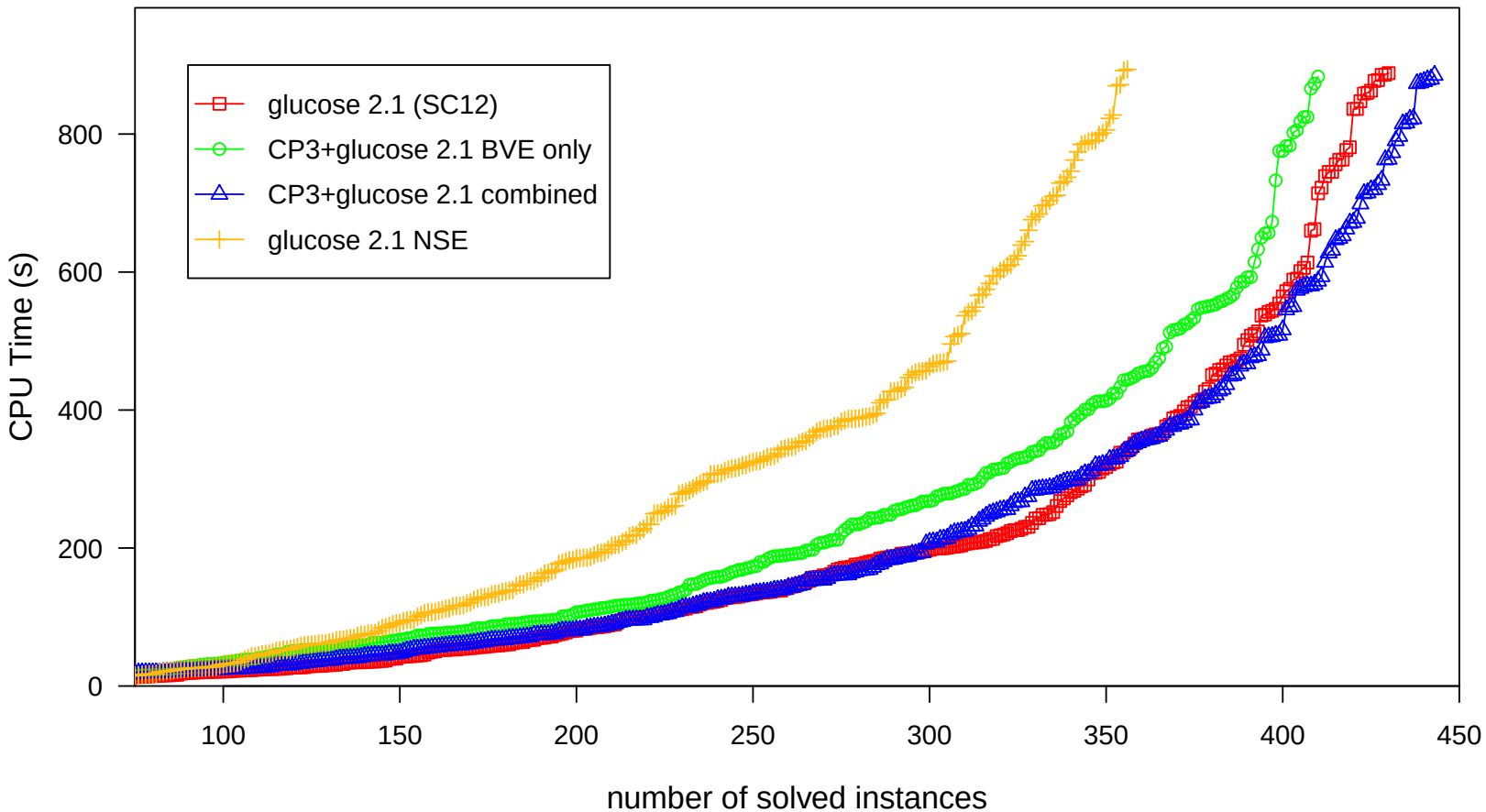
Heatmap visualization of the Spearman correlation matrix

PPT for CDCL - CP3+glucose 2.1 - Combined PPT Analysis

- Optimize the parameters of each PPT keeping the order fixed

PPT for CDCL - CP3+glucose 2.1 - Combined PPT Analysis

- Optimize the parameters of each PPT keeping the order fixed



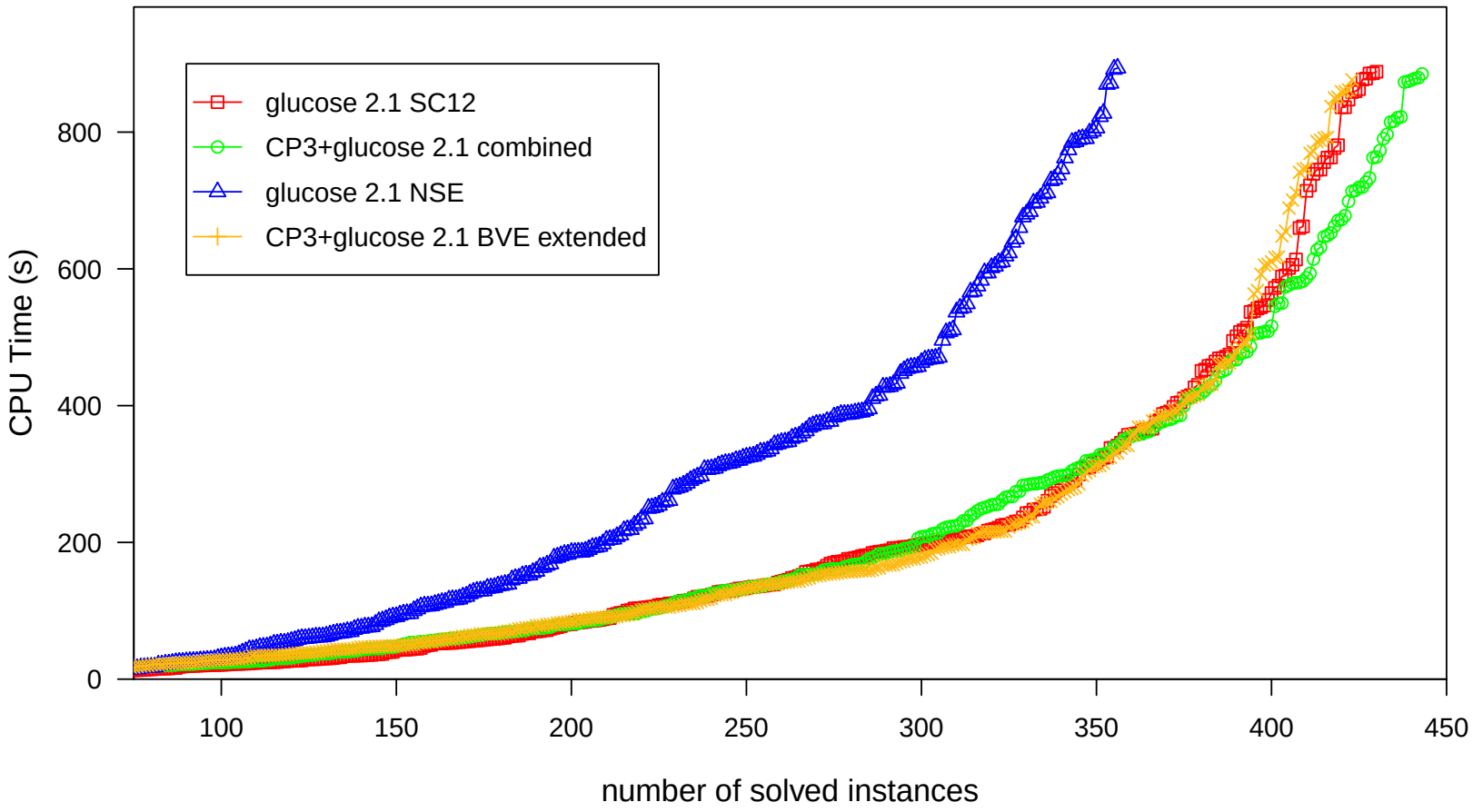
Best configuration uses Unhide+BVE+BVA.

PPT for CDCL - CP3+glucose 2.1 - Extended PPT Analysis

- Optimize the new parameters of BVE

PPT for CDCL - CP3+glucose 2.1 - Extended PPT Analysis

- Optimize the new parameters of BVE

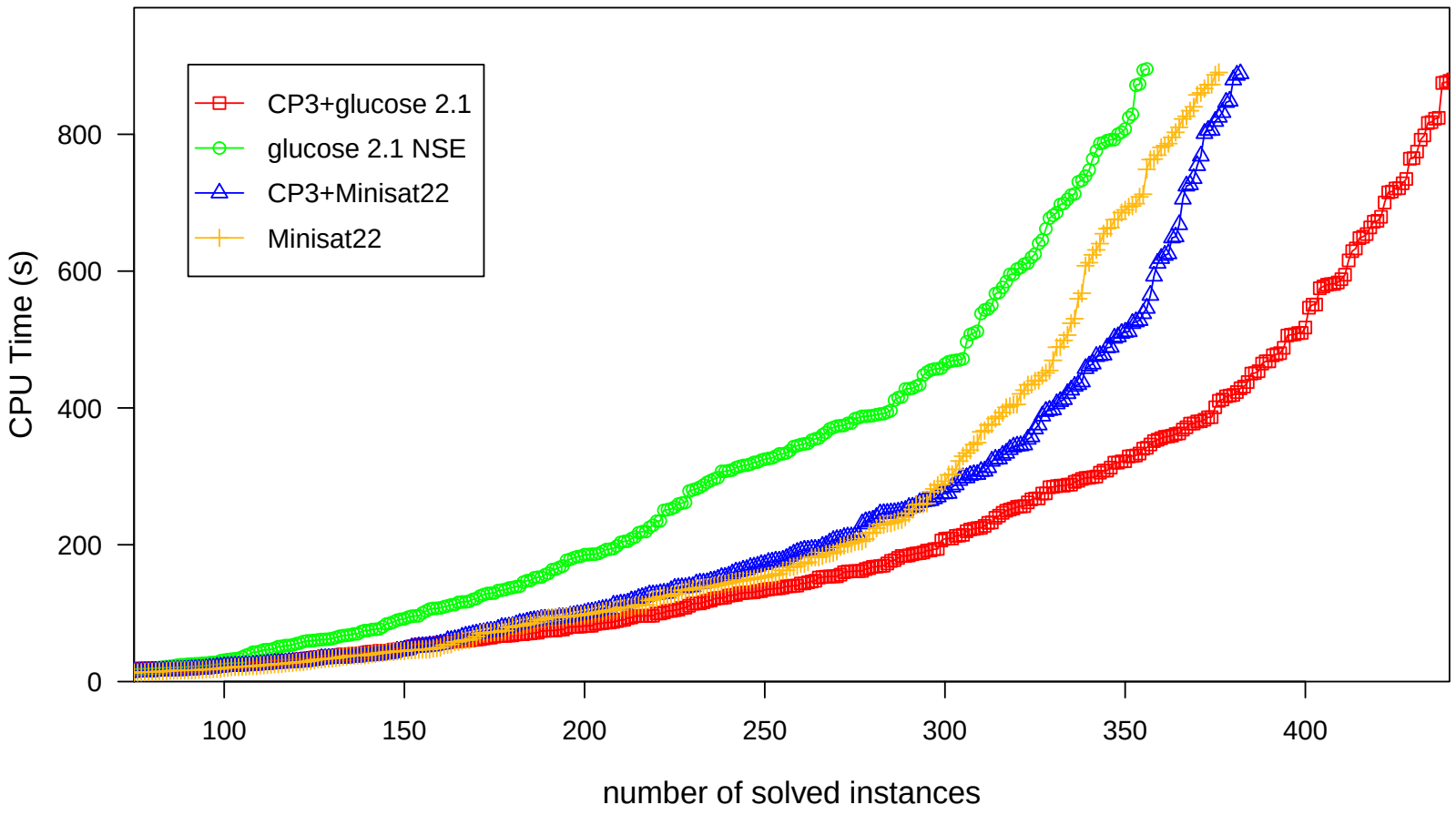


PPT for CDCL - Applicability to other Solvers

- Replace glucose 2.1 with Minisat 2.2

PPT for CDCL - Applicability to other Solvers

- Replace glucose 2.1 with Minisat 2.2



Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)

Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)
- BVE is the most important PPT $\rightarrow$ further analysis needed

Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)
- BVE is the most important PPT $\rightarrow$ further analysis needed
- If the solver can not be improved any more take a look at the PP

Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)
- BVE is the most important PPT $\rightarrow$ further analysis needed
- If the solver can not be improved any more take a look at the PP
- PPTs integrated also as inprocessing (IP) have high potential

Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)
- BVE is the most important PPT $\rightarrow$ further analysis needed
- If the solver can not be improved any more take a look at the PP
- PPTs integrated also as inprocessing (IP) have high potential
- The application order and the number of application of each PPT has not been optimized yet (search space explosion)

Conclusion and Future Work

- PPTs setting depend on the solver used (SLS/CDCL)
- BVE is the most important PPT $\rightarrow$ further analysis needed
- If the solver can not be improved any more take a look at the PP
- PPTs integrated also as inprocessing (IP) have high potential
- The application order and the number of application of each PPT has not been optimized yet (search space explosion)
- Optimizing the parameters of the solver and PP might yield even better improvements

This work at SAT Competition 2013

CP3+Sparrow in HC SAT

1. Runs CP3 with the HC-SLS configuration
2. Executes an improved version of Sparrow

SparrowToRiss in HC SAT+UNSAT

1. Runs CP3 with the HC-SLS configuration
2. Executes an improved version of Sparrow for $5 \cdot 10^8$ flips
3. Runs CP3 with the HC-CDCL configuration
4. Runs CDCL solver RISS (based on glucose 2.2 which incorporates CP3) initializing the phase savings with the last assignment found by Sparrow in chronological order (i.e. last flipped variable first)