Computing Diverse Optimal Stable Models

Javier Romero, Torsten Schaub, and Philipp Wanko

Institute for Computer Science
University of Potsdam, Germany

TC-ICLP 2016 New York

Introduction

- Answer Set Programming (ASP) is a paradigm for solving combinatorial problems
  $\Rightarrow$ in worst-case exponential number of solutions

Potassco

Introduction

- Answer Set Programming (ASP) is a paradigm for solving combinatorial problems
  $\Rightarrow$ in worst-case exponential number of solutions
- Impose preference relations among solutions to filter out optimal ones

Introduction

- Answer Set Programming (ASP) is a paradigm for solving combinatorial problems
  $\Rightarrow$ in worst-case exponential number of solutions
- Impose preference relations among solutions to filter out optimal ones
- Certain preference relations and applications exhibit vast numbers of optimal solutions
  eg multi-objective optimization problems in system synthesis, timetabling, configuration, planning

Introduction

- Answer Set Programming (ASP) is a paradigm for solving combinatorial problems
  $\Rightarrow$ in worst-case exponential number of solutions
- Impose preference relations among solutions to filter out optimal ones
- Certain preference relations and applications exhibit vast numbers of optimal solutions
  eg multi-objective optimization problems in system synthesis, timetabling, configuration, planning
  $\Rightarrow$ Further filtering necessary, eg diverse optimal stable models

Course timetabling

- Courses a, b with two lessons each
- One room
- Two days with four periods

Course timetabling

- Courses *a*, *b* with two lessons each
- One room
- Two days with four periods

- Assign courses to days and periods

Course timetabling

- Courses a, b with two lessons each
- One room
- Two days with four periods

- Assign courses to days and periods

Period	Day 1	Day 2
1	a	
2		a
3	b	b
4		

Potassco

Preferences

- Compactness: Avoid empty periods after a scheduled lesson

Preferences

- Compactness: Avoid empty periods after a scheduled lesson
- Spread: Avoid placing lessons of same course on same day

Preferences

- Compactness: Avoid empty periods after a scheduled lesson
- Spread: Avoid placing lessons of same course on same day

- Combine both preferences in a Pareto preference relation

Pareto preference

- Multi-objective optimization
- Aggregation of several preference relations
- Generally no single optimal solution
- Set of Pareto optimal solutions (Pareto frontier)
- eg "better" means $<$ for objectives in $\mathbb{Z}$:
  - $(2, 3, 1)$ dominates $(3, 3, 1)$
  - $(2, 3, 1)$ is incomparable to $(2, 1, 3)$
- x is Pareto optimal $\iff$ there is no y that dominates x

Dominated solution

Period	Day 1	Day 2
1	a	
2		a
3	b	b
4		

Compactness violations: 3
Spread violations: 0

Optimal solution

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Compactness violations: 2
Spread violations: 0

Optimal solution

Period	Day 1	Day 2
1		
2	a	b
3	b	a
4		

Compactness violations: 2
Spread violations: 0

Optimal solution

Period	Day 1	Day 2
1	b	
2	a	
3	b	
4	a	

Compactness violations: 1
Spread violations: 2

Diverse optimal solution

- Many optimal solutions
- Select subset to present to user

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		
2	a	b
3	b	a
4		

Period	Day 1	Day 2
1	b	
2	a	
3	b	
4	a	

Potassco

Diverse optimal solution

- Many optimal solutions
- Select subset to present to user

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		
2	a	b
3	b	a
4		

Period	Day 1	Day 2
1	b	
2	a	
3	b	
4	a	

Diverse optimal solution

- Many optimal solutions
- Select diverse subset to present to user

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3		a
4		b

Period	Day 1	Day 2
1	a	
2	b	
3	a	
4	b	

Overview

- Existing work:
  - T. Eiter, E. Erdem, H. Erdogan, and M. Fink. Finding similar/diverse solutions in answer set programming. 2013.
  - Y. Zhu and M. Truszczyński. On optimal solutions of answer set optimization problems. 2013.

Overview

- Existing work:
  - T. Eiter, E. Erdem, H. Erdogan, and M. Fink. Finding similar/diverse solutions in answer set programming. 2013.
  - Y. Zhu and M. Truszczyński. On optimal solutions of answer set optimization problems. 2013.
- Contributions:
  - Automate existing ASP solving schemes

Overview

- Existing work:
  - T. Eiter, E. Erdem, H. Erdogan, and M. Fink. Finding similar/diverse solutions in answer set programming. 2013.
  - Y. Zhu and M. Truszczyński. On optimal solutions of answer set optimization problems. 2013.
- Contributions:
  - Automate existing ASP solving schemes
  - Generalize existing diversification techniques to logic programs with preferences

Potassco

Overview

- Existing work:
  - T. Eiter, E. Erdem, H. Erdogan, and M. Fink. Finding similar/diverse solutions in answer set programming. 2013.
  - Y. Zhu and M. Truszczyński. On optimal solutions of answer set optimization problems. 2013.
- Contributions:
  - Automate existing ASP solving schemes
  - Generalize existing diversification techniques to logic programs with preferences
  - Introduce new diversification techniques for logic programs with preferences

Overview

- Existing work:
  - T. Eiter, E. Erdem, H. Erdogan, and M. Fink. Finding similar/diverse solutions in answer set programming. 2013.
  - Y. Zhu and M. Truszczyński. On optimal solutions of answer set optimization problems. 2013.
- Contributions:
  - Automate existing ASP solving schemes
  - Generalize existing diversification techniques to logic programs with preferences
  - Introduce new diversification techniques for logic programs with preferences

    $\Rightarrow$ Provide comprehensive framework for computing diverse (or similar) optimal solutions

Outline

ASP Solving Schemes

- Schemes are employed to implement diversification techniques but are readily available for independent usage

1. Maxmin
2. Guess and check
3. Querying
4. Preferences[2]

Outline

Diversification Techniques

1. Enumeration
2. Replication
3. Approximation

Diversification Techniques

1. Enumeration
2. Replication
3. Approximation
   - Iterate optimal solutions distant to a previous set of optimal solutions
   - Uses maxmin, querying and preferences[2]

Approximation

Input: Logic program with preferences P, set size $n \in \mathbb{N}$
Output: Set $\mathcal{X}$ approximating n most diverse optimal solutions

Approximation

Input: Logic program with preferences P, set size $n \in \mathbb{N}$
Output: Set $\mathcal{X}$ approximating n most diverse optimal solutions

$\mathcal{X} = \emptyset$
while $|\mathcal{X}| < n$ **and** $P \cup \mathcal{X}$ is satisfiable
  $\mathcal{X} := \mathcal{X} \cup solve(P, \mathcal{X})$
return $\mathcal{X}$

Approximation

Input: Logic program with preferences P, set size $n \in \mathbb{N}$
Output: Set $\mathcal{X}$ approximating n most diverse optimal solutions

$\mathcal{X} = \emptyset$
while $|\mathcal{X}| < n$ **and** $P \cup \mathcal{X}$ is satisfiable
  $\mathcal{X} := \mathcal{X} \cup \mathit{solve}(P, \mathcal{X})$
return $\mathcal{X}$

- All variants implement different versions of $\mathit{solve}(P, \mathcal{X})$
- With increasing variant number: less exact, more performance

Potassco

Algorithm I

- *solve*$(P, \mathcal{X})$ returns optimal model of P most dissimilar to $\mathcal{X}$

Algorithm I

- $solve(P, \mathcal{X})$ returns optimal model of P most dissimilar to $\mathcal{X}$
- Ensures next optimal solution is most distant to whole previous set $\mathcal{X}$
  by using maxmin and preference[2]

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$$\mathcal{X} =$$

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Potassco

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3		a
4		b

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3		a
4		b

Period	Day 1	Day 2
1		a
2	a	a,b
3	b	a,b
4		a

Potassco

Algorithm I example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3		a
4		b

Period	Day 1	Day 2
1	a	
2	b	
3	a	
4	b	

Algorithm II

- $\mathit{solve}(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$

Algorithm II

- $solve(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Then returns optimal model of P closest to I

Algorithm II

- *solve*$(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Then returns optimal model of P closest to I
- Ensures next optimal solution is as close as possible to I by using maxmin and preference[2]

Algorithm II

- *solve*$(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Then returns optimal model of P closest to I
- Ensures next optimal solution is as close as possible to I by using maxmin and preference[2]
- Less accuracy but more performance since there is no direct comparison to earlier optimal solutions

Potassco

Algorithm II: Distant partial interpretation I

Given set of previous optimal models $\mathcal{X}$, select I

1. randomly
2. heuristically chosen
3. most diverse wrt $\mathcal{X}$
4. complementary to last optimal solution

Algorithm II: Distant partial interpretation I

Given set of previous optimal models $\mathcal{X}$, select I

1. randomly
2. heuristically chosen
3. most diverse wrt $\mathcal{X}$
4. complementary to last optimal solution

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$$\mathcal{X} =$$

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1	a,b	a,b
2	b	b
3	a	a
4	a,b	a,b

Potassco

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3	a	
4	b	

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3	a	
4	b	

Period	Day 1	Day 2
1	a,b	b
2	a,b	a
3	b	a,b
4	a	a,b

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3	a	
4	b	

Period	Day 1	Day 2
1	a	
2	b	
3	b	
4	a	

Algorithm II: Example

Enumerate three most diverse optimal solutions for timetabling.

$\mathcal{X} =$

Period	Day 1	Day 2
1		
2	a	a
3	b	b
4		

Period	Day 1	Day 2
1		a
2		b
3	a	
4	b	

Period	Day 1	Day 2
1	a	
2	b	
3	b	
4	a	

Algorithm III

- *solve*$(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$

Algorithm III

- *solve*$(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Tries to find optimal model of P closest to I by changing sign heuristic to same values as in I

Algorithm III

- $solve(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Tries to find optimal model of P closest to I by changing sign heuristic to same values as in I
- Approximation of Algorithm II

Algorithm III

- *solve*$(P, \mathcal{X})$ computes partial interpretation I distant to $\mathcal{X}$
- Tries to find optimal model of P closest to I by changing sign heuristic to same values as in I
- Approximation of Algorithm II
- Purely heuristic, no guaranteed diversity; no optimization, less complex calculation

Algorithm III: Distant partial interpretation I

Given set of previous optimal models $\mathcal{X}$, select I

1. randomly
2. heuristically chosen
3. most diverse wrt $\mathcal{X}$
4. complementary to last optimal solution

Outline

1. Introduction and Motivation
2. ASP Solving Schemes
3. Diversification Techniques
4. Conclusion

Experiments

	III	I	III*
T/TO	**165s/70**	482s/672	317s/351
Rank	16	**1**	4

- 816 instances enumerating four Pareto optimal diverse models with 16 Approximation configurations
- Timeout of 600s

Potassco

Summary

- Provide comprehensive framework for computing diverse optimal stable models
- Five ASP solving schemes: maxmin, guess and check, querying, preference[2]
- Three diversification techniques: enumeration, replication, approximation
- Replication and enumeration inefficient
- Approximation techniques display tradeoff between diversification and performance

Thank You!