Hello DO!

An Overview of My Teaching and Research Activities

Mohamed Siala
http://homepages.laas.fr/msiala

17 avril 2019

INSA Toulouse, LAAS-CNRS

Disclaimer

Disclaimer

- This is not a technical presentation

Disclaimer

- This is not a technical presentation
- It took me really a long time to decide what to talk about

Disclaimer

- This is not a technical presentation
- It took me really a long time to decide what to talk about
  - choose one paper to present ?
  - give a lecture about CP/SAT ?
  - talk about my research project ?
  - focus on potential collaborations ?
  - ...

Curriculum Vitae

Education

Education

- 2005 - 2007 : **Classes préparatoires en Mathématiques et Physiques**, IPEIS, Tunisia

Education

- 2005 - 2007 : **Classes préparatoires en Mathématiques et Physiques**, IPEIS, Tunisia
- 2007 - 2010 : **Engineering Diploma in Computer Science**, ENSI, Tunisia

Education

- 2005 - 2007 : **Classes préparatoires en Mathématiques et Physiques**, IPEIS, Tunisia
- 2007 - 2010 : **Engineering Diploma in Computer Science**, ENSI, Tunisia
- 2010 - 2012 : **Master's in Artificial Intelligence and Decision**, ENSI, Tunisia

Education

- 2005 - 2007 : **Classes préparatoires en Mathématiques et Physiques**, IPEIS, Tunisia
- 2007 - 2010 : **Engineering Diploma in Computer Science**, ENSI, Tunisia
- 2010 - 2012 : **Master's in Artificial Intelligence and Decision**, ENSI, Tunisia
- Decembre 2011, Mai 2015 : **PhD in Computer Science**, INSA Toulouse, LAAS-CNRS, ROC.

Work Experience

Work Experience

- 2011-2014 : **PhD. Candidate**, LAAS-CNRS, Toulouse, France,
  Supervisors : Emmanuel Hebrard and Christian Artigues
  Funding : CNRS, Google, Région Midi-Pyrénées

Work Experience

- 2011-2014 : **PhD. Candidate**, LAAS-CNRS, Toulouse, France,
  Supervisors : Emmanuel Hebrard and Christian Artigues
  Funding : CNRS, Google, Région Midi-Pyrénées
- 2012-2014 : **Teaching assistant**, INSA Toulouse

Work Experience

- 2011-2014 : **PhD. Candidate**, LAAS-CNRS, Toulouse, France,
  Supervisors : Emmanuel Hebrard and Christian Artigues
  Funding : CNRS, Google, Région Midi-Pyrénées
- 2012-2014 : **Teaching assistant**, INSA Toulouse
- Janvier 2013 : **Visiting researcher**, NICTA, Sydney, Australia,
  Supervisors : Toby Walsh and Nina Narodytska

Work Experience

- 2011-2014 : **PhD. Candidate**, LAAS-CNRS, Toulouse, France,
  Supervisors : Emmanuel Hebrard and Christian Artigues
  Funding : CNRS, Google, Région Midi-Pyrénées
- 2012-2014 : **Teaching assistant**, INSA Toulouse
- Janvier 2013 : **Visiting researcher**, NICTA, Sydney, Australia,
  Supervisors : Toby Walsh and Nina Narodytska
- 03/2015 - 08/2018 : **Post-Doctoral Researcher**, University College Cork, Insight, Centre for Data Analytics, Cork, Ireland
  Supervisor : Barry O'Sullivan
  Funding : Science Foundation Ireland (85%), and United Technologies Research Center (15%)

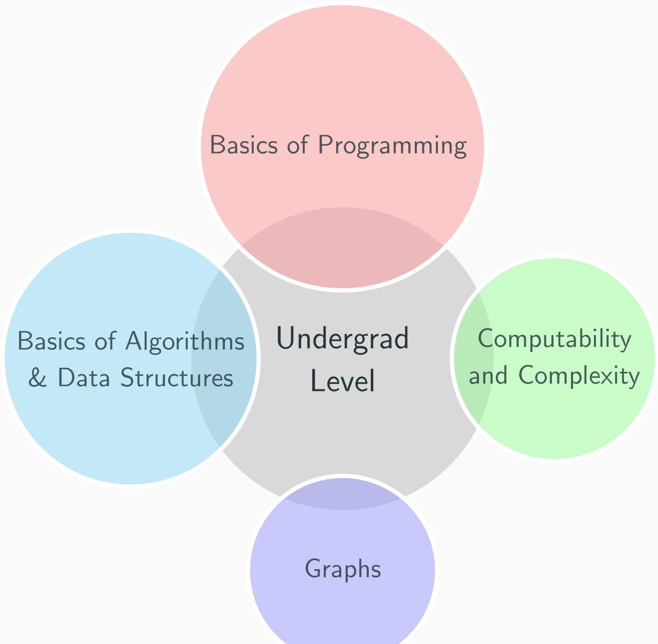
Work Experience

- 2011-2014 : **PhD. Candidate**, LAAS-CNRS, Toulouse, France,
  Supervisors : Emmanuel Hebrard and Christian Artigues
  Funding : CNRS, Google, Région Midi-Pyrénées
- 2012-2014 : **Teaching assistant**, INSA Toulouse
- Janvier 2013 : **Visiting researcher**, NICTA, Sydney, Australia,
  Supervisors : Toby Walsh and Nina Narodytska
- 03/2015 - 08/2018 : **Post-Doctoral Researcher**, University College Cork, Insight, Centre for Data Analytics, Cork, Ireland
  Supervisor : Barry O'Sullivan
  Funding : Science Foundation Ireland (85%), and United Technologies Research Center (15%)
- Since 08/2018 : **Associate Professor**, INSA Toulouse, LAAS-CNRS

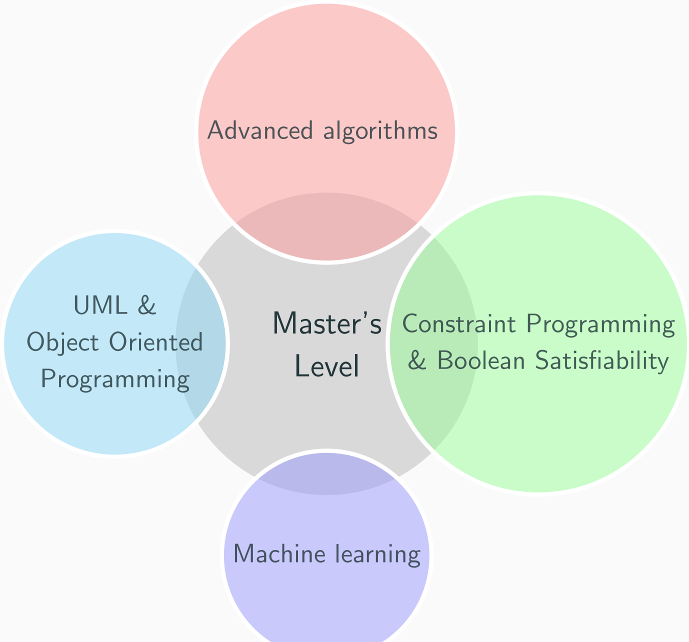
An overview of my Teaching activities

- INSA Toulouse : Engineering School (5 years),
- Department of Electrical and Computer Engineering
- Involved in teaching courses from 1st year to 5th year

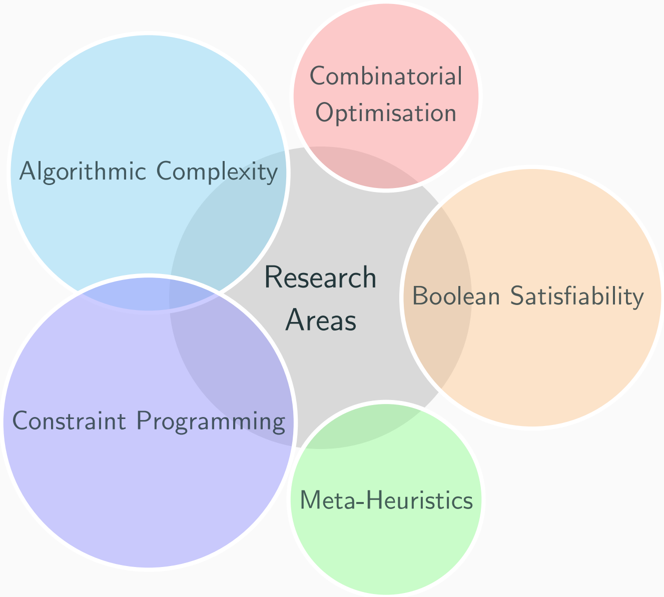
Undergrad Level (1st, 2nd, 3rd year)



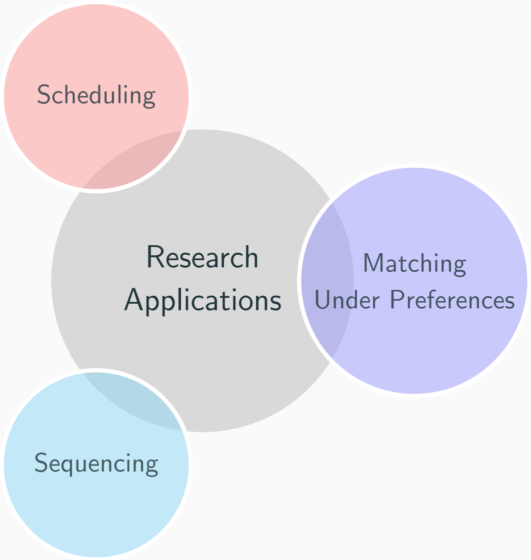
Master’s Level (4th and 5th year)



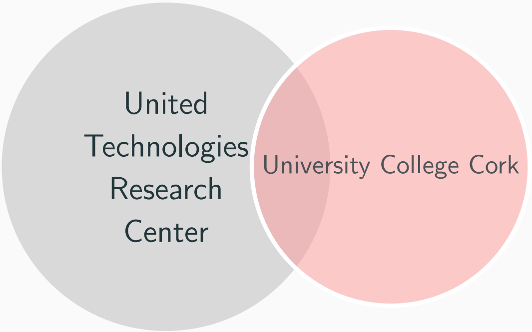
An overview of past research



An overview of past research applications



An overview of past industrial collaborations : UCC-UTRC



PhD Supervision

Begum Genc : Supervision with Barry O'Sullivan.
An Approach to Robustness in Stable Marriage and Stable Roommates Problems

PhD defence : May 3rd 2019

An Introduction to Constraint Programming

Context

Context

- Hard computational problems
- Combinatorial problems

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)
  $\ominus$ Not flexible

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)
  $\ominus$ Not flexible
- Specific modeling language (MIP/SAT)

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)
  $\ominus$ Not flexible
- Specific modeling language (MIP/SAT)
  $\ominus$ Modelisation matters

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)
  $\ominus$ Not flexible
- Specific modeling language (MIP/SAT)
  $\ominus$ Modelisation matters
- Metaheuristics

Context

- Hard computational problems
- Combinatorial problems

Solving Combinatorial Problems

- Adhoc methods (A solution working for only one specific problem)
  $\ominus$ Not flexible
- Specific modeling language (MIP/SAT)
  $\ominus$ Modelisation matters
- Metaheuristics
  $\oplus$ fast approximation
  $\ominus$ Do not guarantee optimal solution
  $\ominus$ Decision problems ?

Constraint programming

Constraint programming

- Declarative approach

Constraint programming

- Declarative approach
- The user states the problem, the computer finds a solution

Constraint programming

- Declarative approach
- The user states the problem, the computer finds a solution
- Clear separation between modeling and solving.

Formal Definitions

Formal Definitions

Constraint Network

- **Variables :** $x_1, x_2, \ldots$

Formal Definitions

Constraint Network

- **Variables :** $x_1, x_2, \ldots$
- **Domains :** $D(x_1) \subset \mathbb{Z},\ D(x_2) \subset \mathbb{Z}, \ldots$

Formal Definitions

Constraint Network

- **Variables :** x_1, x_2, ...
- **Domains :** $D(x_1) \subset \mathbb{Z}$, $D(x_2) \subset \mathbb{Z}$, ...
- **Constraints** : A constraint is a finite relation that is checkable in polynomial time.

Formal Definitions

Constraint Network

- **Variables :** x_1, x_2, ...
- **Domains :** $D(x_1) \subset \mathbb{Z}$, $D(x_2) \subset \mathbb{Z}$, ...
- **Constraints** : A constraint is a finite relation that is checkable in polynomial time.

Constraint Satisfaction Problem

Is there an assignment to the variables that satisfy all the constraints ?

Formal Definitions

Constraint Network

- **Variables :** x_1, x_2, ...
- **Domains :** $D(x_1) \subset \mathbb{Z}$, $D(x_2) \subset \mathbb{Z}$, ...
- **Constraints** : A constraint is a finite relation that is checkable in polynomial time.

Constraint Satisfaction Problem

Is there an assignment to the variables that satisfy all the constraints ?

Constraint Optimisation Problem

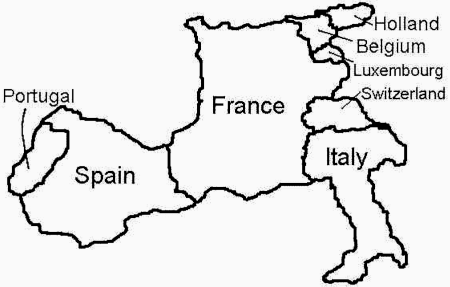
Is there an assignment to the variables that satisfy all the constraints *and optimise an objective function* ?

Important note

Observe that no restriction on the constraints is made. A constraint could be

- Logical formula
- Arithmetic
- Linear
- Non Linear
- Tabular
- A conjunction of other constraints

Example : Map coloring



Example : Map coloring

Model

- Variables :
  $Portugal, Spain, France, Italy, Switzerland, Luxembourg, Belgium, Holland$
- Domains : $\{Black, Blue, Green, Red, White, Yellow\}$
- Constraints :
  1. $Portugal \neq Spain$
  2. $Spain \neq France$
  3. $France \neq Italy$
  4. $France \neq Switzerland$
  5. $France \neq Luxembourg$
  6. $France \neq Belgium$
  7. $Belgium \neq Holland$
  8. $Belgium \neq Luxembourg$
  9. $Switzerland \neq Luxembourg$
  10. $Switzerland \neq Italy$

Philosophy of Constraint Programming

Philosophy of Constraint Programming

- In order to find solutions, eliminate as much possible non-solutions [**filtering**]

Philosophy of Constraint Programming

- In order to find solutions, eliminate as much possible non-solutions [**filtering**]
- Make *smart* decisions [**heuristic search**]

Philosophy of Constraint Programming

- In order to find solutions, eliminate as much possible non-solutions [**filtering**]
- Make *smart* decisions [**heuristic search**]
- Backtrack when you fail

Philosophy of Constraint Programming

- In order to find solutions, eliminate as much possible non-solutions [**filtering**]
- Make *smart* decisions [**heuristic search**]
- Backtrack when you fail

Example : Sudoku

8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

Filtering

7	2	5
3	1	8
?	4	9

Filtering

7	2	5
3	1	8
6	4	9

Filtering

7	?	5	
3	1	?	
?	4	9	

Filtering

7	$\{2, 6, 8\}$	5	
3	1	$\{2, 6, 8\}$	
$\{2, 6, 8\}$	4	9	

Filtering

7	$\{2, 6, 8\}$	5	$[4, 2, 1]$
3	1	$\{2, 6, 8\}$	
$\{2, 6, 8\}$	4	9	
	$[9, 6, 3]$	$[7, 3, 2]$	

Filtering

7	$\{\cancel{2},\cancel{6},8\}$	5	$[4, 2, 1]$
3	1	$\{2,6,8\}$	
$\{2,6,8\}$	4	9	
	$[9,6,3]$	$[7,3,2]$	

Filtering

7	$\{\cancel{2}, \cancel{6}, 8\}$	5	$[4, 2, 1]$
3	1	$\{2, 6, \cancel{8}\}$	
$\{2, 6, \cancel{8}\}$	4	9	
	$[9, 6, 3]$	$[7, 3, 2]$	

Filtering

7	$\{\cancel{2}, \cancel{6}, 8\}$	5	$[4, 2, 1]$
3	1	$\{\cancel{2}, 6, \cancel{8}\}$	
$\{2, 6, \cancel{8}\}$	4	9	
	$[9, 6, 3]$	$[7, 3, 2]$	

Filtering

7	$\{\cancel{2}, \cancel{6}, 8\}$	5	$[4, 2, 1]$
3	1	$\{\cancel{2}, 6, \cancel{8}\}$	
$\{2, \cancel{6}, \cancel{8}\}$	4	9	
	$[9, 6, 3]$	$[7, 3, 2]$	

Constraint Programming

Constraint Programming

- Filtering :
  - Every Constraint is associated with (at least) a filtering algorithm to make local inferences

Constraint Programming

- Filtering :
  - Every Constraint is associated with (at least) a filtering algorithm to make local inferences
  - A filtering algorithm takes the current domains as input and removes some inconsistent values according to the semantic of the constraint.

Constraint Programming

- Filtering :
    - Every Constraint is associated with (at least) a filtering algorithm to make local inferences
    - A filtering algorithm takes the current domains as input and removes some inconsistent values according to the semantic of the constraint.
- Decisions : Domain operations ($x = v$, $x \leq v$, ...)

Things We Care About Regarding Filtering

Things We Care About Regarding Filtering

- Algorithmic Complexity ? Ideally no worse than a quadratic worst case time

Things We Care About Regarding Filtering

- Algorithmic Complexity ? Ideally no worse than a quadratic worst case time
- The level of filtering ? this is is a well defined theoretical property that gives a characterisation of the filtering. For instance, a filtering algorithm maintains **"arc consistency"** iff for every value/variable pair $\langle v, x \rangle$ (where $v \in D(x)$) can be extended to a solution.

Things We Care About Regarding Filtering

- Algorithmic Complexity ? Ideally no worse than a quadratic worst case time
- The level of filtering ? this is is a well defined theoretical property that gives a characterisation of the filtering. For instance, a filtering algorithm maintains **"arc consistency"** iff for every value/variable pair $\langle v, x \rangle$ (where $v \in D(x)$) can be extended to a solution.

Think About Constraints Differently !

Think About Constraints Differently !

- A complex problem is defined as a conjunction of smaller problems (sub-problems)

Think About Constraints Differently !

- A complex problem is defined as a conjunction of smaller problems (sub-problems)
- Each sub-problem can be seen as a constraint

Think About Constraints Differently !

- A complex problem is defined as a conjunction of smaller problems (sub-problems)
- Each sub-problem can be seen as a constraint
- We have to come up with a filtering algorithm for each sub-problem

Think About Constraints Differently !

- A complex problem is defined as a conjunction of smaller problems (sub-problems)
- Each sub-problem can be seen as a constraint
- We have to come up with a filtering algorithm for each sub-problem
- We usually know how to solve such sub-problem individually. However, we need to reason about them from a filtering point of view.

Think About Constraints Differently !

- A complex problem is defined as a conjunction of smaller problems (sub-problems)
- Each sub-problem can be seen as a constraint
- We have to come up with a filtering algorithm for each sub-problem
- We usually know how to solve such sub-problem individually. However, we need to reason about them from a filtering point of view.
- Excellent strategy if the sub-problem is recurrent ! $\Longrightarrow$ we can use the same constraint in many problems

Example of a Research Paper : Rotation-Based Formulation for Stable Matching Mohamed Siala and Barry O'Sullivan, CP-'17, August 2017, Melbourne, Australia

Matching Under Preferences

Matching Under Preferences

- Assign residents to hospitals
- Every resident has a personnel preference over hospitals
- Each hospital has a preference list over residents

Matching Under Preferences

- Assign students to universities
- Every student has a personnel preference over universities
- Each university has a ranking list over students

Many-to-Many Stable Matching

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms
- Each agent has a preference list over agents of the other set

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms
- Each agent has a preference list over agents of the other set
- Each agent has a quota

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms
- Each agent has a preference list over agents of the other set
- Each agent has a quota
- A matching M is a set of acceptable pairs

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms
- Each agent has a preference list over agents of the other set
- Each agent has a quota
- A matching M is a set of acceptable pairs
- M is stable if the quota constraints are respected and no pair $\langle w, f \rangle$ has an incentive to deviate from M by being matched together

Many-to-Many Stable Matching

- $W = \{w_1, w_2, \ldots, w_{n_W}\}$: a set of workers
- $F = \{f_1, f_2, \ldots, f_{n_F}\}$: a set of firms
- Each agent has a preference list over agents of the other set
- Each agent has a quota
- A matching M is a set of acceptable pairs
- M is stable if the quota constraints are respected and no pair $\langle w, f\rangle$ has an incentive to deviate from M by being matched together

Example of Stability

w_1	$f_1\, f_2\, f_3$
w_2	$f_2\, f_3\, f_1$
w_3	$f_3\, f_1\, f_2$

f_1	$w_3\, w_2\, w_1$
f_2	$w_1\, w_2\, w_3$
f_3	$w_3\, w_1\, w_2$

Example of Stability

w_1	$f_1\, f_2\, f_3$
w_2	$f_2\, f_3\, f_1$
w_3	$f_3\, f_1\, f_2$

f_1	$w_3\, w_2\, w_1$
f_2	$w_1\, w_2\, w_3$
f_3	$w_3\, w_1\, w_2$

- $M_1 = \{\langle w_1, f_2\rangle, \langle w_2, f_3\rangle, \langle w_3, f_1\rangle\}$ is not stable because $\langle w_3, f_3\rangle$ is blocking M_1

Example of Stability

w_1	$f_1\, f_2\, f_3$
w_2	$f_2\, f_3\, f_1$
w_3	$f_3\, f_1\, f_2$

f_1	$w_3\, w_2\, w_1$
f_2	$w_1\, w_2\, w_3$
f_3	$w_3\, w_1\, w_2$

- $M_1 = \{\langle w_1, f_2\rangle, \langle w_2, f_3\rangle, \langle w_3, f_1\rangle\}$ is not stable because $\langle w_3, f_3\rangle$ is blocking M_1
- $M_2 = \{\langle w_1, f_2\rangle, \langle w_2, f_1\rangle, \langle w_3, f_3\rangle\}$ is stable

Hard Variants

- Many-2-Many stable matching is solvable in $O(L)$ time
- However, the problem becomes NP-Hard with side constraints

Hard Variants

- Many-2-Many stable matching is solvable in $O(L)$ time
- However, the problem becomes NP-Hard with side constraints

Sex-Equal & Balanced Stable Matching

Hard Variants

- Many-2-Many stable matching is solvable in $O(L)$ time
- However, the problem becomes NP-Hard with side constraints

Sex-Equal & Balanced Stable Matching

- Let M be a stable matching
  - C_M^w is the sum of the ranks of each worker's last partner
  - C_M^f is the sum of the ranks of each firm's last partner

Hard Variants

- Many-2-Many stable matching is solvable in $O(L)$ time
- However, the problem becomes NP-Hard with side constraints

Sex-Equal & Balanced Stable Matching

- Let M be a stable matching
  - C_M^w is the sum of the ranks of each worker's last partner
  - C_M^f is the sum of the ranks of each firm's last partner
- Balanced stable matching : find a stable matching M with the minimum value of $max\{C_M^w, C_M^f\}$

Hard Variants

- Many-2-Many stable matching is solvable in $O(L)$ time
- However, the problem becomes NP-Hard with side constraints

Sex-Equal & Balanced Stable Matching

- Let M be a stable matching
  - C_M^w is the sum of the ranks of each worker's last partner
  - C_M^f is the sum of the ranks of each firm's last partner
- Balanced stable matching : find a stable matching M with the minimum value of $max\{C_M^w, C_M^f\}$
- Sex-Equal Stable matching : find a stable matching M with the minimum value of $|C_M^m - C_M^w|$

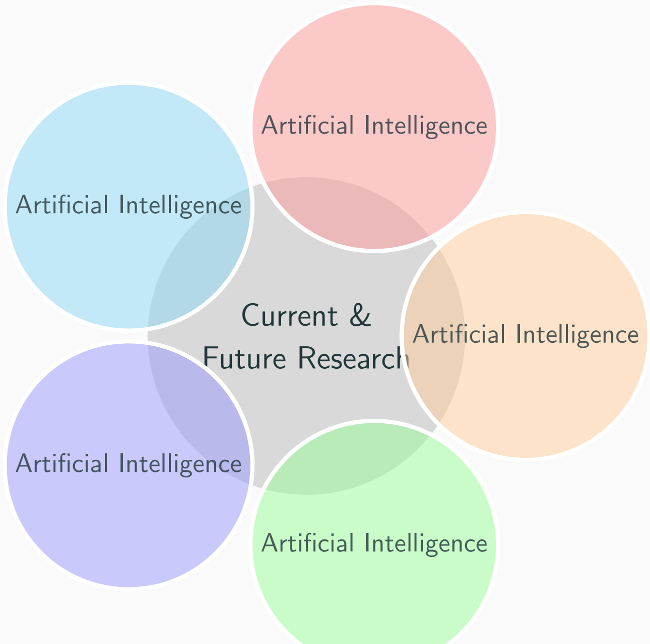
Stable Matching via CP

- Stability as a constraint
- Reformulation as a set of clauses
- We showed that arc consistency can be maintained in $O(L^2)$ worst case time complexity

Future Research and Potential Collaborations in DO

Current & Future Research ?

Current & Future Research ?



Current & Future Directions

Current & Future Directions

- More applications of CP (and SAT) in different areas

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT

Current & Future Directions

- More applications of CP (and SAT) in different areas
    - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
    - Internship of Hosseim NAHAL : PGMO project with EDF

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
  - Internship of Hosseim NAHAL : PGMO project with EDF
  - Summer Internship of Julien FERRY : Explainable Machine Learning

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
  - Internship of Hosseim NAHAL : PGMO project with EDF
  - Summer Internship of Julien FERRY : Explainable Machine Learning
- My future focus will be on machine learning (explainable and robust machine learning)

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
  - Internship of Hosseim NAHAL : PGMO project with EDF
  - Summer Internship of Julien FERRY : Explainable Machine Learning
- My future focus will be on machine learning (explainable and robust machine learning)
- Potential collaborations in DO :

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
  - Internship of Hosseim NAHAL : PGMO project with EDF
  - Summer Internship of Julien FERRY : Explainable Machine Learning
- My future focus will be on machine learning (explainable and robust machine learning)
- Potential collaborations in DO :
  - DISCO : SAT, MaxSAT, and CP for Diagnosis ?

Current & Future Directions

- More applications of CP (and SAT) in different areas
  - Internship of Hao HU : Reformulation of Machine Learning with SAT and MaxSAT
  - Internship of Hosseim NAHAL : PGMO project with EDF
  - Summer Internship of Julien FERRY : Explainable Machine Learning
- My future focus will be on machine learning (explainable and robust machine learning)
- Potential collaborations in DO :
  - DISCO : SAT, MaxSAT, and CP for Diagnosis ?
  - MAC : Applications of CP in continuous optimisation ?

Thank You !

Awards

- European Association for Artificial Intelligence Distinguished Dissertation Award 2015, honourable mention
- XCSP3 Solver Competition 2017 : First place in the optimisation track
- Honorable mention for the paper "An Optimal Arc Consistency Algorithm for a Chain of Atmost Constraints with Cardinality", CP2012 Conference