# DM877 (5 ECTS – 1st quarter)

## Constraint Programming

[Constraint Programmering]

Marco Chiarandini
lektor, IMADA
www.imada.sdu.dk/~marco

# Course Formalities

| | |
|---|---|
| Prerequisites: | ✔ Programming (Java or Python or Matlab)<br>✔ Algorithms and data structures (DM507) |
| Target: | students from comp sci, applied math, math and econ at 3rd (but challenging), 5th semester of Bachelor degree or at Master level |

# Constraint Programming

Constraint Programming is
a set of methods and tools for modelling and solving
constraint (optimisation or decision) problems.

# Constraint Programming

Constraint Programming is
a set of methods and tools for modelling and solving
constraint (optimisation or decision) problems.

Application areas:

- Configuration
- Design
- Logistics
- Planning
- Scheduling
- Transport
- Testing
- ...

# Constraint Programming

Constraint Programming is
a set of methods and tools for modelling and solving
constraint (optimisation or decision) problems.

Application areas:

- Configuration
- Design
- Logistics
- Planning
- Scheduling
- Transport
- Testing
- ...

Using techniques from:

- Artificial Intelligence
- Combinatorics
- Computational Logic
- Concurrent Computation
- Database Management
- Discrete Mathematics
- Operations Research
- Programming Languages

# Constraint Programming

Constraint Programming is
a set of methods and tools for modelling and solving
constraint (optimisation or decision) problems.

Application areas:

- Configuration
- Design
- Logistics
- Planning
- Scheduling
- Transport
- Testing
- ...

Using techniques from:

- Artificial Intelligence
- Combinatorics
- Computational Logic
- Concurrent Computation
- Database Management
- Discrete Mathematics
- Operations Research
- Programming Languages

# Constraint Programming

Constraint Programming is
a set of methods and tools for modelling and solving
constraint (optimisation or decision) problems.

Application areas:

- Configuration
- Design
- Logistics
- Planning
- Scheduling
- Transport
- Testing
- ...

Using techniques from:

- Artificial Intelligence
- Combinatorics
- Computational Logic
- Concurrent Computation
- Database Management
- Discrete Mathematics
- Operations Research
- Programming Languages

Constraint Programming = Representation (modeling) + Reasoning (search + inference)

# Characteristics of the Tasks (Problems)

- There are no dedicated algorithms
  - NP-completeness
  - Different strategies and heuristics have to be tested.

# Characteristics of the Tasks (Problems)

- There are no dedicated algorithms
  - NP-completeness
  - Different strategies and heuristics have to be tested.
- Requirements are quickly changing:
  - Programs should be flexible enough to adapt to these changes rapidly.

# Characteristics of the Tasks (Problems)

- There are no dedicated algorithms
  - NP-completeness
  - Different strategies and heuristics have to be tested.
- Requirements are quickly changing:
  - Programs should be flexible enough to adapt to these changes rapidly.
- Decision support required
  - Co-operate with user
  - Friendly interfaces

# Constraint Programming

## Modeling

# Constraint Programming

## Modeling

Modelling in MILP and SAT

Modelling in CP

# Problems with Constraints

Social Golfer Problem (Combinatorial Design)

# Problems with Constraints

## Social Golfer Problem (Combinatorial Design)

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | ? ? ? | ? ? ? | ? ? ? |
| **Day 1** | ? ? ? | ? ? ? | ? ? ? |
| **Day 2** | ? ? ? | ? ? ? | ? ? ? |
| **Day 3** | ? ? ? | ? ? ? | ? ? ? |

- 9 golfers: 1, 2, 3, 4, 5, 6, 7, 8, 9
- wish to play in groups of 3 players in 4 days
- such that no golfer plays in the same group with any other golfer more than just once.

Is it possible?

# Constraint Programming

## Modeling

# Constraint Programming

## Modeling

Golfers

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | ? ? ? | ? ? ? | ? ? ? |
| **Day 1** | ? ? ? | ? ? ? | ? ? ? |
| **Day 2** | ? ? ? | ? ? ? | ? ? ? |
| **Day 3** | ? ? ? | ? ? ? | ? ? ? |

# Constraint Programming

## Modeling

Golfers

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | ? ? ? | ? ? ? | ? ? ? |
| **Day 1** | ? ? ? | ? ? ? | ? ? ? |
| **Day 2** | ? ? ? | ? ? ? | ? ? ? |
| **Day 3** | ? ? ? | ? ? ? | ? ? ? |

Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

## Modeling

Golfers

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | ? ? ? | ? ? ? | ? ? ? |
| **Day 1** | ? ? ? | ? ? ? | ? ? ? |
| **Day 2** | ? ? ? | ? ? ? | ? ? ? |
| **Day 3** | ? ? ? | ? ? ? | ? ? ? |

Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |

**Integer variables:**

`assign[i,j]` variable whose value is from the domain $\{1, 2, 3\}$

# Constraint Programming

## Modeling

Golfers

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | ? ? ? | ? ? ? | ? ? ? |
| **Day 1** | ? ? ? | ? ? ? | ? ? ? |
| **Day 2** | ? ? ? | ? ? ? | ? ? ? |
| **Day 3** | ? ? ? | ? ? ? | ? ? ? |

Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |

**Integer variables:**

`assign[i,j]` variable whose value is from the domain $\{1, 2, 3\}$

**Constraints:**

C1: each group has exactly groupSize players

C2: each pair of players only meets once

# Constraint Programming

## Model with Integer Variables

```
int: golfers = 9;
int: groupSize = 3;
int: days = 4;
int: groups = golfers/groupSize;

set of int: Golfer = 1..golfers;
set of int: Day = 1..days;
set of int: Group = 1..groups;

array[Golfer, Day] of var Group: assign; % Variables
```

# Constraint Programming

## Model with Integer Variables

```
int: golfers = 9;
int: groupSize = 3;
int: days = 4;
int: groups = golfers/groupSize;

set of int: Golfer = 1..golfers;
set of int: Day = 1..days;
set of int: Group = 1..groups;

array[Golfer, Day] of var Group: assign; % Variables

solve :: int_search([assign[i,j] | i in Golfer, j in Day ],
                    first_fail, indomain_min, complete)  satisfy;

constraint
  % C1: Each group has exactly groupSize players
  forall (gr in Group, d in Day)( % c1
   sum (g in Golfer) (bool2int(assign[g,d] = gr)) = groupSize
  )
  /\
  % C2: Each pair of players only meets at most once
  forall (g1, g2 in Golfer, d1, d2 in Day  where g1 != g2 /\ d1 != d2) (
   (bool2int(assign[g1,d1] = assign[g2,d1]) + bool2int(assign[g1,d2] = assign[g2,d2])) <=1
  )
;
```

# Constraint Programming

## Solution: Assign and Propagate

### Golfers

| | Group 1 | Group 2 | Group 3 |
|---|---|---|---|
| **Day 0** | 0 1 2 | | |
| **Day 1** | | | |
| **Day 2** | | | |
| **Day 3** | | | |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | {2,3} | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

## Solution: Assign and Propagate

### Golfers

| | **Group 1** | **Group 2** | **Group 3** |
|---|---|---|---|
| **Day 0** | 0 1 2 | 3 4 5 | |
| **Day 1** | | | |
| **Day 2** | | | |
| **Day 3** | | | |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | {3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | {3} | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | {3} | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

## Solution: Assign and Propagate

### Golfers

| | **Group 1** | **Group 2** | **Group 3** |
|---|---|---|---|
| **Day 0** | 0 1 2 | 3 4 5 | 6 7 8 |
| **Day 1** | | | |
| **Day 2** | | | |
| **Day 3** | | | |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

## Solution: Assign and Propagate

### Golfers

| | **Group 1** | **Group 2** | **Group 3** |
|---|---|---|---|
| **Day 0** | 0 1 2 | 3 4 5 | 6 7 8 |
| **Day 1** | 0 | | |
| **Day 2** | | | |
| **Day 3** | | | |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | 1 | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | {2,3} | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {2,3} | {1,2,3} | {1,2,3} |
| Golfer 3 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

## Solution: Assign and Propagate

### Golfers

| | **Group 1** | **Group 2** | **Group 3** |
|---|---|---|---|
| **Day 0** | 0 1 2 | 3 4 5 | 6 7 8 |
| **Day 1** | 0 | 1 | |
| **Day 2** | | | |
| **Day 3** | | | |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | 1 | {1,2,3} | {1,2,3} |
| Golfer 1 | 1 | 2 | {1,2,3} | {1,2,3} |
| Golfer 2 | 1 | {3} | {1,2,3} | {1,2,3} |
| Golfer 3 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 4 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 5 | 2 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 6 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 7 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |
| Golfer 8 | 3 | {1,2,3} | {1,2,3} | {1,2,3} |

# Constraint Programming

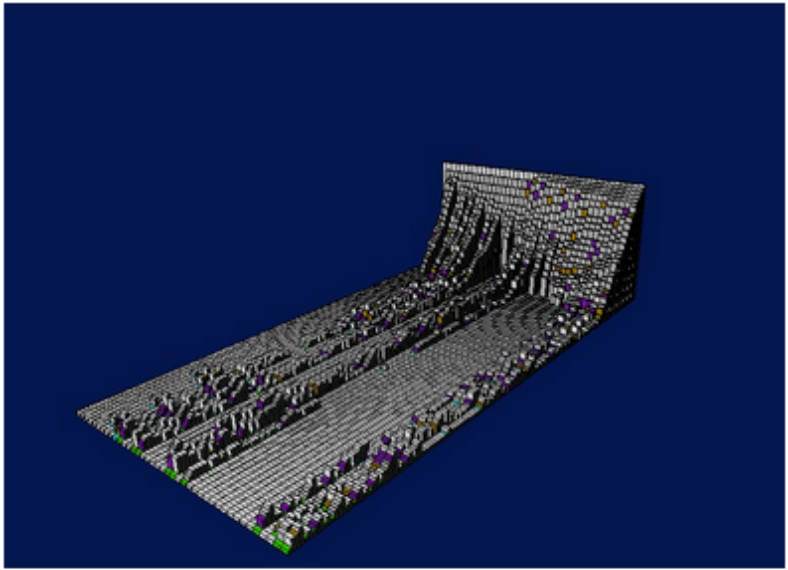
## Solution: Assign and Propagate

### Golfers

| | **Group 1** | **Group 2** | **Group 3** |
|---|---|---|---|
| **Day 0** | 0 1 2 | 3 4 5 | 6 7 8 |
| **Day 1** | 0 3 6 | 1 4 7 | 2 5 8 |
| **Day 2** | 0 4 8 | 1 5 6 | 2 3 7 |
| **Day 3** | 0 5 7 | 1 3 8 | 2 4 6 |

### Groups

| | Day 0 | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Golfer 0 | 1 | 1 | | |
| Golfer 1 | 1 | 2 | | |
| Golfer 2 | 1 | {3} | | |
| Golfer 3 | 2 | | | |
| Golfer 4 | 2 | | | |
| Golfer 5 | 2 | | | |
| Golfer 6 | 3 | | | |
| Golfer 7 | 3 | | | |
| Golfer 8 | 3 | | | |



Effect of constraint propagation on the domains
of variables during search in a placement problem.

# Contents: Constraint Programming

- Modelling and Applications
  Integer variables, set variables, float variables, constraints
- Principles
  Consistency levels
- Filtering Algorithms
  Alldifferent, cardinality, regular expressions, etc.
- Search:
  Backtracking, Strategies
- Symmetry Breaking
- Restart Techniques
- CP Systems: Minizinc

# Aims & Contents

- model problems with constraint programming
- implement the models in a CP system
- assess the programs
- describe with appropriate language

# Assessment (5 ECTS)

Three obligatory assignments (last one project based):

- individual
- deliverables: program + short written report
- graded with internal censor

# DM877 (5 ECTS – 1st quarter)

## Constraint Programming

[Constraint Programmering]

Marco Chiarandini
lektor, IMADA
www.imada.sdu.dk/~marco