

# Chapter 1: Introduction

Helmut Simonis

Cork Constraint Computation Centre  
Computer Science Department  
University College Cork  
Ireland

ECLiPSe ELearning [Overview](#)



## Licence

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License.

To view a copy of this license, visit [http:](http://creativecommons.org/licenses/by-nc-sa/3.0/)

[//creativecommons.org/licenses/by-nc-sa/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/) or

send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



# Outline

- 1 Constraint Programming
- 2 Chapter Overview
- 3 Chapter Details



# What we want to introduce

- Constraint Programming
- Using ECLIPSe Language
- With Saros Eclipse IDE



# Constraint Programming (CP)

- Solve hard combinatorial problems
- With minimal programming effort
- Exploit strategies and heuristics
- Understand and control problem solving



# ECLiPSe Language

- Open source constraint programming language
- Flexible toolkit to develop/use constraints
- Contains different constraint solvers
- Here: Use of finite domains/(mixed) integer programming



## Aims and Outcomes

- Understand what constraint programming is
- How constraint programs can be applied to a problem
- Which application problems are good candidates for CP
- How to write/run/analyze simple ECLiPSe programs



## You should already know about...

- No hard requirements
- Basic understanding of programming assumed
- Useful to have some background in one of:
  - Network Management
  - Integer Programming
  - Combinatorial Optimization



# Choices of materials

**Slides** PDF files for computer viewing

- Contains animations of visualization
- Large file sizes

**Handout** PDF files for printing

- 2 slides per page
- Does not contain all animations

**Transcript** Text of presentation as articles

**Video** Video presentation with audio (640x480 pixels)

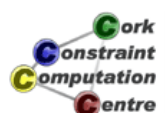
**iPhone** Video presentation tuned for iPhone display (480x320 pixels)

**Wiki** Hyperlink to wiki page on this chapter to add comments, corrections and suggestions



# Chapters

Introduction ( <b>You are here</b> )	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
First Steps - Hello World	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Application Overview	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Basic Constraint Reasoning	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Global Constraints	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Search Strategies	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Optimization	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Symmetry Breaking	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Choosing the Model	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Customizing Search	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Limits of Propagation	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Systematic Development	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Visualization Techniques	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Finite Set and Continuous Variables	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
Network Applications	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>
More Global Constraints	<a href="#">Video</a>	<a href="#">iPhone</a>	<a href="#">Slides</a>	<a href="#">Handout</a>	<a href="#">Wiki</a>



## More Chapters

Using Mixed Integer Linear Programming  
    A Hybrid Model  
    Comparing Technologies  
Working with Implications  
    Adding Material  
    Lessons Learned

[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)



## Applications

Application Overview  
SEND+MORE=MONEY  
    Sudoku  
    N-Queens  
Routing and Wavelength Assignment  
    RWA - Demand Acceptance 1  
    RWA - Demand Acceptance 2  
    RWA - Static Design 2  
Balanced Incomplete Block Designs  
    Sports Scheduling  
    Progressive Party  
    Costas Array  
SONET/SDH Ring Design  
    Network Applications  
    Car Sequencing  
    Shikaku

[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)  
[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)



# Introduction

- Aims and Outcomes
- Overview of chapters
- Hyperlinks to all materials

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# First Steps - Hello World

- How to install ECLiPSe and Saros
- Writing a first program
- Running the program
- Where to find information

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Application Overview

- Why constraint programming is interesting
- Solving industrial problems with CP
- Main application areas
  - Assignment
  - Scheduling
  - Network problems
  - Transportation
  - Personnel Assignment

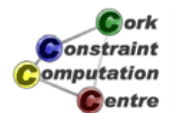
[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Basic Constraint Reasoning - SEND+MORE = MONEY

- Finite Domain variables
- CP: Variables + Constraints + Search
- Bounds reasoning on arithmetic constraints
- Simple visualizers

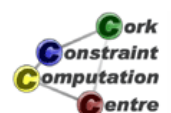
[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)





## Global Constraints - Sudoku

- Modelling the Sudoku puzzle
- One model, different behaviours
- Global constraint: `alldifferent`
- Bounds and domain consistency
- A domain consistent `alldifferent`

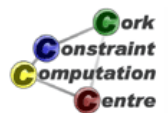
[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Search Strategies - N Queens

- How to search for a solution
- Variable and value choice
- How to avoid deep backtracking
- Partial search strategies

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Optimization - Routing and Wavelength Assignment

- Optimization
- Graph algorithms library
- Integer Programming with `eplex`
- Problem decomposition
- Routing and Wavelength Assignment in Optical Networks

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Symmetry Breaking - Balanced Incomplete Block Designs

- Balanced Incomplete Block Designs
- Planning Experiments and Testing Features
- Problems with highly symmetrical structure
- Symmetry Breaking with `lex` constraints

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Choosing the Model - Sports Scheduling

- Complex sports scheduling problem
- How to decide which model to use
- Improving reasoning by channeling

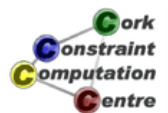
[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Customizing Search - Progressive Party

- Scheduling Meetings between Teams
- Teams only meet once
- Capacity Limits
- Build customized search routines tailored to problem
- Problem decomposition: decide which problem to solve

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Limits of Propagation - Costas Array

- Antenna/Sonar Design
- Hard Benchmark Problem
- Naive Enumeration works best
- When clever reasoning doesn't pay off
- Cautionary Tale

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Systematic Development

- Developing Programs
- Testing
- Profiling
- Documentation

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Visualization Techniques

- How to visualize constraint programs
- Variable Visualizers
- Understanding Search Trees
- Constraint Visualizers
- Complex Visualizations

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Finite Set and Continuous Variables - SONET Design Problem

- Finite set variables
- Continuous domains
- Optimization from below
- Advanced symmetry breaking
- SONET design problem without inter-ring traffic

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Network Applications

- Overview of Network Applications
- Traffic Placement
- Capacity Management
- Network Design
- Using Advanced Techniques

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# More Global Constraints - Car Sequencing

- New global constraints: `gcc` and `sequence`
- Choosing a better search strategy

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



# Using Mixed Integer Linear Programming - RWA Demand Acceptance 1

- Mixed Integer Linear Programming in ECLiPSe
- `eplex` Library
- Alternative Models for Routing and Wavelength Assignment in Optical Networks

[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)



# A Hybrid Model - RWA Demand Acceptance 2

- Hybridisation by decomposition
- Combination of MIP and FD solver
- Best current solution to routing and wavelength assignment problem

[Video](#) [iPhone](#) [Slides](#) [Handout](#) [Wiki](#)



## Comparing Technologies

- Compare static design and demand acceptance versions of RWA
- See impact of objective function
- Compare finite domain, MIP and SAT solutions

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Working with Implications - Shikaku

- Solving a placement problem without specialized constraints
- Decomposition into pattern generation and set partitioning
- Using implications to propagate information

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)





## Adding Material

- How to add new chapters
- Copying template files
- Configuring templates
- Adding frames to body
- Integrating with other chapters

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## Lessons Learned

- New ELearning course for ECLiPSe
- Modelling and programming with constraints
- Based on sample problems solved and explained in detail
- *A view* on core constraint programming skills
- Strong dependence on visualization to explain behavior

[Video](#)

[iPhone](#)

[Slides](#)

[Handout](#)

[Wiki](#)



## To continue

- Branch from here to all materials
- Choose presentation form which suits you

