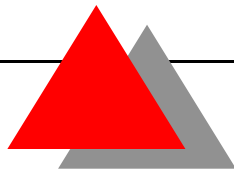


# **A Problem Classification Scheme - When to use CLP**

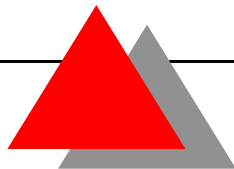
**Helmut Simonis  
COSYTEC SA**





## Overview

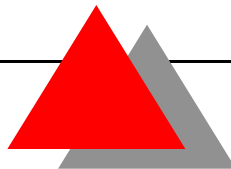
- ◆ **Part 1**
  - What is Constraint Logic Programming (CLP) ?
  - Competing techniques
- ◆ **Part 2**
  - Problem classification scheme
- ◆ **Part 3**
  - Problem failures
  - Modeling checklist
  - Comparison studies
  - What really works

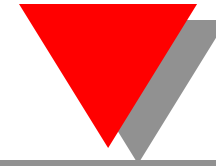




## What we discuss

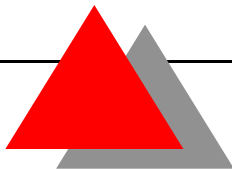
- ◆ **Overview of application domains**
- ◆ **Comprehensive study of industrial usage**
  - large scale applications
  - prototypes, studies
- ◆ **Common characteristics of applications**
- ◆ **Reasons for success/failure**





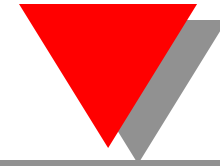
## What we don't discuss

- ◆ How to solve problems
- ◆ Cost/benefit studies
- ◆ Software engineering studies
- ◆ Integration issues
- ◆ Tool selection



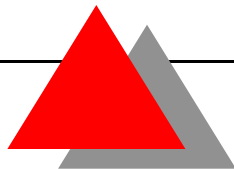
## Approach

- ◆ **Look what has been done**
  - overview of applications and studies using CLP
  - centered on COSYTEC/ECRC experience
  - results from competing systems may not be up to date
- ◆ **Find common points**
  - group applications according to application characteristics
- ◆ **Try to explain effects**
  - based on CHIP; results with other tools may be different
- ◆ **Group success/failure**
  - personal view
  - new techniques/studies may change this

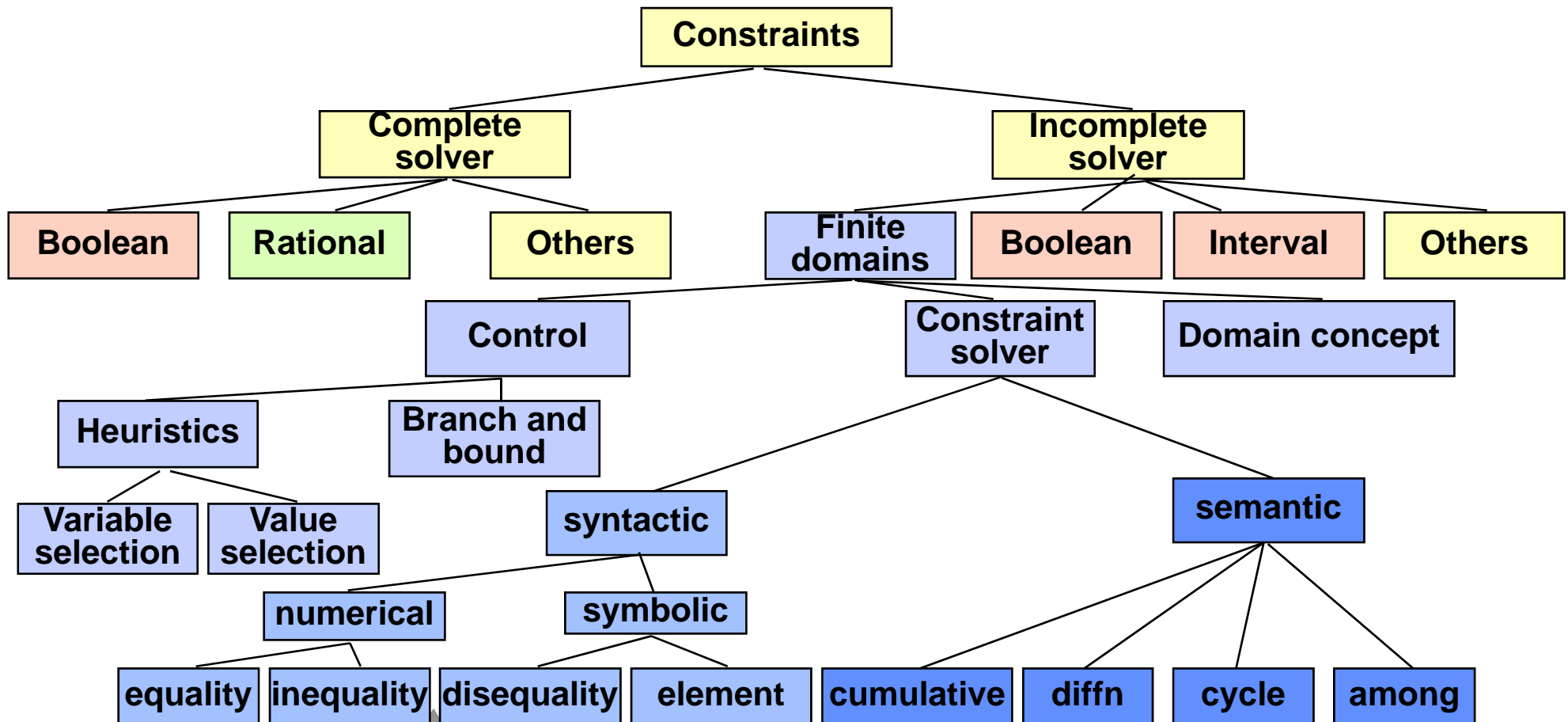


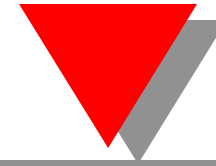
## Part 1

### Constraint logic programming



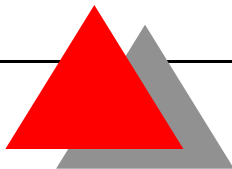
## Constraint system structure





## Computation domains

- ◆ **Finite domains**
- ◆ **Linear arithmetic**
- ◆ **Boolean**
- ◆ **Intervals**
- ◆ **Pseudo Boolean**
- ◆ **Non-linear arithmetic**
- ◆ **Sets, sequences, lists**



## Incomplete finite domain solver

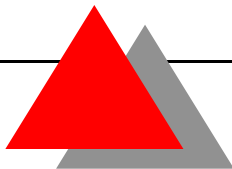
- ◆ **Domain**
  - finite sets of values
  - subsets of natural numbers
- ◆ **Need for enumeration**
- ◆ **Classification criteria**
  - constraint granularity
  - richness of constraint sets
  - propagation results
  - user definable constraints/control
- ◆ **Methods**
  - explicit domain representation
  - bound propagation/ removal of interior values
  - heuristics based on domains



## Domain concept

- ◆ **Domain variables**
  - subset of natural numbers
- ◆ **Variable assignment**
  - **indomain**, non-deterministic instantiation
- ◆ **Search and backtracking**

```
labeling([]).  
labeling([H|T):-  
    indomain(H),  
    labeling(T).
```



## Syntactic propagation methods

### ◆ Forward checking

- wait until only one variable is left in constraint
- remove inconsistent values

$$X \neq Y$$

### ◆ Lookahead

- for each variable and each value check if consistent values exist
- remove inconsistent values
- repeat whenever constraints change

### ◆ Partial lookahead

- bound propagation

$$X > Y$$

## Bound propagation: Linear terms

$$a_1X_1 + a_2X_2 + \dots + a_nX_n + c_1 = b_1Y_1 + b_2Y_2 + \dots + b_mY_m + d_1$$

$$a_1X_1 + a_2X_2 + \dots + a_nX_n + c_1 \in [\min_l, \max_l]$$

$$b_1Y_1 + b_2Y_2 + \dots + b_mY_m + d_1 \in [\min_r, \max_r]$$

$$\min = \max(\min_l, \min_r)$$

$$\max = \min(\max_l, \max_r)$$

$$[X, Y, Z]:: 1..10,$$
$$2*X + 3*Y + 3 \neq Z$$

$$X \text{ in } \{1,2\}$$

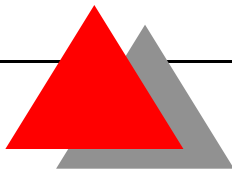
$$Y = 1$$

$$Z \text{ in } \{8,9,10\}$$



## Symbolic constraints

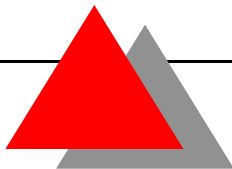
- ◆ **Disequality**
  - X is different from Y
- ◆ **Alldifferent**
  - all elements in list are pair-wise different
  - more compact than single disequality, but no better propagation
  - pigeon hole principle -> need for stronger propagation
- ◆ **Element**
  - element(X,L,Y)
    - ◆ Y is the Xth element of the list L
    - ◆ works in both directions



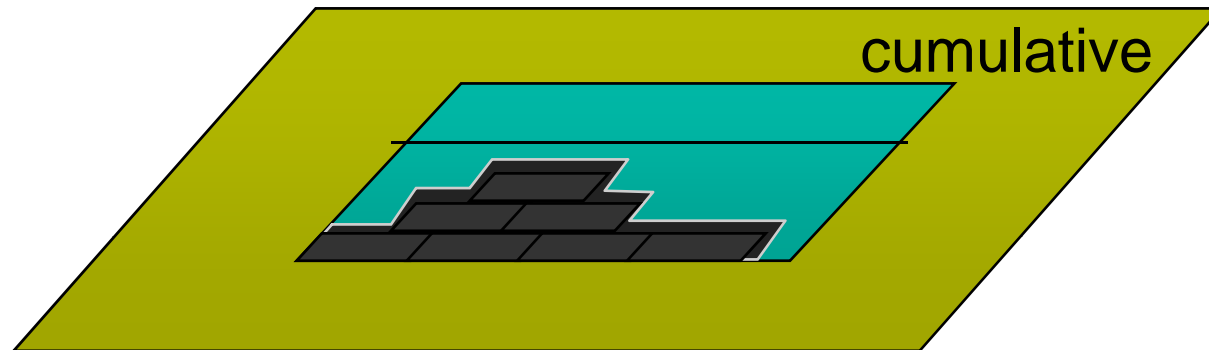


## Global constraints

- ◆ **Work on sets of variables**
  - global conditions, not local constraints
- ◆ **Semantic methods**
  - Operations Research
  - spatial algorithms
  - graph theory
  - network flows
- ◆ **Building blocks (high-level constraint primitives)**
  - as general as possible
  - multi-purpose
  - very strong propagation (within acceptable algorithmic complexity)



## The Cumulative global constraint



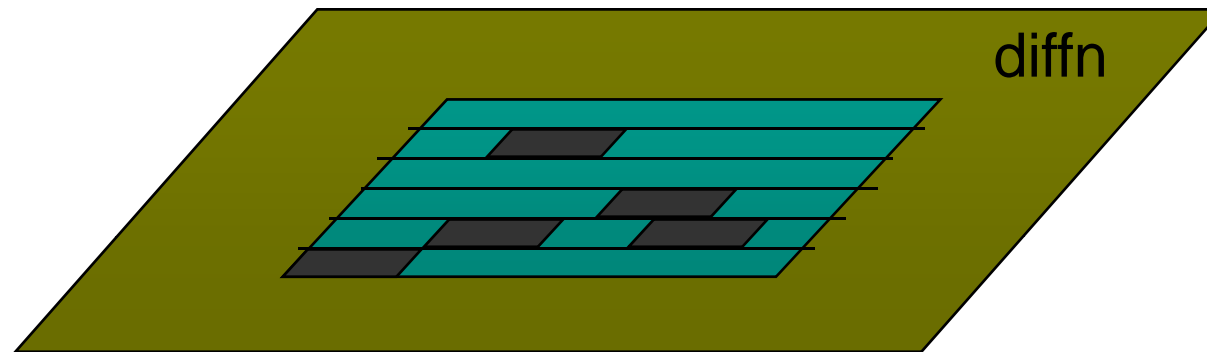
### ◆ Cumulative constraint

- Resource limits over periods of time
- Upper/lower limits
- Soft/hard limits
- Gradual constraint relaxation

### ◆ Application

- Resource restrictive scheduling, producer consumer constraints, disjunctive schedule, manpower constraints, overtime

## The Diffn global constraint



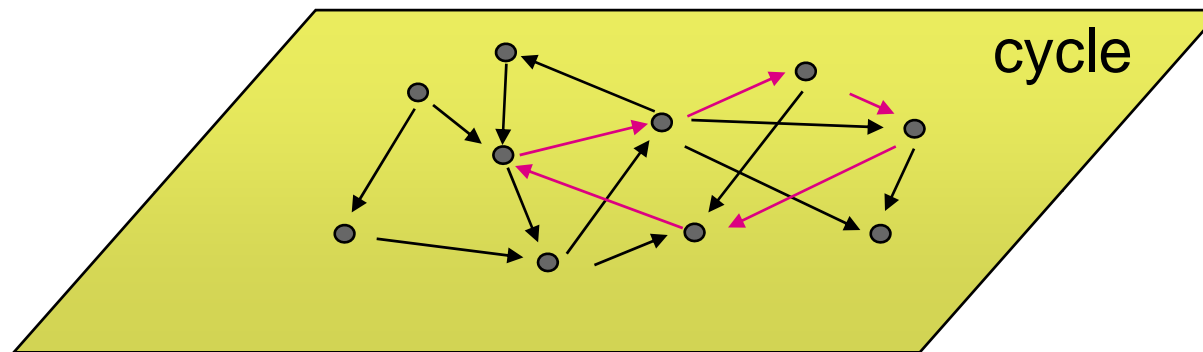
### ◆ Diffn constraint

- non overlapping areas on n-dimensional rectangles
- distances between rectangles
- limit use of areas

### ◆ Application

- layout, packing, resource assignment, setup, distribution planning, time-tabling

## The Cycle global constraint



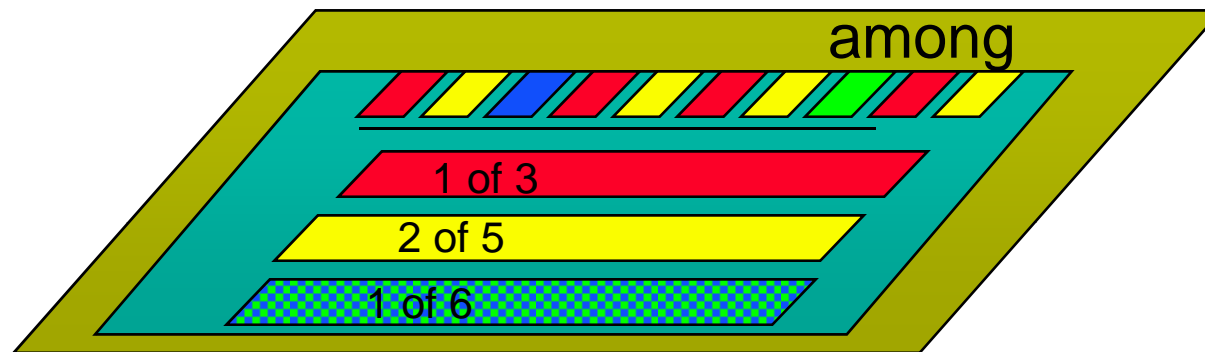
### ◆ Cycle constraint

- Finds cycles in directed graphs with minimal cost
- Assign resources, find compatible start dates

### ◆ Applications

- Tour planning, personnel rotation, distribution problems, production sequencing

## The Among global constraint



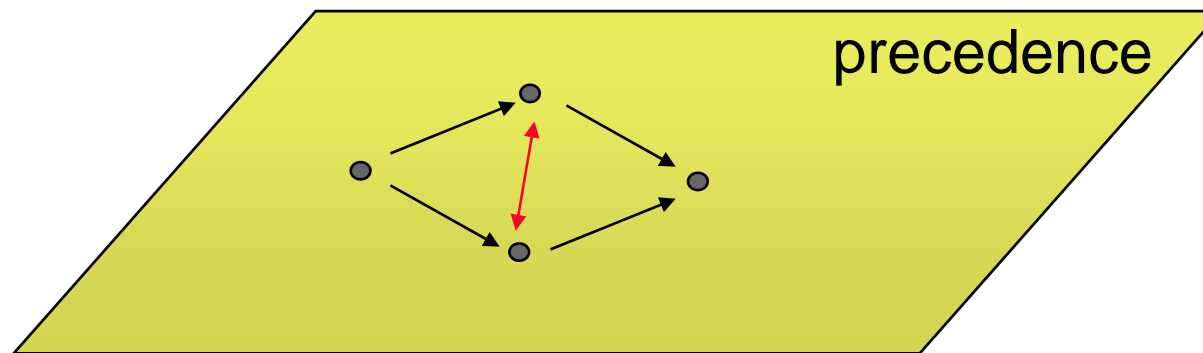
### ◆ Among constraint

- How often do values occur in (sub)sequences
- based on counting arguments
- interaction between sequences

### ◆ Applications

- production sequencing, time tabling, coloring problems, set covering

## The Precedence global constraint



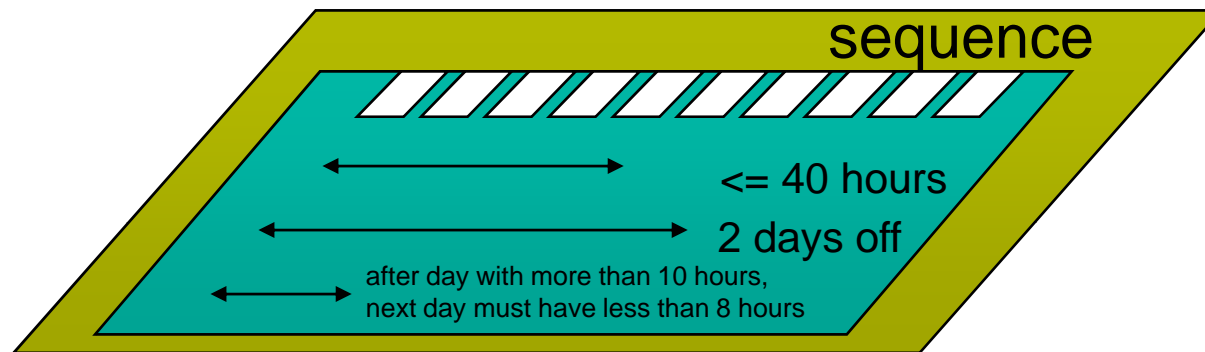
### ◆ Precedence constraint

- Combine resource constraints and precedence networks
- Reasoning on latency (position in network)
- Co-operation between multiple resources

### ◆ Applications

- resource restricted scheduling, channel routing, frequency allocation

## The Sequence global constraint



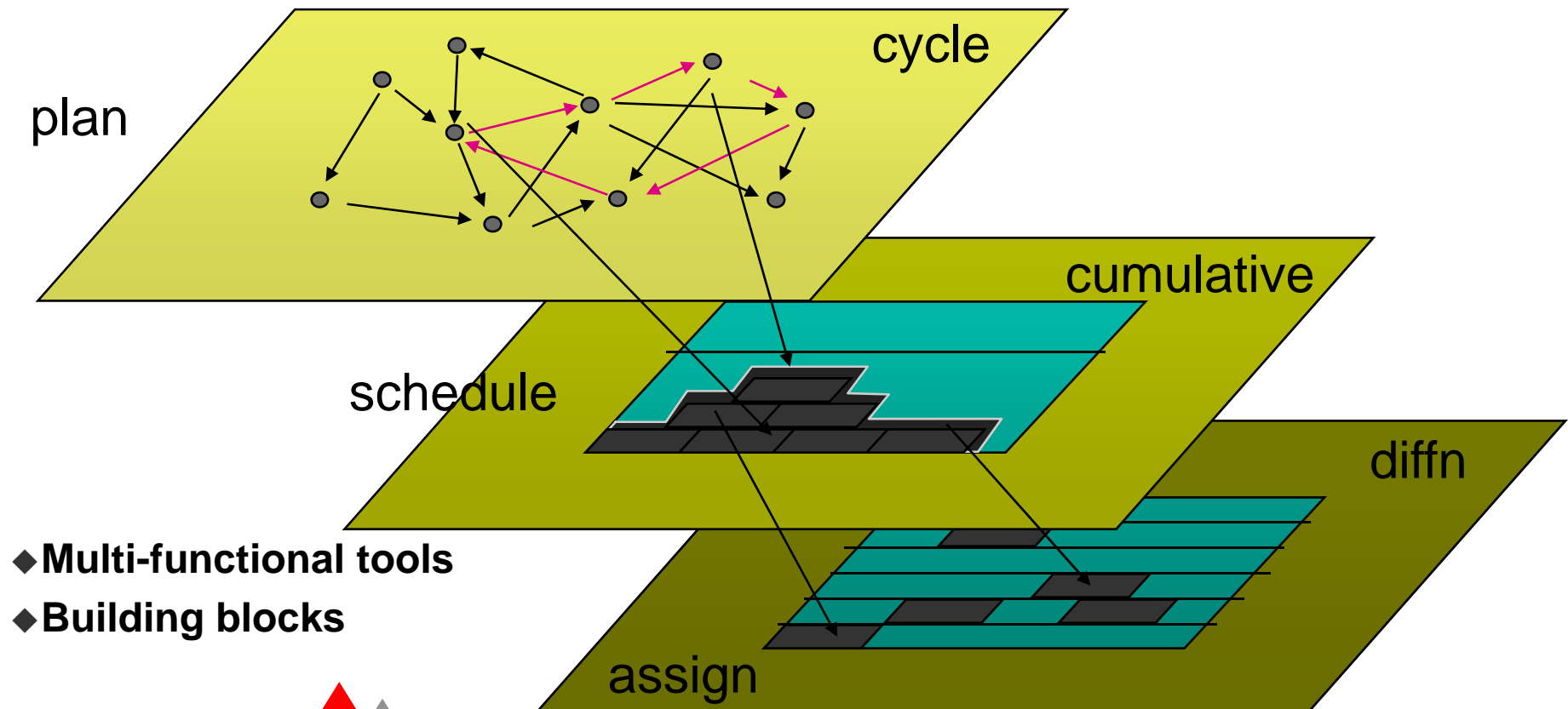
### ◆ Sequence constraint

- constraints on pattern inside sequences
- combinatorial pattern matching
- counting arguments

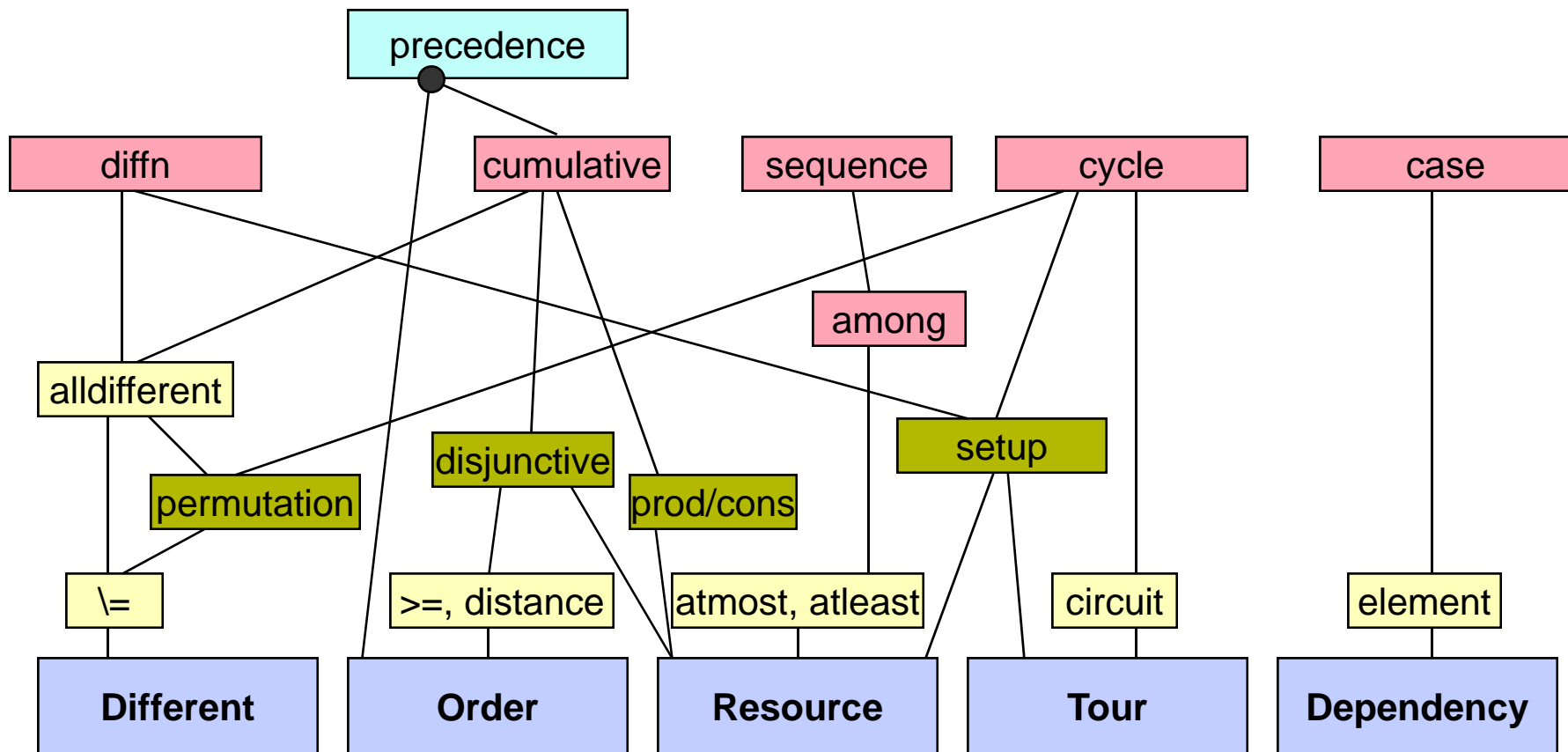
### ◆ Applications

- Time tabling, personnel assignment,
- work rules, scheduling with daily working time limits

## The power of global constraints



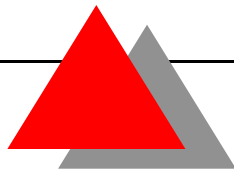
## Constraint morphology





## Competing techniques

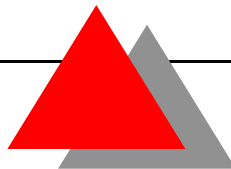
- ◆ **Linear/integer programming**
- ◆ **Heuristic algorithms**
- ◆ **Decomposition methods**
- ◆ **Neighborhood search**
  - simulated annealing
  - tabu search
  - genetic algorithms
- ◆ **Relaxation methods**





## Linear/integer programming

- ◆ **Express problems with linear equalities/inequalities**
- ◆ **Additions required to handle**
  - disequality
  - disjunction
- ◆ **Solve constraint systems with Simplex method**
  - other methods exist
  - very well developed tool kits
- ◆ **Some systems include simple modelling languages**
  - generate model from data
- ◆ **Search for integer solutions**
  - cutting planes
  - branch and bound techniques



## LP/MIP (2)

### ◆ Advantages

- highly developed mathematical theory
- good tools
- large knowledge base

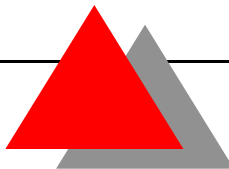
### ◆ Disadvantages

- restriction in modeling
  - ◆ types of constraints
  - ◆ types of variables
- programming with constraints
  - ◆ incremental
  - ◆ meta programming/explanations
- some problem types do not give good results
  - ◆ scheduling



## Heuristic algorithms

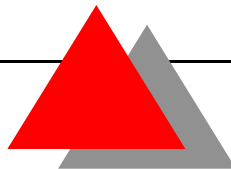
- ◆ **Progressive building solutions by adding pieces one at a time**
- ◆ **Items added chosen by heuristics**
- ◆ **Good solutions for weakly constrained problems**
- ◆ **Bad results for strongly constrained problems**
  - finding admissible solutions
- ◆ **Heuristics should take constraints into account**
  - dynamic, not static ordering required
- ◆ **Systems can be very fast**
  - no initial propagation cost





## Decomposition techniques

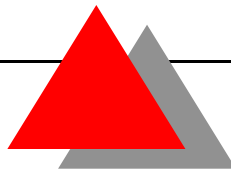
- ◆ **Cut problem into more manageable parts**
  - helps handle large/complex problems
- ◆ **Different ways of decomposing problems**
  - Hierarchical
    - ◆ **bottom-up and top-down**
    - ◆ **requires certain problem structure**
  - Structural
    - ◆ **considering different degrees of freedom independently**
  - Temporal/Spatial
    - ◆ **solving sub problems for limited time period or limited number of resources**





## Branch and bound

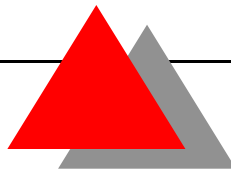
- ◆ **Create successive sub problems by enumeration on variables**
- ◆ **Exploration of search tree**
  - pruning of branches
  - lower bound approximation
- ◆ **Standard OR technique**
- ◆ **Search strategies must be defined carefully**
- ◆ **Very good results for complex problems**
- ◆ **High development effort**





## Neighborhood search

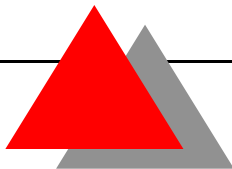
- ◆ **Search by finding initial solution and “improving” it**
  - feasible initial solution
  - modification function
  - cost evaluation
- ◆ **Allows different variations**
  - steepest ascent
  - hill climbing
  - simulated annealing
  - tabu search
  - genetic algorithms
- ◆ **Local optimization**





## Neighborhood search (2)

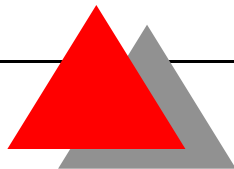
- ◆ **Constraint handling**
  - Constraints expressed in cost
  - Modification function checks constraints
- ◆ **Good for additive costs**
  - Local changes which improve costs
- ◆ **Difficult for very constrained problems**
  - Finding initial solution
  - Admissible modifications

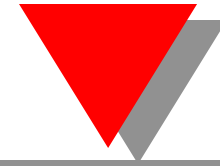




## Relaxation methods

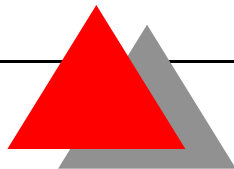
- ◆ Solving “simpler” problem helps finding solution to complex problems
- ◆ Ignoring/simplifying certain constraints
- ◆ Obtain lower/upper bounds
- ◆ Proof of optimality
- ◆ Initial solutions





## Part 2

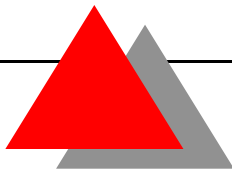
### Problem classification scheme





## Problem classification scheme

- ◆ Overview of attempts to solve problems
- ◆ Some large, operational systems
- ◆ Many examples are studies, not 'real' systems
- ◆ Many models do not scale (I think)
- ◆ Shows which areas are susceptible to approach



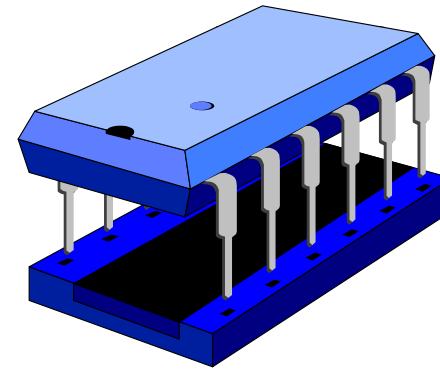
## Overview

- ◆ **Hardware design**
- ◆ **Compilation**
- ◆ **Financial problems**
- ◆ **Placement**
- ◆ **Cutting problems**
- ◆ **Stand allocation**
- ◆ **Air traffic control**
- ◆ **Frequency allocation**
- ◆ **Network configuration**
- ◆ **Product design**
- ◆ **Production step planning**
- ◆ **Production sequencing**
- ◆ **Production scheduling**
- ◆ **Satellite tasking**
- ◆ **Maintenance planning**
- ◆ **Product blending**
- ◆ **Time tabling**
- ◆ **Crew rotation**
- ◆ **Aircraft rotation**
- ◆ **Transport**
- ◆ **Personnel assignment**
- ◆ **Personnel requirement planning**

## Hardware design

### ◆ Different domains

- Circuit verification
  - ◆ check consistency with specification
- Diagnosis
  - ◆ find/explain fault in defective machine
- Testing
  - ◆ prepare tests to confirm proper operation
- Synthesis
  - ◆ create hardware design from specification
- Layout
  - ◆ create geometrical structure from design

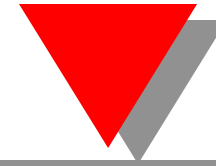


## Characteristics

- ◆ **Specialized solvers**
  - problem specific
  - not reusable for other domains
- ◆ **Narrow domain**
  - industrial usage restricted to few companies
- ◆ **Constraint methods used in conventional algorithms**
  - example D-Algorithm
- ◆ **Successful in right environment**
  - CVE (Siemens)
    - ◆ hardware verification tool for ASIC circuits

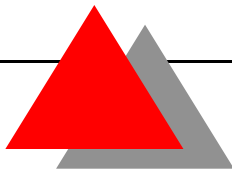
## Compilation

- ◆ **Register allocation**
  - graph coloring problem
- ◆ **Instruction scheduling**
  - pipelining/parallel execution
- ◆ **Microcode labeling (ECRC)**
  - distributing microcode over address space; simplified addressing
- ◆ **DSP scheduling (ECRC, cc(FD), COSYTEC)**
  - cyclic scheduling



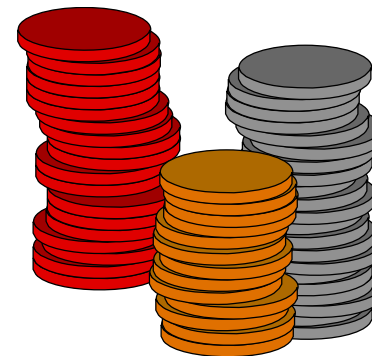
## Characteristics

- ◆ **Graph coloring problems**
- ◆ **Simple scheduling**
  - cyclic problems
  - disjunctive resources
  - machine assignment problems
- ◆ **Difficult to achieve in real-time**



## Financial problems

- ◆ **Portfolio management (SEVE - CDC)**
  - which shares to buy/sell
  - assumption on economic development
  - mixed mode solver
  - operational since 92
- ◆ **Asset/liability (Amro Bank)**
- ◆ **Stock option trading (C. Lassez)**
- ◆ **Constraint Spreadsheet (Hyvonen)**



## Characteristics

- ◆ **Continuous domain**
  - use rationals or reals
- ◆ **Non-linear constraints common**
  - linearisation
  - implement non-linear solver on top of linear one
- ◆ **Problem often incremental**
  - not all constraints known from beginning
  - programming with constraints (explanation, what-if)
- ◆ **Large problem instances**
- ◆ **Possible alternative techniques**
  - non-linear interval solvers
- ◆ **Requires proprietary information**
  - model (not only data) often confidential

## Placement

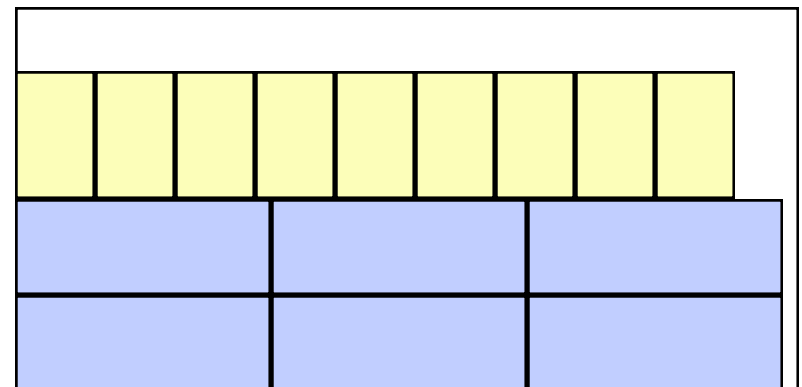
- ◆ **HIT container stacking (ICL)**
  - where to put containers to easily retrieve them later
- ◆ **Lorry loading (EBI)**
  - loading unloading of boxes in lorry
    - ◆ support constraint
    - ◆ stacking order
    - ◆ first in /last out
- ◆ **Container loading (Michelin)**
  - added degree of freedom
- ◆ **Map labeling (ECRC, Bull, COSYTEC, Gist)**
  - where to put labels on map
  - preference position not always achievable
  - depends on right model

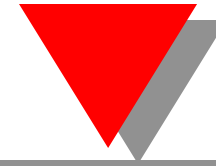
## Characteristics

- ◆ **2D**
  - often overconstrained
  - strong preferences not always achievable
  - very good results can be obtained
- ◆ **3/4D**
  - very hard or very easy
  - needs powerful heuristics
- ◆ **General**
  - very poor results with syntactic methods
  - some common constraints very hard to express
  - specialized domain heuristics not easy to compute

## Cutting problems

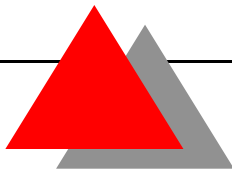
- ◆ **Cutting stock (ECRC)**
  - cutting rectangles from rectangles
  - 2D finite problem
- ◆ **Made (Dassault)**
  - combining sheet metal pieces for aircraft parts
  - approximated by combination of rectangles
- ◆ **Glass cutting (Oz)**
- ◆ **Wood cutting for furniture (Prolog III)**





## Characteristics

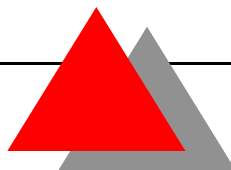
- ◆ **Strong competition from MIP**
  - continuous roll cutting
- ◆ **Problems to handle irregular shapes**
  - leather, clothes
  - problem for any mathematical model
- ◆ **Good heuristic solutions**





## Stand allocation

- ◆ **APACHE (COSYTEC)**
  - stand allocation for airport
- ◆ **HIT (ICL)**
  - assign ships to berths in container harbor
- ◆ **Train platform assignment (Ilog, Siemens)**
  - assign trains to platforms in at stations
- ◆ **Refinery berth allocation (ISAB)**
  - where to load/unload ships in refinery





## APACHE

### ◆ Stand allocation system

- originally developed with Air France, CDG2
- packaged for large airports

### ◆ Complex constraint problem

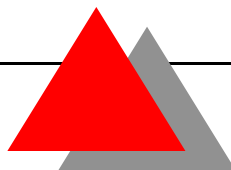
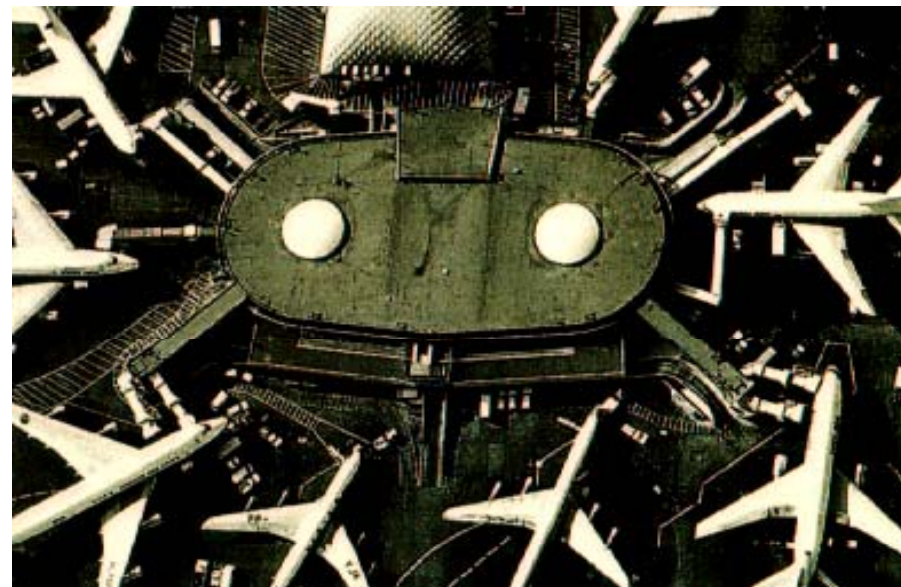
- technical & operational constraints
- incremental re-scheduler

### ◆ Cost model

- maximize no passengers in contact
- minimize towing, bus usage

### ◆ Status

- technology demonstrator



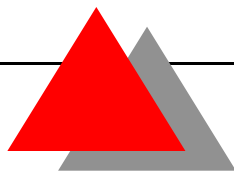
## Characteristics

- ◆ **Base constraint relatively easy**
  - Graph coloring in interval graphs
  - Complete propagation possible for alldifferent
  - additional constraints/cost model more complex
- ◆ **Rescheduling requirements**
  - constraints change with every delay
  - resolving problem without disturbing all of previous solution
- ◆ **Solver can be very fast**
  - few seconds
- ◆ **Proof of optimality very complex due to symmetry**
  - needs separate lower bound calculation



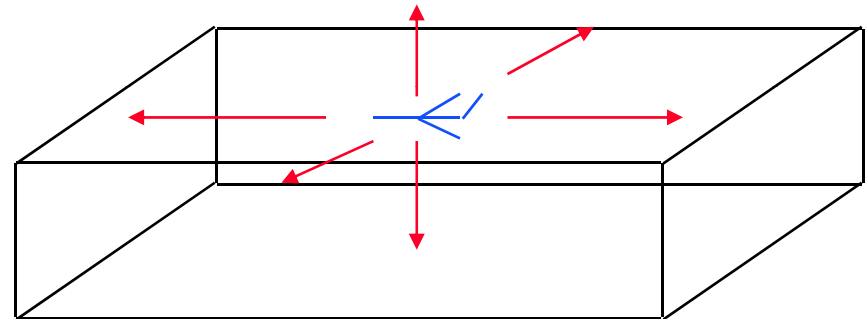
## Air traffic control

- ◆ **CENA**
  - slot capacity
- ◆ **Thomson**
  - landing approach
- ◆ **Matra**
  - mission planning (military)



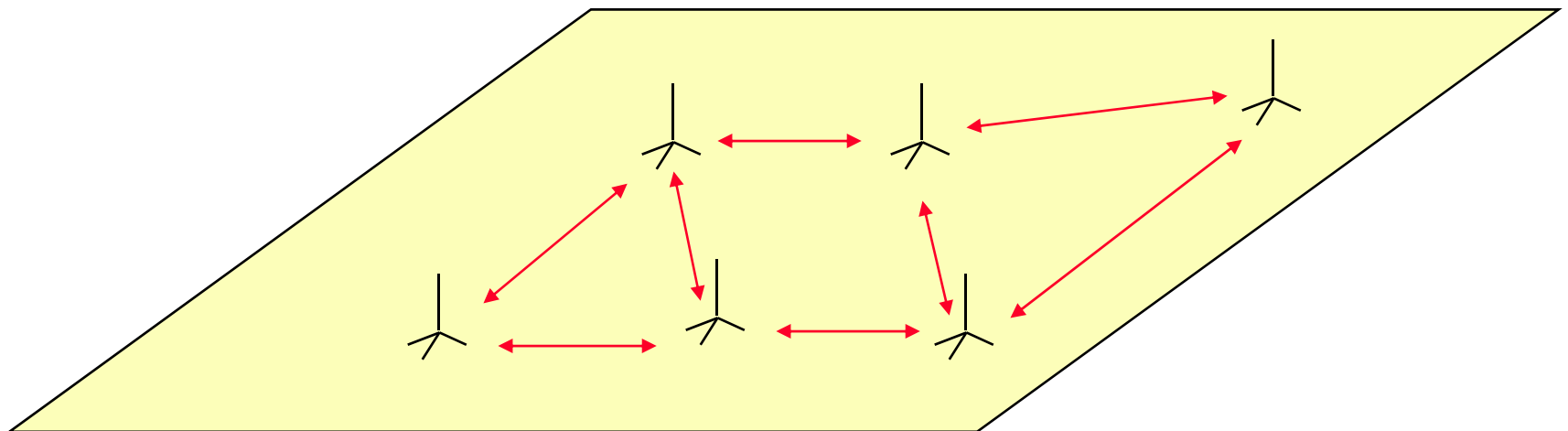
## Characteristics

- ◆ **Temporal and spatial constraints**
  - Box model
- ◆ **General ATC problem very hard to express**
  - trajectories as 4D objects
  - “closeness” of trajectories
- ◆ **Large problem sizes**
- ◆ **Special case solutions interesting**



## Frequency allocation

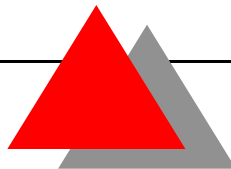
- ◆ Thomson
- ◆ SICS
- ◆ Celar Benchmark (Bull, Ilog, COSYTEC)





## Characteristics

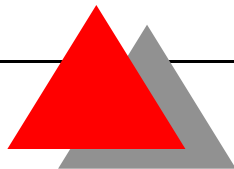
- ◆ **Optimization difficult**
  - symmetry reduction
- ◆ **Solver too weak**
  - cliques in graphs
- ◆ **Locally overconstrained**
  - some constraints are actually preferences
- ◆ **Model may vary depending on degree of exactness**
  - disequality/distance constraints





## Network configuration

- ◆ **Locarim (France Telecom, COSYTEC)**
  - cabling of building
- ◆ **Planets (UCB, Enher)**
  - electrical power network reconfiguration
- ◆ **Load Balancing in Banking networks (ICON)**
  - distributed applications
  - control network traffic
- ◆ **Water Networks (UCB)**



## FRANCE TELECOM - LOCARIM

### ◆ Intelligent cabling system for large buildings

- developed with Telesystemes for France Telecom

### ◆ Application

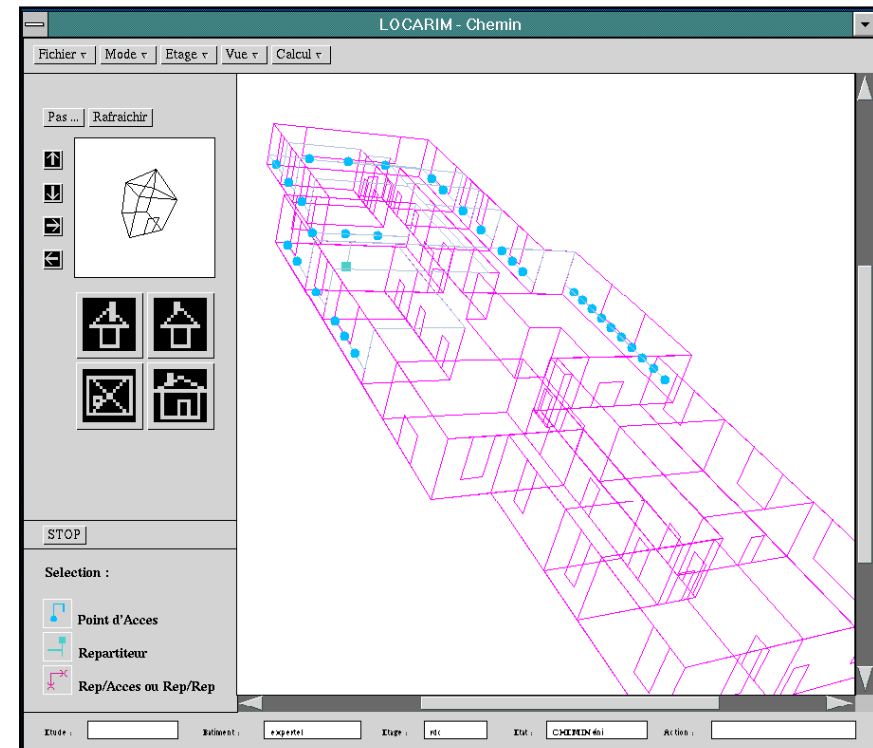
- input scanned drawing
- specify requirements

### ◆ Optimization

- minimize cabling, drilling, switches
- shortest path

### ◆ Status

- operational in 5 Telecom sites
- generates quotations



## Characteristics

- ◆ **Different types of problems**
- ◆ **Many related to warehouse location**
  - with/without capacity
- ◆ **Mixed methods worthwhile**
  - finite domain solver
  - rational solver
- ◆ **Competition from MIP**
  - simple model
  - nice mathematical properties

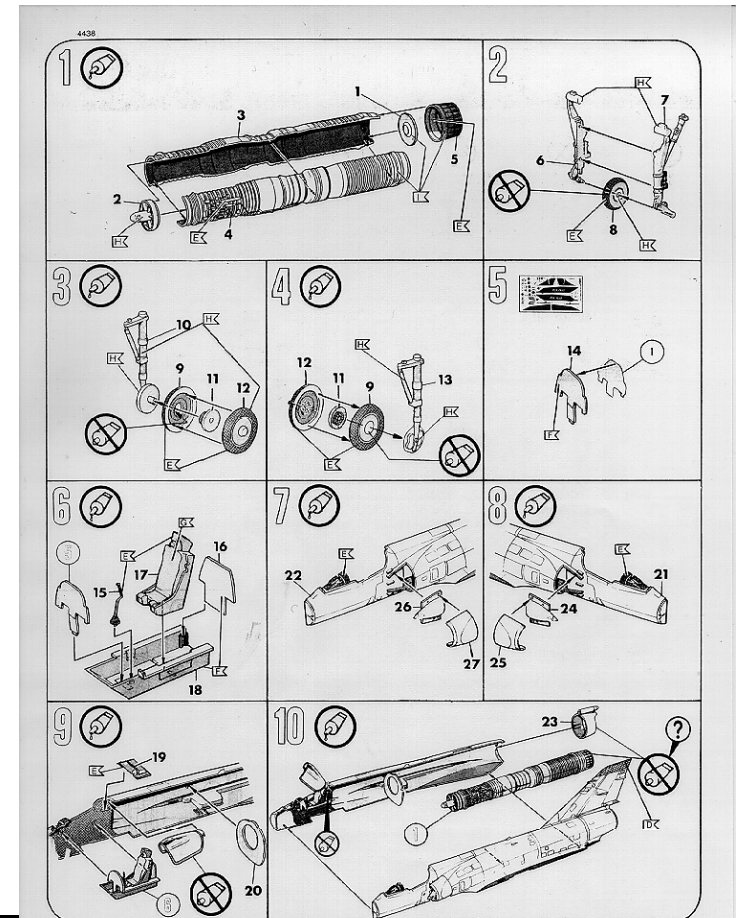
## Product design

- ◆ **Key system generation (Vachette, Bull)**
  - design key structure for large buildings
  - one key opens multiple doors
  - security restrictions in different levels
  - parts of key control different locks
  - interaction of different access groups
- ◆ **Mechanical design (Cisa)**

## Production step planning

### ◆ COCA (Dassault)

- define in which order the production steps are performed
- basis for scheduling
- very large problem
  - ◆ several 10000 steps
  - ◆ decomposition possible

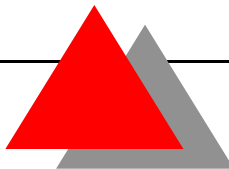


PAGE 2 Should you find any missing parts, please write to REVELL AG, 2609 + 2620, department X, D-4980 Emden, W. Germany, and we will send you these as quickly as possible with our compliments. Please indicate clearly with a cross in the instruction leaflet, which piece(s) are missing. Always remember to write your name and address clearly and in full.



## Characteristics

- ◆ **Temporal and spatial constraints**
  - some steps must be done before others
- ◆ **Access to location**
  - not possible to work on two adjacent compartments concurrently
- ◆ **Rotation state of aircraft frame**
  - allows /excludes access
- ◆ **Safety rules**
  - operations may not be performed at the same time



## Production sequencing

- ◆ **Amylum (Beyers)**
  - Glucose production
- ◆ **Cerestar (Beyers)**
  - Glucose production
- ◆ **Car Sequencing (ECRC, COSYTEC)**
  - assembly line scheduling
- ◆ **Bowater (Bull, COSYTEC)**
  - carton printing, reuse of colors
- ◆ **MOSES (COSYTEC)**
  - animal feed production

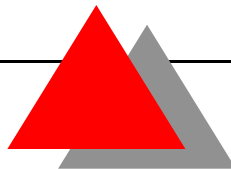
## Characteristics

- ◆ **Forbidden sequences**
  - this product must never follow that product
  - this product should not follow that product
- ◆ **Setup cost/time**
  - cleaning time
  - downgrading product
  - waste
- ◆ **Combination with scheduling**
  - due dates
  - machine choice
- ◆ **Additional constraints**
  - capacity



## Production scheduling

- ◆ **Plane (Dassault)**
  - mid/long term scheduling
- ◆ **Made (Dassault)**
  - short term work cell scheduling
- ◆ **Saveplan (Sligos)**
  - production scheduling
- ◆ **ATLAS (Beyers, COSYTEC)**
  - herbicide manufacturing
- ◆ **MOSES (COSYTEC)**
  - animal feed production
- ◆ **Trefi Metaux (Sligos)**
  - heavy industry production scheduling



## ATLAS

### ◆ Chemical packaging & inventory control system

- developed for US agro-chemical supplier
- joint development with Beyers & Partners

### ◆ Extensive use of CHIP interfaces

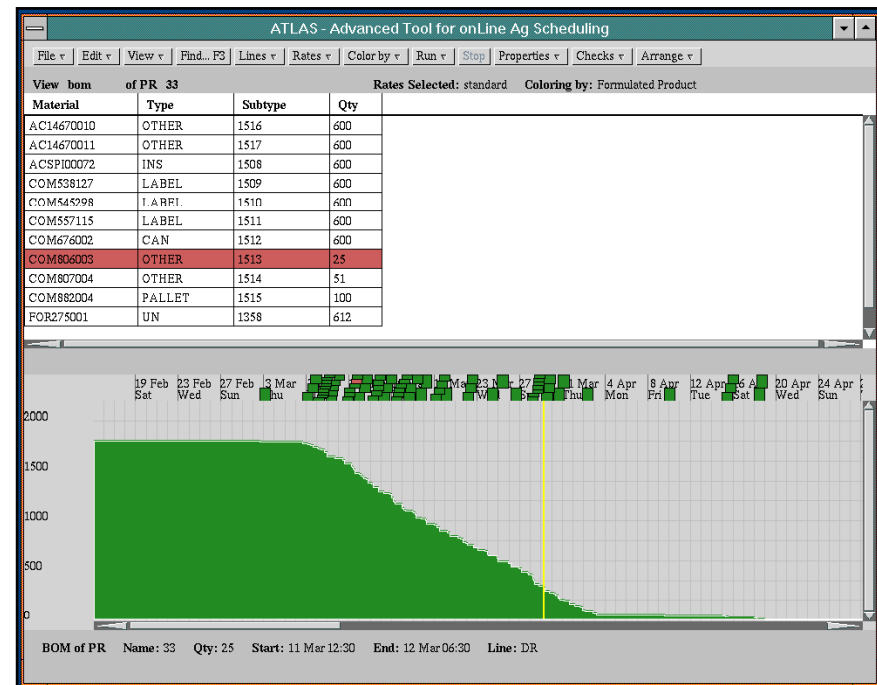
- XGIP GUI interfacing to RDBMS
- multi-user UNIX & PC system

### ◆ Scheduling

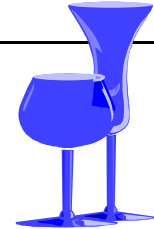
- formulation & packaging
- checks highlights problems

### ◆ Benefits and status

- operational since Jun 93
- better control, reduced stock



## VCA - ORDO-VAP



### ◆ Production scheduling for glass factory

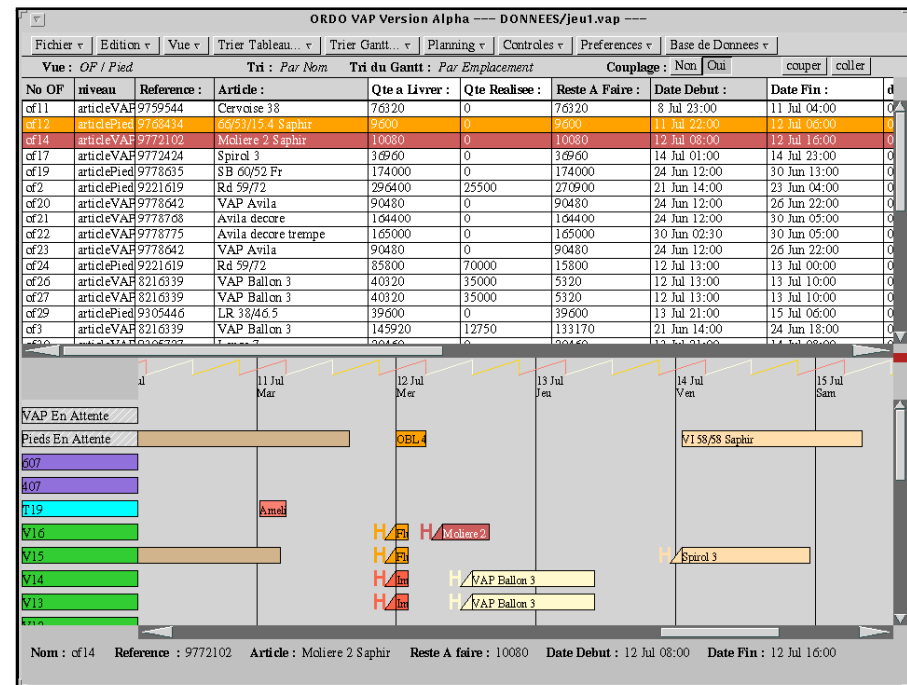
- integrated with Ingres Information system
- manual and automatic scheduling

### ◆ Constraints

- multi-stage manufacturing
- consumer/producer
- varying production rates, setup
- balance manpower utilization
- minimize downtime

### ◆ Status

- 2 phases
- operational in March 96
- will replace manual operation

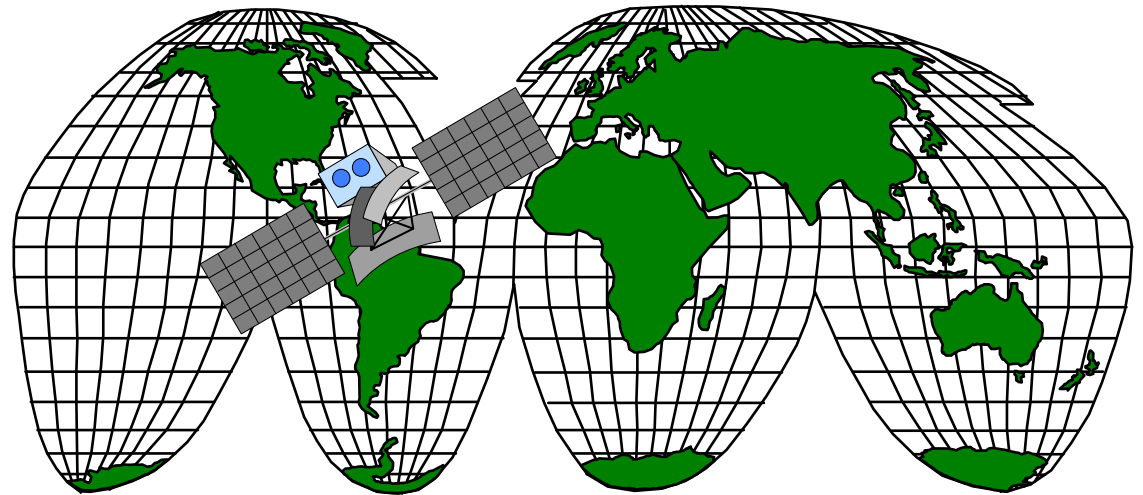


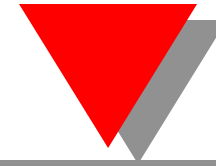
## Characteristics

- ◆ **Large systems operational**
- ◆ **Complete environments**
  - integration
  - frameworks
- ◆ **Many types of constraints**
  - precedence
  - disjunctive /cumulative resources
  - producer/consumer
  - machine assignment
  - setup
  - due dates/release dates
- ◆ **Well developed methodology**

## Satellite tasking

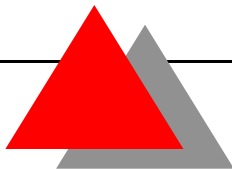
- ◆ **Dassault**
  - low earth orbit satellite configuration power management
- ◆ **Alcatel**
  - earth observation scheduling
    - ◆ memory
    - ◆ transmission times
    - ◆ energy use
    - ◆ observation windows





## Maintenance planning

- ◆ **Sema**
  - aircraft maintenance
- ◆ **Coopers & Lybrand**
- ◆ **Hong Kong Public Transport**
  - maintenance jobs on train/subway service
- ◆ **Edia - SNCF**



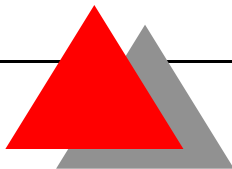
## Characteristics

- ◆ **Cost model very weak**
  - interested in sum of costs
- ◆ **Problem set not known a priori**
  - some jobs may be postponed/canceled
- ◆ **Problem separable in time periods**
  - sequential/independent optimization
  - reduces complexity
  - provides lower bounds/heuristics

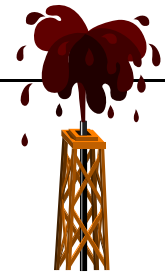


## Product Blending

- ◆ **Forward (TECHNIP, COSYTEC)**
  - gasoline blending
  - crude mix
- ◆ **Sanofi (ILOG, COSYTEC)**
  - cosmetics
- ◆ **Michelin**
  - rubber blending, rework optimization



## FINA - FORWARD



### ◆ Oil refinery production scheduling

- joint development by TECHNIP and COSYTEC
- incorporates ELF FORWARD LP tool

### ◆ Schedules daily production

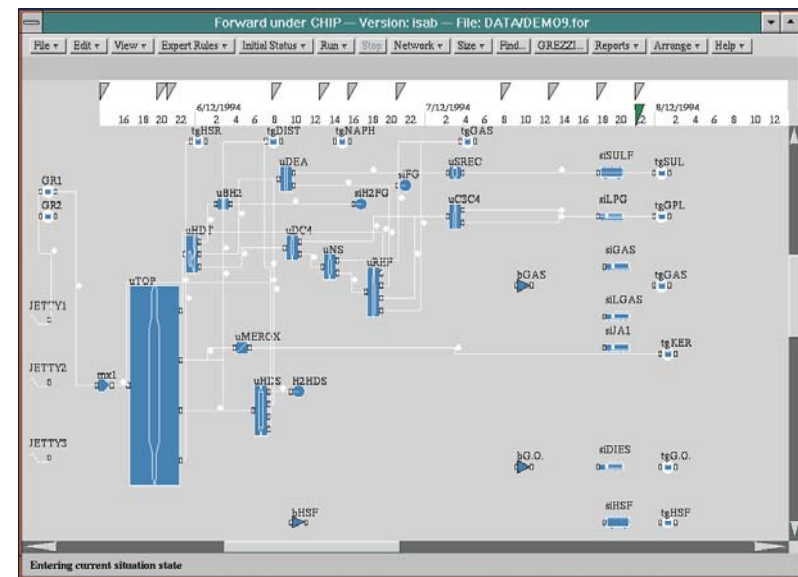
- crude arrival -> processing -> delivery
- design, optimize and simulate

### ◆ Product Blending

- explanation facilities
- handling of overconstrained problems

### ◆ Status

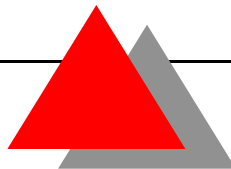
- generic tool developed in 240 man days
- operational since June 94
- Operational at FINA, ISAB, BP





## Characteristics

- ◆ **Strong domain for LP/IP**
  - constraint model only based on inequalities
  - finite domain solvers don't offer much
  - continuous domains/cutting plane methods
- ◆ **Constraints can provide explanation facility**
  - programming with constraints
  - advantage over LP packages
- ◆ **Handles smaller problem size than LP/MIP systems**



## Time tabling

### ◆ School/university time tables

- which courses are held
  - ◆ when
  - ◆ by whom
  - ◆ in which room

### ◆ Exam scheduling

- which exams to place when in which rooms, possibility to combine exams in same room

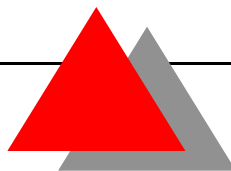
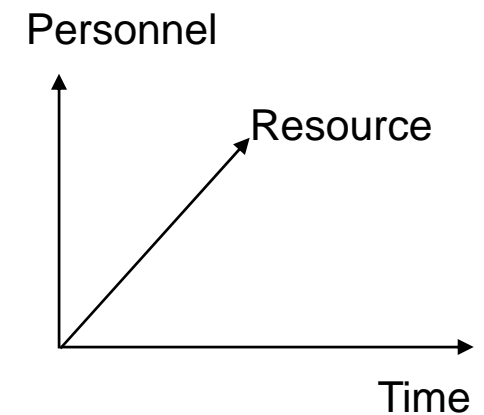
### ◆ Training course scheduling (Nat West)

- which courses to run in which week of year
  - ◆ limited accommodation
  - ◆ course sequences
  - ◆ course repetition



## Problem scheme

Time	Mon	Tue	Wed	Thu	Fri
8-10					
10-12					
14-16	Analysis I Prof A Room 221				
16-18					



## Characteristics

- ◆ **Experimental systems**
- ◆ **Solvers too weak**
  - easy problems simple
  - hard problems impossible
- ◆ **Relaxation of constraints required**
  - overconstrained problems
  - strong preferences
- ◆ **Balancing of time table**
  - equal quality for everybody
- ◆ **Dedicated, specialized packages exist**

## Crew rotation

- ◆ **Pilot (SAS, COSYTEC)**
  - re-planning
- ◆ **DAYSY (Lufthansa, COSYTEC, Sema, U. Patras)**
  - day to day management
- ◆ **Air Littoral (PrologIA)**
  - use of Simplex
- ◆ **Servair (GSI)**
  - capacity planning/scheduling/assignment
- ◆ **NWRR (COSYTEC)**
  - train driver rotations
- ◆ **SuperBus (PrologIA, Brunel U.)**
  - public transport

## SAS- PILOT

### ◆ Intelligent re-scheduling

- SAS have 115 aircraft, 3 home bases, 3000 cabin 2000 flight
- initial development by COSYTEC continued by SAS Data

### ◆ Solve open flights

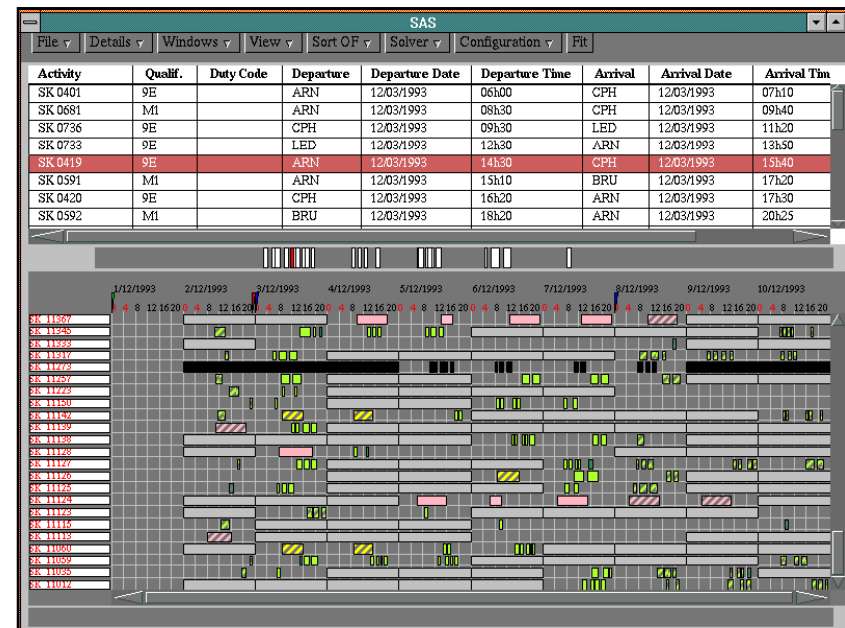
- delay, illness, cancellation, new flight
- 50% in 5 minutes for 100 crew
- 80% in 5 hours for 1000 crew

### ◆ Black box solver

- based on cycle constraint
- constraint + legality checker

### ◆ Status

- operational Sept 1995



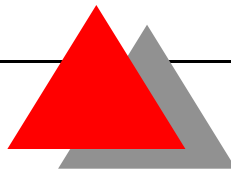
## Characteristics

- ◆ **Very complex constraints**
  - evolving over time
- ◆ **Difficult to express/check**
  - dedicated rule checking systems
- ◆ **Very large problem sets**
  - several thousand crew
  - several ten thousand flights
- ◆ **Crew preferences**
  - incompatible with each other
- ◆ **Soft/Hard rules**
  - Government regulations, safety regulations, seniority rules
- ◆ **Needs very expressive/powerful solver**
- ◆ **Competition: very strong monthly planning tools based on LP**



## Aircraft/Train rotation

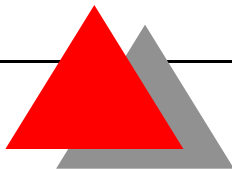
- ◆ **SNCF - Bull**
  - capacity planning : which trains to move overnight
  - specialized algorithm: min flow
- ◆ **SNCF - Ilog**
  - train engine rotation
  - specialized algorithm: TSP
- ◆ **BA - IC Parc**
  - aircraft rotations
  - repair methods





## Problem scheme

- ◆ **List of services**
  - Dep 10:00 CDG Arr 10:05 LHR
- ◆ **Covering services with available engines**
- ◆ **Passive movement to make machines available at right place**
- ◆ **Maintenance/service stops**
- ◆ **Balancing/minimizing engine usage**
- ◆ **Location continuity**
  - Start/stop at depot/ home base



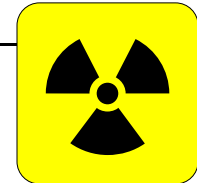
## Characteristics

- ◆ **Less constrained than crew problems**
  - no unions to worry about
- ◆ **Location continuity**
  - added dimension over scheduling
- ◆ **Large problem sizes**
  - all of French train services; decomposition possible
- ◆ **Unknown qty of passive transport required**
  - difficult to express a priori with constraints
- ◆ **Results show problems of expressing/solving with syntactic methods**
  - resolve problem with dedicated, non incremental algorithm

## Transport

- ◆ **EVA (EDF, Gist, COSYTEC)**
  - nuclear waste transport
- ◆ **EBI**
  - warehouse - customer transport
- ◆ **TACT (COSYTEC)**
  - integrated transport food manufacturing
- ◆ **PASZA (COSYTEC)**
  - feed mill transport
- ◆ **SIPE**
  - bus transport

## EDF- EVA



### ◆Transportation of nuclear waste

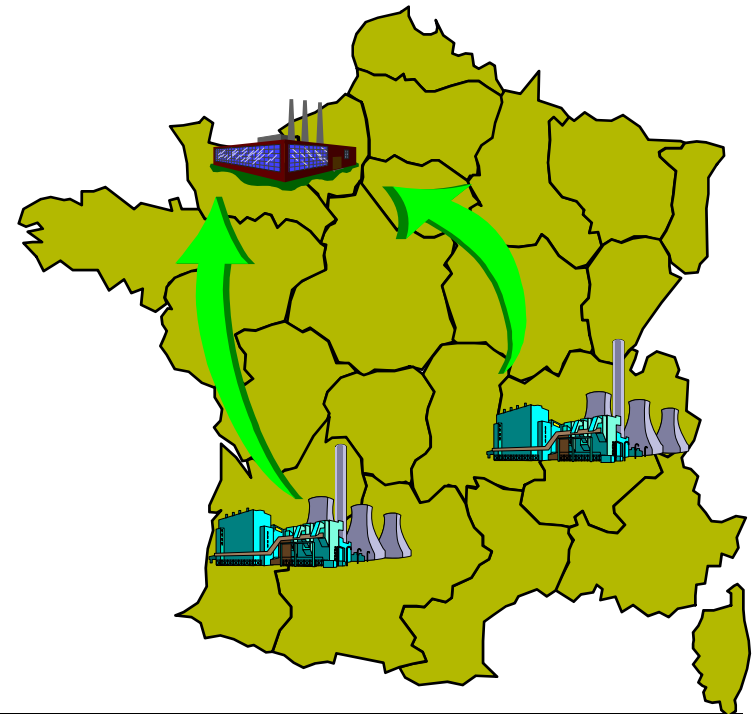
- developed by GIST + COSYTEC
- plans evacuation and transport for 54 sites

### ◆Constraints

- availability of transport vehicles and vessels
- no and capacity of storage tanks
- compatibility of waste to vessels
- size of convoy, time

### ◆Status

- operational since Oct 94
- 6 month plan in 5 minutes



## TACT

### ◆Transport planning and assignment

- plans activities for factories
- assigns activities to teams, drivers, lorries, fork lifts

### ◆Problem solver

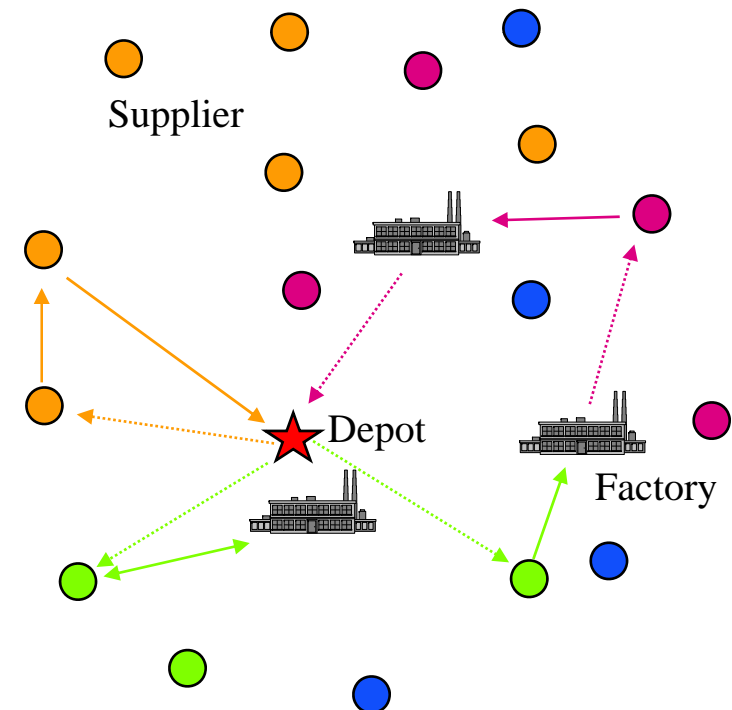
- generates minimum no trips
- balance production, optimizes resources

### ◆Rules, constraints

- production, storage, legal, vet
- roster, workforce, unavailability

### ◆Status

- operational Feb 1995
- developed Aug 94-Jan 95



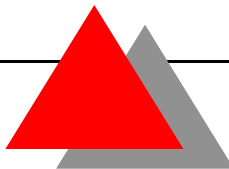
## Characteristics

- ◆ **Location continuity**
  - start end of trips
  - depots
- ◆ **Passive transportation**
  - unknown quantity
- ◆ **Important scheduling component**
  - driven by crucial resource (lorries, drivers, supply/demand side)
- ◆ **Producer/Consumer behavior**
  - JIT delivery
  - limited stock, shelf life
- ◆ **Even more difficult if work rules must be handled**
  - total driving time, breaks, rest periods, start/end time, rotas

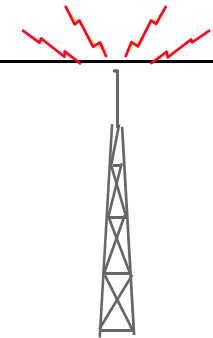


## Personnel assignment

- ◆ **Servair (GSI, ITMI, COSYTEC)**
  - train bar/restaurant
- ◆ **RFO (Gist, COSYTEC)**
  - reporters/technicians for TV/radio
- ◆ **Banque Bruxelles Lambert (Ilog)**
  - bank personnel
- ◆ **Nurse scheduling (Ilog, Bull)**
  - hospital
- ◆ **Crisis Management (Bull)**
  - Olympic winter games 1992



## RFO - MAESTRO



### ◆ Assignment of technical staff to tasks

- overseas radio broadcaster - Radio France Outre-mer
- joint development by GIST and COSYTEC

### ◆ Features

- schedule manually, check, automatic
- rule builder to specify cost formulas

### ◆ Optimization

- minimize overtime, temporary staff
- compute cost of schedule

### ◆ Status

- operational Dec 95
- to be installed worldwide in 9 sites

RFO Tableaux de Services — DATA/previsionnelrfo —

Fichier v | Edition v | Vue v | Tri de la Liste v | Tri du Gantt v | Chercher... | Planifier v | Preferences v | Verification v | Rapports v | Aide v

Vue : Activités Nombre d'objets = 57

Produit	Phase	Activité	Lieu	Debut	Fin	Pauses	ARTV	Script	Cadreur
JT du soir	Direct Studio	Regie + Plateau	studio	12 Mai 19:30	12 Mai 21:00	zero	0	0	2
JT du soir	Direct Studio	Regie + Plateau	studio	11 Mai 19:30	11 Mai 21:00	zero	0	0	2
JT du soir	Direct Studio	Regie + Plateau	studio	10 Mai 19:30	10 Mai 21:00	zero	0	0	2
JT du soir	Direct Studio	Regie + Plateau	studio	9 Mai 19:30	9 Mai 21:00	zero	0	0	2
JT du soir	Reportages	Script	station	15 Mai 15:00	15 Mai 21:00	zero	0	1	0
JT du soir	Reportages	Script	station	14 Mai 15:00	14 Mai 21:00	zero	0	1	0
JT du soir	Reportages	Script	station	13 Mai 15:00	13 Mai 21:00	zero	0	1	0
JT du soir	Reportages	Script	station	12 Mai 15:00	12 Mai 21:00	zero	0	1	0
JT du soir	Reportages	Script	station	11 Mai 15:00	11 Mai 21:00	zero	0	1	0
JT du soir	Reportages	Script	station	10 Mai 15:00	10 Mai 21:00	zero	0	1	0

## SERVAIR - CREW

### ◆ Crew rostering system

- assign service staff to TGV train timetable
- joint implementation with GSI

### ◆ Problem solver

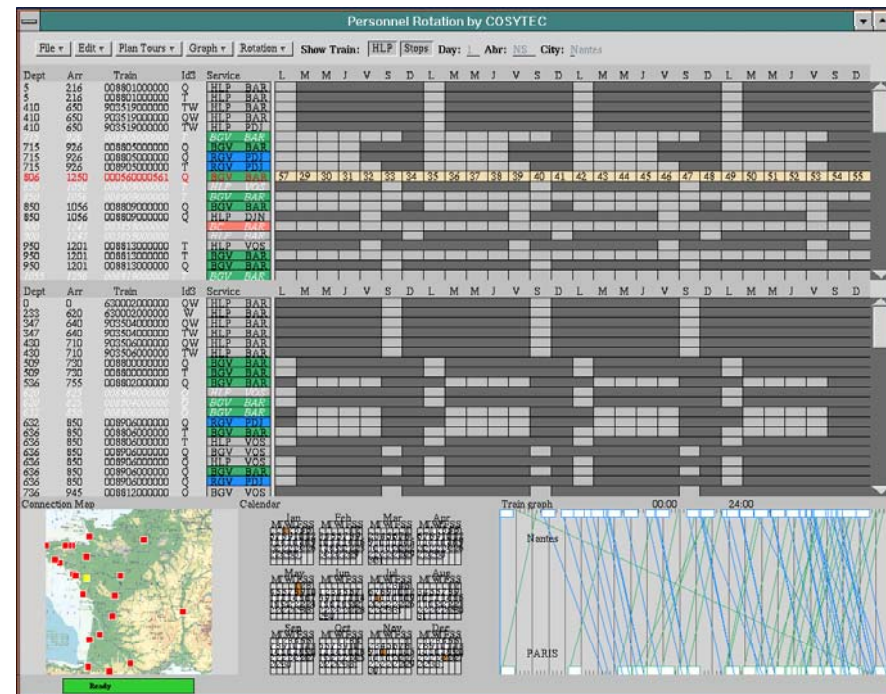
- generates tours/cycles
- assigns skilled personnel

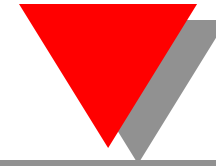
### ◆ Constraints

- union, physical, calendar

### ◆ Status

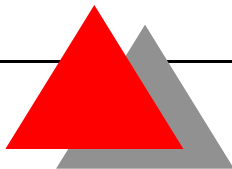
- operational since Mar 1995





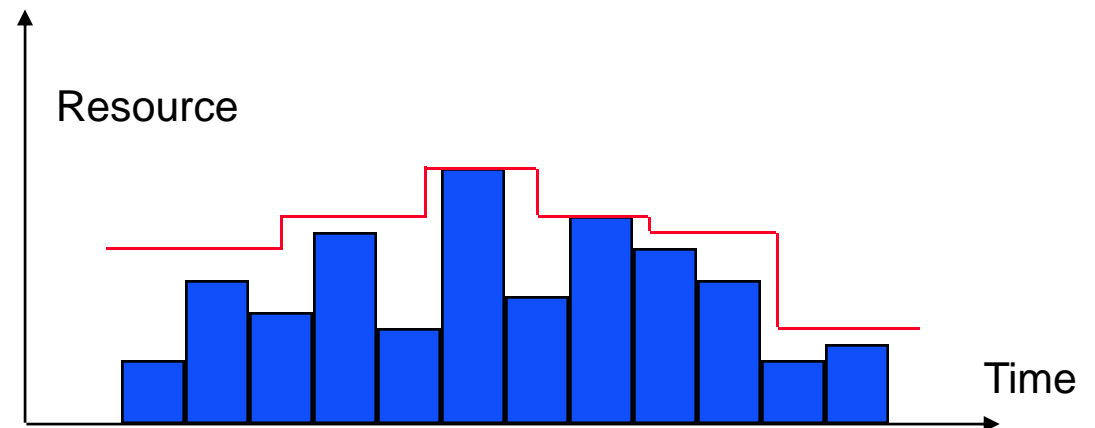
## Characteristics

- ◆ **Work rules**
  - as for transport and rostering
- ◆ **Balancing**
  - spread difficult/tedious jobs
  - total work time per month
- ◆ **Perfect problems very hard**



## Personnel requirement planning

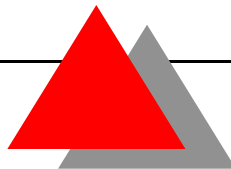
- ◆ **Ghost (Sligos)**
  - capacity planning credit card service
- ◆ **Havas (COSYTEC, EBI)**
  - ground crew management
- ◆ **911 planning (2LP)**
  - emergency center capacity planning

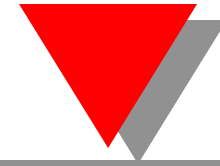




## Characteristics

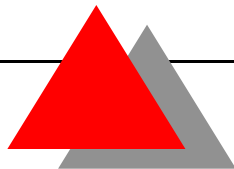
- ◆ **Rather small problem size**
- ◆ **Covering demand per time unit**
- ◆ **Lower bound constraints**
- ◆ **Alternative models**
  - integer programming: inequalities
  - capacity planning: cumulative
- ◆ **IP approach quite strong**

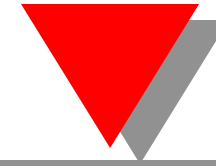




## Part 3

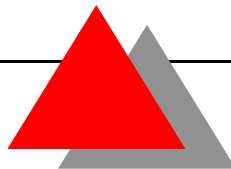
## Evaluation





## Which problems failed

- ◆ **Not much information available**
- ◆ **Paper by J.Y. Cras, ILPS 1994**
- ◆ **Difference between**
  - project failure
    - ◆ **customer is not happy**
  - problem failure
    - ◆ **constraint researcher is not happy**



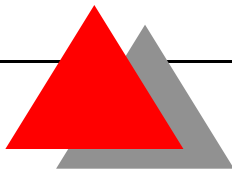
## Project failure

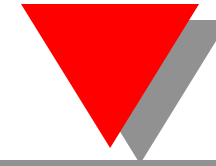
- ◆ **Project management**
  - technically too ambitious
  - due dates not achievable
  - effort not estimated correctly
- ◆ **End - user acceptance**
  - end user not involved early on
- ◆ **Business process change**
  - need disappears while system is being developed
  - problem changes beyond recognition



## Problem failure

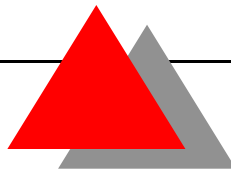
- ◆ **Wrong problem**
- ◆ **Wrong solver**
- ◆ **Wrong model**
- ◆ **Wrong test case**

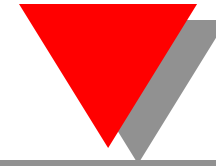




## Wrong problem

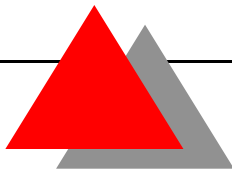
- ◆ **Solving the wrong problem**
  - focusing on technology rather than need of customer
- ◆ **Pure problem**
  - specialized methods/algorithms exist
- ◆ **Relaxation essential**
  - if no constraints are hard, then there is no propagation
- ◆ **Too generic**
  - solving “the generic scheduling” problem
  - using problem specific knowledge is key to success





## Wrong solver

- ◆ **Wrong domain**
  - ex. solving sets of inequalities by bound propagation
  - find most general solution where only one particular solution is required
- ◆ **Solver too weak**
  - idea of solving hard problems by simple methods
  - this is why global constraints were introduced



## Wrong model

- ◆ **Bad choice of variables**
  - avoid 0/1 domain variables
  - avoid very large domains
- ◆ **Constraints do not propagate**
  - important to express all constraints inside model
  - not enough if constraints do not propagate
- ◆ **Bad strategy**
  - use problem specific knowledge
  - try different methods
- ◆ **Cost model too weak**
  - important when doing search
  - proving optimality only possible with good lower cost bound



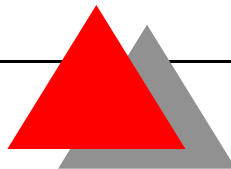
## Wrong test case

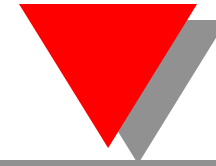
### ◆ Problem does not scale

- classical AI problem
- test with real data
- see whether actual solution satisfies constraint model
- ideally, test with full size data

### ◆ Not enough test cases

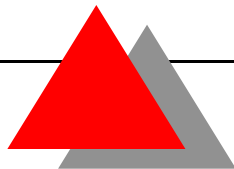
- easy to over-optimize one test case (benchmarks)
- day to day system requires test data from all time periods
  - ◆ seasonal demand variation
  - ◆ peak business
  - ◆ special cases (holidays)





## Modeling check list

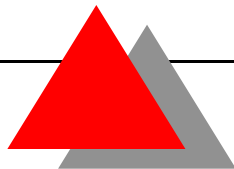
- ◆ **Soft constraints**
- ◆ **Overconstrained problems**
- ◆ **Preferences**
- ◆ **Balancing**
- ◆ **Non-local cost**
- ◆ **Planning type problems**
- ◆ **Passive transport**





## Comparison studies

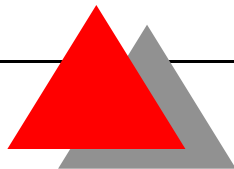
- ◆ **Comparing two methods is much more difficult than testing one**
- ◆ **Most tests topical or small scale**
- ◆ **some comparison of benchmark results**
- ◆ **Tests shown**
  - CLP - OR
  - CLP - LP/MIP
  - CLP - AI
  - CLP - local search
  - CLP/FD - CLP/R





## CLP - OR (specialized algorithms)

- ◆ Warehouse location
- ◆ Job shop
- ◆ Patterson / Alvarez benchmarks



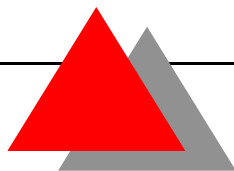
## CLP - LP/MIP

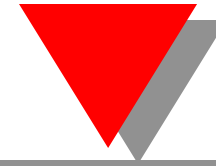
- ◆ Warehouse location
- ◆ Setup scheduling
- ◆ Cutting stock
- ◆ ATC slot allocation
- ◆ Progressive party problem
- ◆ Network flow (Train rotations)
- ◆ Disposing problem (Bisdorff)
- ◆ 2LP
- ◆ Banking networks



## CLP - AI

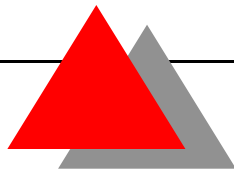
- ◆ Car sequencing





## CLP - Local search

- ◆ Car sequencing
- ◆ Map labeling
- ◆ Scheduling (BT)



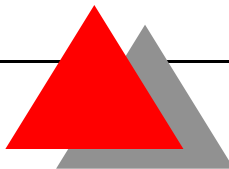
## CLP vs CLP

- ◆ **Comparison of different solver methods for some applications**
- ◆ **Warehouse location**
- ◆ **Hoist scheduling**
- ◆ **Bisdorff**



## What really works

- ◆ **Which of the problems should be solved by CLP**
  - CLP appropriate tool
  - competing techniques have no clear advantage
- ◆ **Methodology**
  - how to express constraints
  - standard models
  - sets of heuristics available



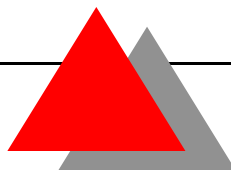
## Scheduling

- ◆ **Good methodology**
  - see PAP 95 tutorial on planning and scheduling
- ◆ **Constraints competitive**
  - benchmark results are comparable to best dedicated systems
- ◆ **Good strategies known**
  - both AI and OR type heuristics
- ◆ **Projects can still be complex**
  - interfaces
  - integration
  - strategies



## Allocation

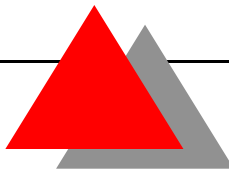
- ◆ **Solver rather simple**
  - disequality/alldifferent/diffn
  - pure problem easy
  - complete propagation for significant sub-problems
- ◆ **Rescheduling rapid**
  - solver finds solution in few seconds
- ◆ **Much simpler than scheduling**
  - do not generalize if not required





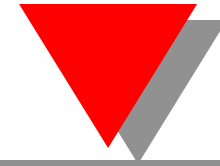
## Transport

- ◆ **Solver quite complex**
  - location continuity
  - unknown amount of passive transport
- ◆ **Multiple loading/unloading a problem**
  - interaction of all activities in one tour
- ◆ **Good strategies required**
- ◆ **Problem size can be limiting**
  - several hundred nodes OK
- ◆ **Handling of personnel work rules makes things very complex**

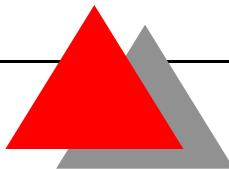
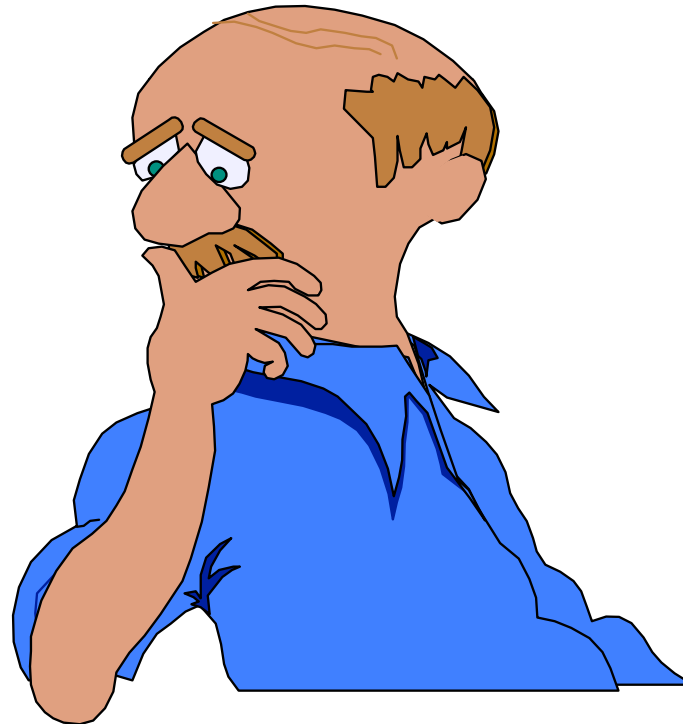


## Crew rotation

- ◆ **Work rules can be very complex**
  - point systems
  - large amounts of data (ex pilot qualification for airports)
- ◆ **Preferences not handled well**
- ◆ **Does not handle long periods**
  - too many activities
  - not enough connection between early and late activities
- ◆ **Rest periods**
  - expressing constraints already complex
  - deducing information very hard
- ◆ **No other tool handles constraints well**
  - Column generation: generate & combine method



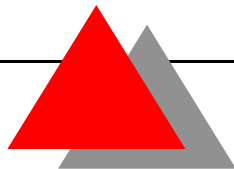
## How to continue?





## Sources

- ◆ Examples drawn from multitude of sources
- ◆ Some are “personal communication” only
- ◆ Literature list





---

## Summary

---

- ◆ **CLP - Emphasis on finite domains**
  - not enough data for other problem solvers
- ◆ **Application classification**
  - what has been done in the field
- ◆ **Explaining problems**
  - why projects/problems fail
- ◆ **Modeling checklist**
  - what to watch out for

