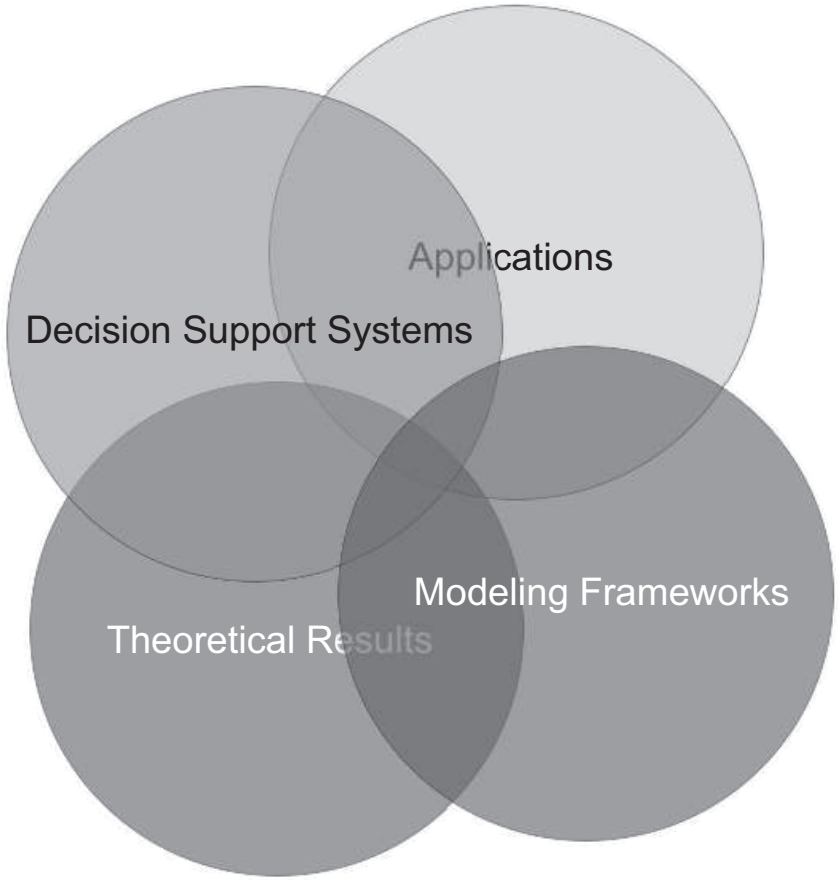
Stochastic Constraint Programming: a seamless modeling framework for decision making under uncertainty

Dr Roberto Rossi[1]

[1]LDI, Wageningen UR, the Netherlands

Mansholt Lecture

A Complete Overview

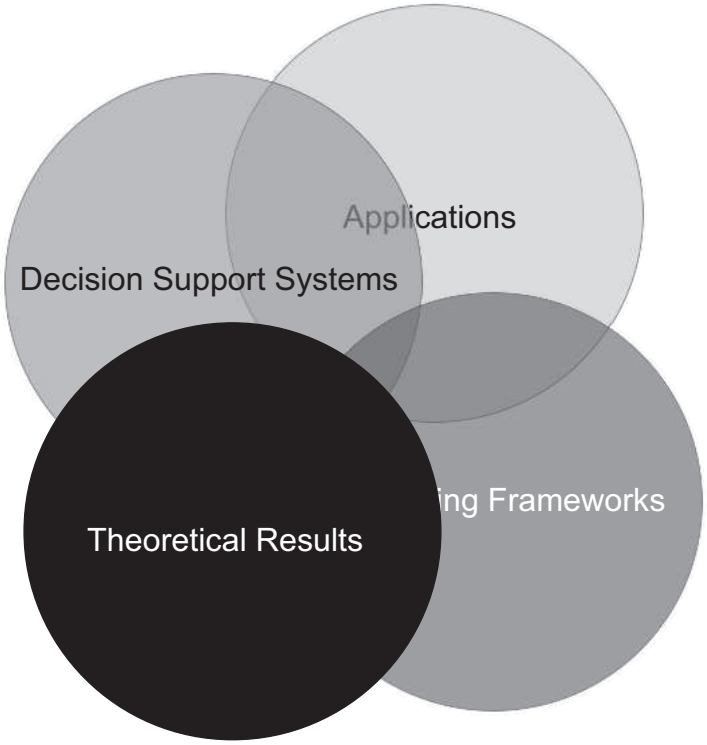


Theoretical Results

Algorithm 1: Filtering Algorithm

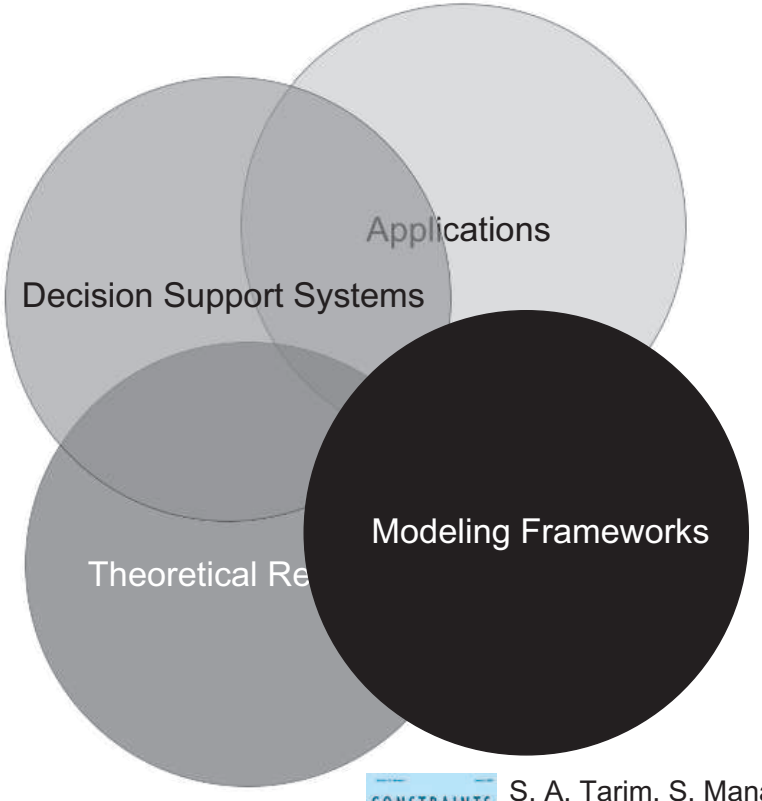
input : h; $\mathcal{PT}$; $\mathcal{A}$.
output: Filtered $\mathcal{PT}$ wrt h.

```
1  begin
2      for each i ∈ {0,...,N−1} do
3          for each v ∈ D(PT[i]) do
4              f[i,v] ← 0;
5      for each p ∈ Ψ do
6          Create a copy c of h_p and of the decision variables it constrains;
7          Enforce GAC on c using A;
8          for each index i of the variables in c do
9              for each v in domain of the copy of PT[i] do
10                 f[i,v] ← f[i,v] + Pr{arcs(p)};
11         delete c and the respective copies of the decision variables;
12     for each i ∈ {0,...,N−1} do
13         max[i] ← 0;
14         for each v ∈ D(PT[i]) do
15             max[i] ← max(max[i], f[i,v]);
16     for each k ∈ {1,...,m} do
17         g[k] ← 0;
18         for each i ∈ M_k do
19             g[k] ← g[k] + max[i]
20     for each k ∈ {1,...,m} do
21         for each i ∈ M_k do
22             for each v ∈ PT[i] do
23                 if g[k] − max[i] + f[i,v] < θ_k then
24                     prune value v from D(PT[i]);
25 end
```



B. Hnich, R. Rossi, S. A. Tarim and S. Prestwich, **"Synthesizing Filtering Algorithms for Global Chance-Constraints"**, submitted for possible publication to the *15th International Conference on Principles and Practice of Constraint Programming* (CP-09) Lisbon, Portugal, September 21-24, 2009

Modeling Frameworks



Objective:

$$\max \left\{ \sum_{i=1}^{k} r_i \bar{X}_i - p\bar{\mathbb{E}} \left[\sum_{i=1}^{k} \mathcal{W}_i \bar{X}_i - c \right]^+ \right\}$$

Subject to:

$$\Pr \left\{ \sum_{i=1}^{k} \mathcal{W}_i X_i \leq c \right\} \geq \theta$$

Decision variables:

$X_i \in \{0,1\} \qquad \forall i \in 1,\ldots,k$

Stochastic variables:

$\mathcal{W}_i \rightarrow$ item i weight $\forall i \in 1,\ldots,k$

Stage structure:

$V_1 = \{X_1,\ldots,X_k\}$

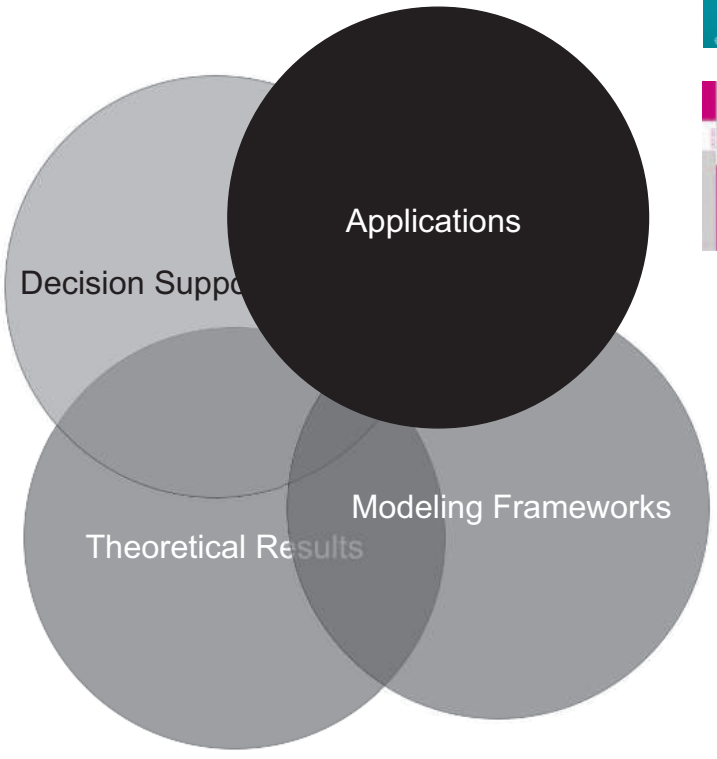
$S_1 = \{\mathcal{W}_1,\ldots,\mathcal{W}_k\}$

$L = [(V_1, S_1)]$

S. A. Tarim, S. Manandhar and T. Walsh,
"**Stochastic Constraint Programming: A Scenario-Based Approach**",
Constraints, Vol.11, pp.53-80, 2006.

Applications

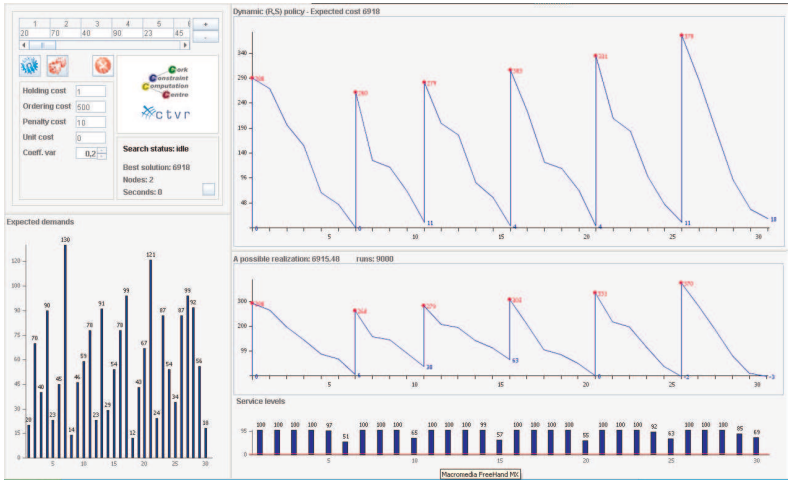
R. Rossi, S. A. Tarim, B. Hnich, S. Prestwich and S. Karacaer, "**Scheduling Internal Audit Activities: A Stochastic Combinatorial Optimization Problem**", Journal of Combinatorial Optimization

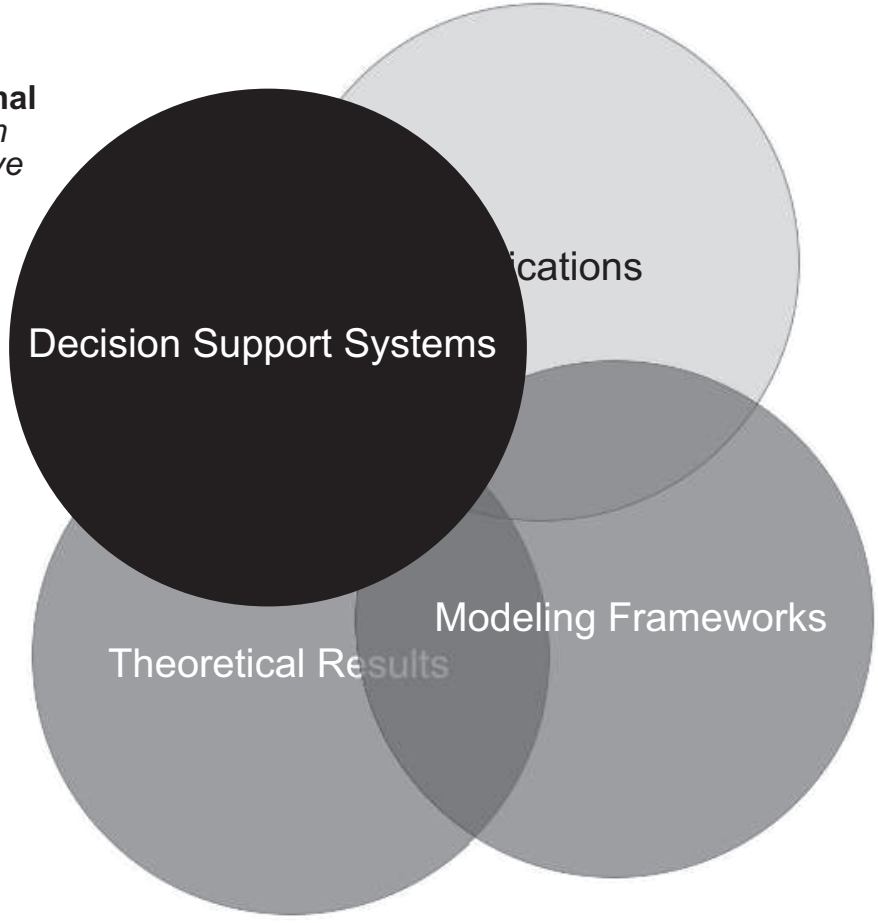


R. Rossi, S. A. Tarim, B. Hnich and S. Prestwich, "**Replenishment Planning for Stochastic Inventory Systems with Shortage Cost**", *In proceedings of The Fourth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR 07) May 23-26, 2007, Brussels, Belgium*, Lecture Notes in Computer Science, Springer-Verlag, LNCS 4510, pp.229-243, 2007

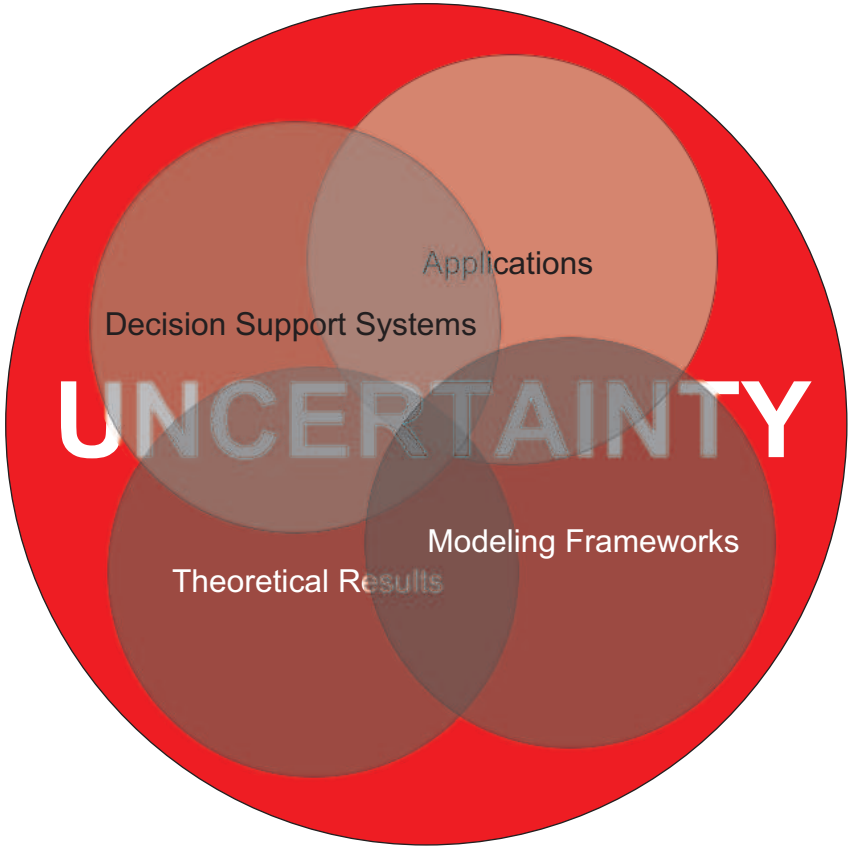
Decision Support Systems

S. A. Tarim, B. Hnich, R. Rossi and S. Prestwich, "**A Decision Support System for Computing Optimal (R,S) Policy Parameters**", *In proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science* (AICS 2008) Aug. 27, 2008, Cork, Ireland





Decision Making Under Uncertainty

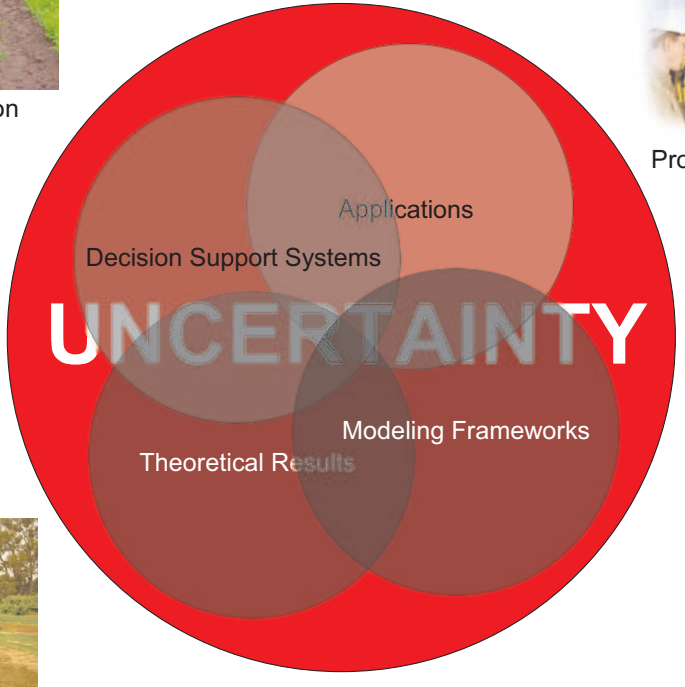


Decision Making Under Uncertainty: A Pervasive Issue

Land-Crop Allocation

Sustainable Energy Production

Food Quality Control

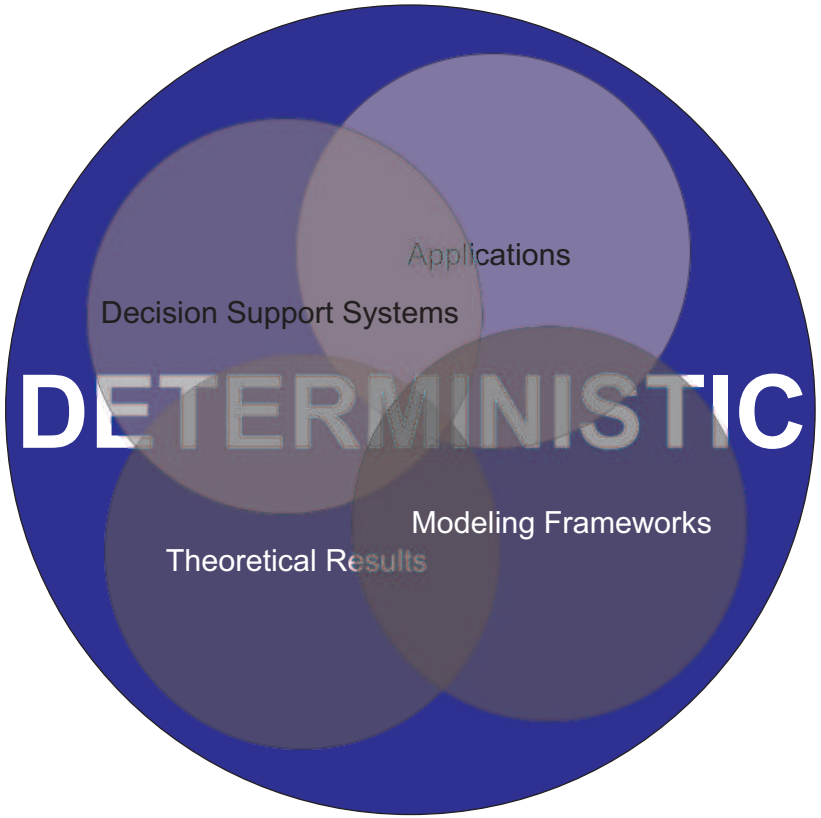


Production Planning

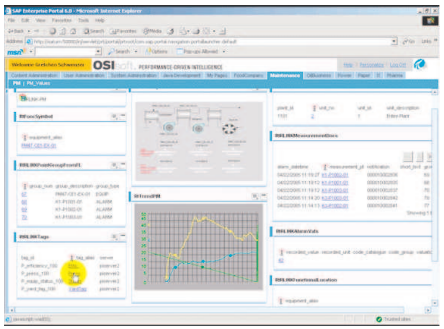
Financial Planning

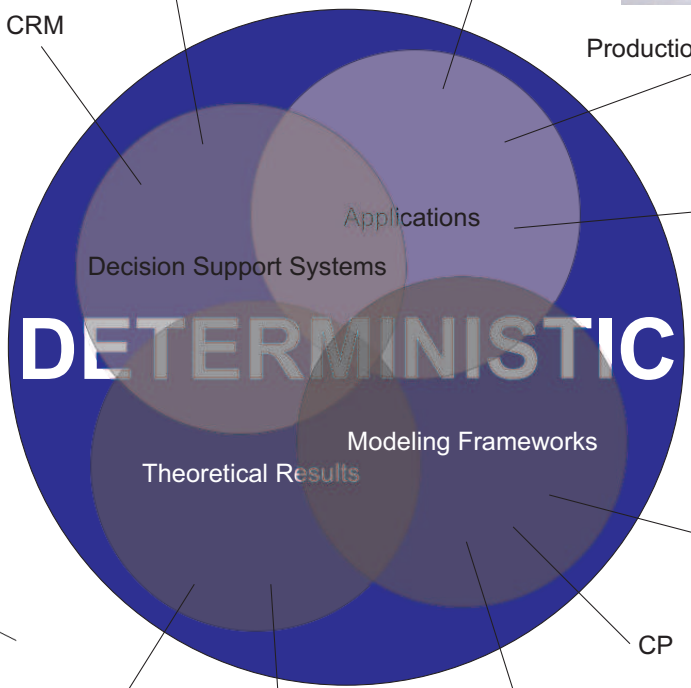
Inventory Control

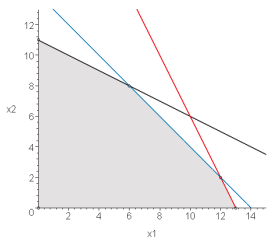
Decision Making in a Deterministic Setting



Decision Making in a Deterministic Setting

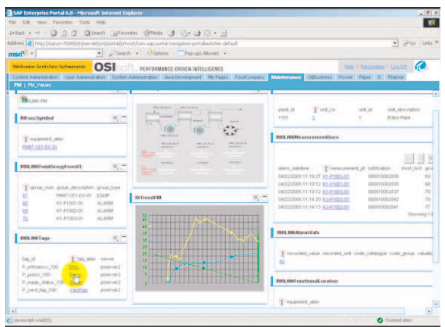


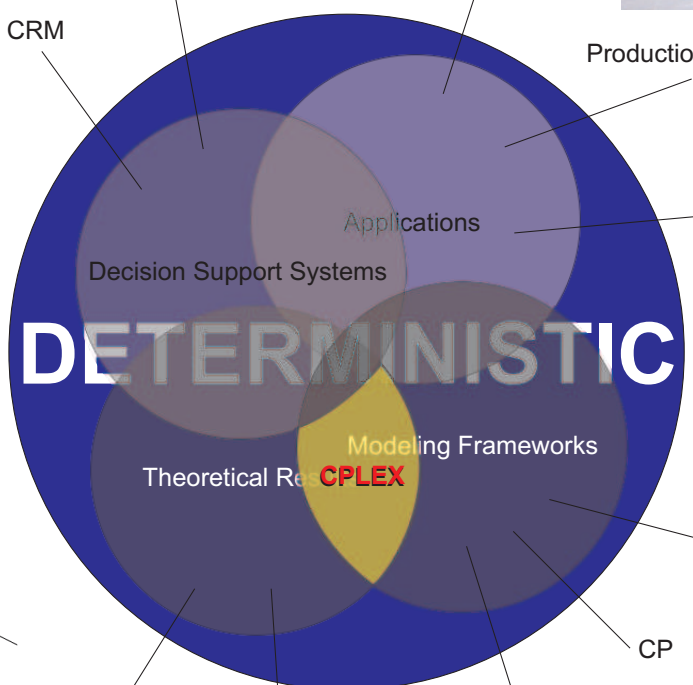


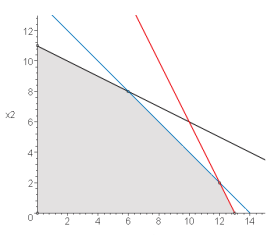


$$\begin{aligned} \text{Minimize} \quad & \sum_j c_j x_j & \\ \text{Subject to} \quad & \sum_j a_{ij} x_j \geq b_i & \forall i \\ & x_j \geq 0 & \forall j \end{aligned}$$

Decision Making in a Deterministic Setting

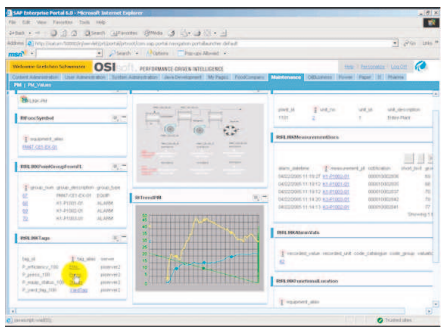


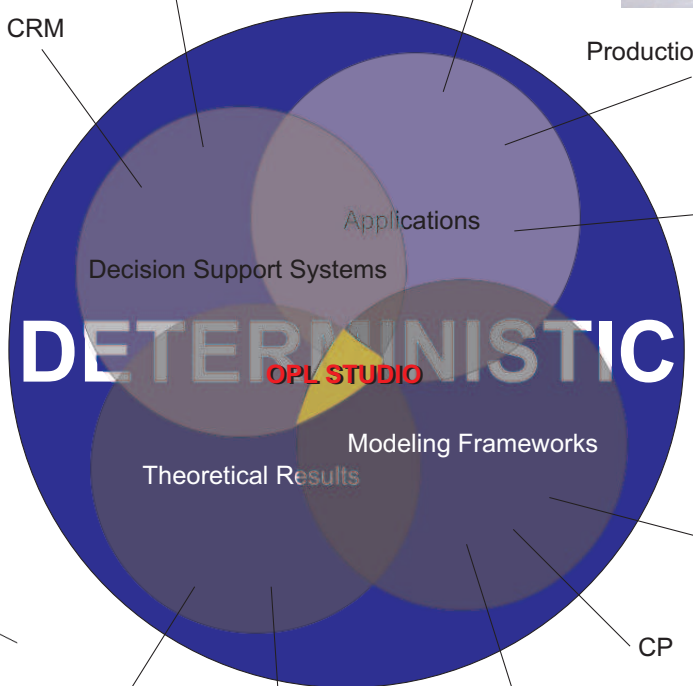


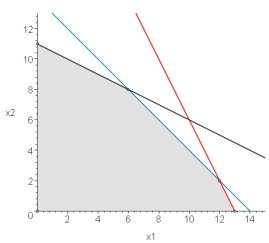


$$\begin{aligned} \text{Minimize} \quad & \sum_j c_j x_j & \\ \text{Subject to} \quad & \sum_j a_{ij} x_j \geq b_i & \forall i \\ & x_j \geq 0 & \forall j \end{aligned}$$

Decision Making in a Deterministic Setting

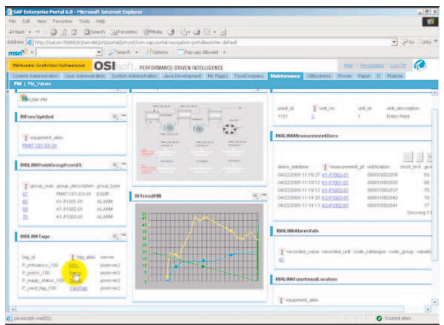


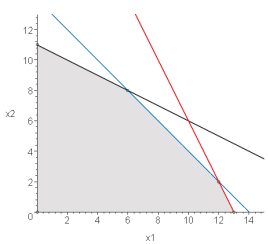




$$\begin{aligned} \text{Minimize} \quad & \sum_j c_j x_j & \\ \text{Subject to} \quad & \sum_j a_{ij} x_j \geq b_i & \forall i \\ & x_j \geq 0 & \forall j \end{aligned}$$

Decision Making under Uncertainty

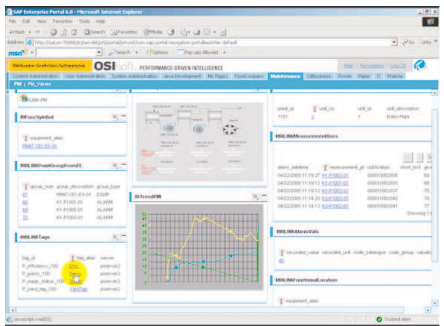


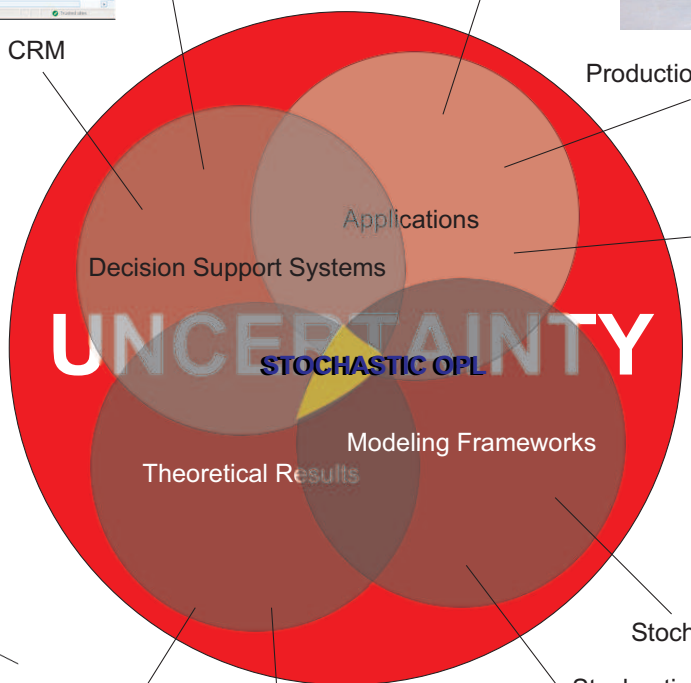


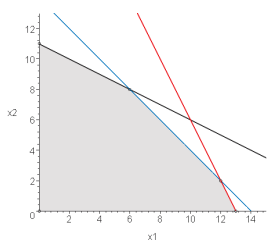
ERP
Inventory Control
CRM
Production Planning
Transport Scheduling
Applications
Decision Support Systems
UNCERTAINTY
Modeling Frameworks
Theoretical Results
Stochastic Dynamic Programming
Stochastic Programming
Simplex
Convex Analysis

$$\min_{x \in \mathcal{S}} \{g(x) \equiv \mathbb{E}_P G(x, W)\}.$$

Decision Making under Uncertainty







$$\min_{x \in \mathcal{S}} \{g(x) \equiv \mathbb{E}_P G(x, W)\}.$$

Decision Making in a Deterministic Setting

0-1 Knapsack Problem

Problem: we have k kinds of **items**, 1 through k. Each kind of item i has

- a value r_i
- a weight w_i.

We usually assume that all values and weights are nonnegative. The **maximum weight** that we can carry in the bag is W.

Objective: *find a set of objects that provides the maximum value and that fits in the given capacity.*

Decision Making in a Deterministic Setting

0-1 KP: MIP Formulation

Objective:

$\max \sum_{i=1}^{k} r_i x_i$

Constraints:

$\sum_{i=1}^{k} w_i x_i \leq W$

Decision variables:

$x_i \in \{0, 1\} \quad \forall i \in \{1, \ldots, k\}$

Decision Making under Uncertainty

Static Stochastic Knapsack Problem

Problem: we have k kinds of **items** and a **knapsack** of size c into which to fit them.

- Each item i, if included in the knapsack, brings a **deterministic profit** r_i.
- The **size** $\mathcal{W}_i$ of each item is not known at the time the decision has to be made, but we assume that the decision maker knows the **probability distribution** of $\overline{\mathcal{W}} = (\mathcal{W}_1, \ldots, \mathcal{W}_k)$.
- A per unit **penalty cost** p has to be paid for exceeding the capacity of the knapsack.
- Furthermore, the **probability of not exceeding** the capacity of the knapsack should be greater or equal to a given **threshold** θ.

Objective: *find the knapsack that maximizes the expected profit.*

Decision Making under Uncertainty

SSKP: Stochastic Programming Formulation

Objective:

$$\max \left\{ \sum_{i=1}^{k} r_i X_i - p\mathbb{E}\left[\sum_{i=1}^{k} \mathcal{W}_i X_i - c \right]^+ \right\}$$

Subject to:

$$\Pr\left\{ \sum_{i=1}^{k} \mathcal{W}_i X_i \leq c \right\} \geq \theta$$

Decision variables:

$$X_i \in \{0,1\} \qquad \forall i \in 1,\ldots,k$$

Stochastic variables:

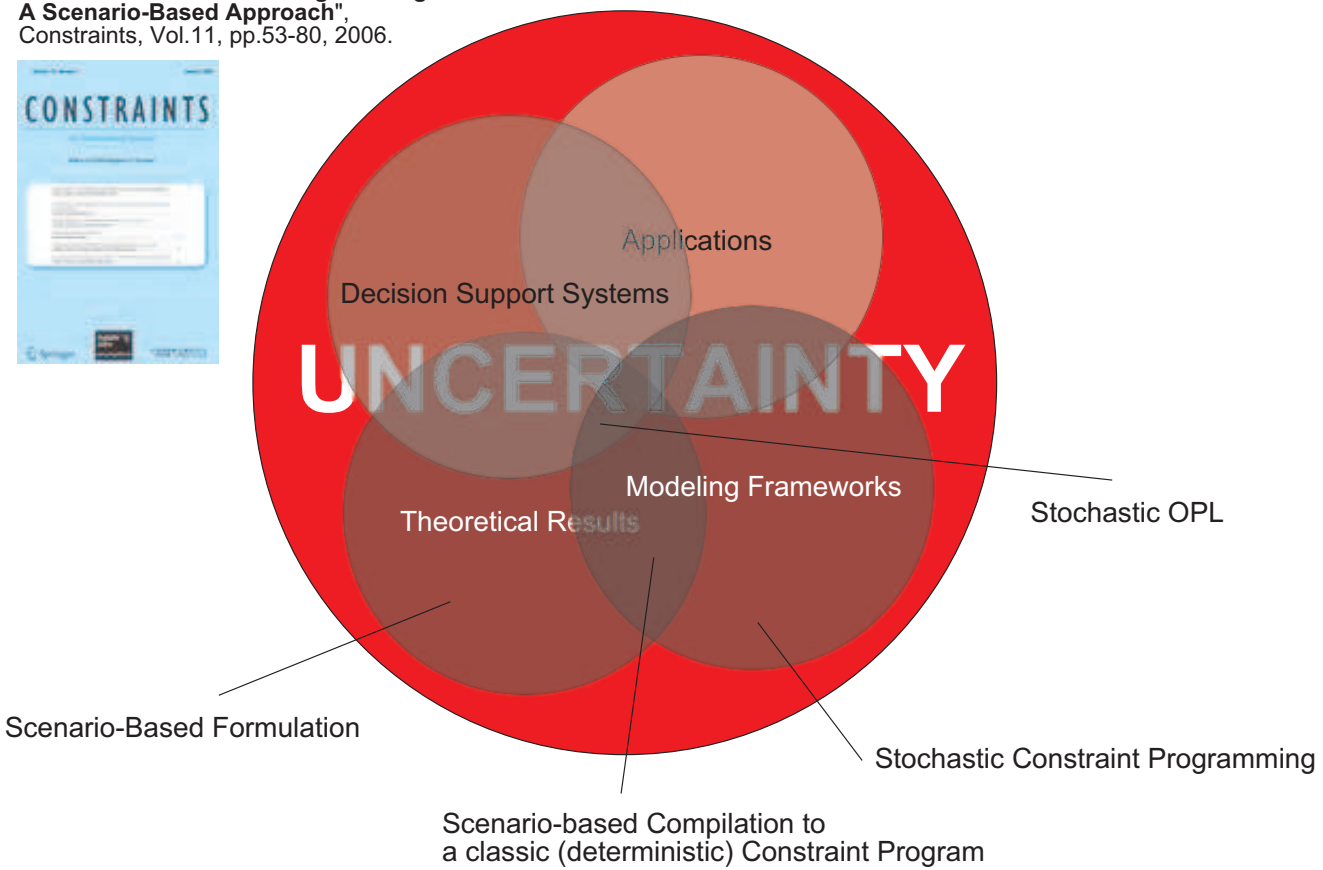
$\mathcal{W}_i \rightarrow$ item i weight $\forall i \in 1,\ldots,k$

Stage structure:

$V_1 = \{X_1,\ldots,X_k\}$
$S_1 = \{\mathcal{W}_1,\ldots,\mathcal{W}_k\}$
$L = [\langle V_1, S_1 \rangle]$

A First Step Towards Seamless Stochastic Optimization

S. A. Tarim, S. Manandhar and T. Walsh,
"**Stochastic Constraint Programming: A Scenario-Based Approach**",
Constraints, Vol.11, pp.53-80, 2006.



A First Step Towards Seamless Stochastic Optimization

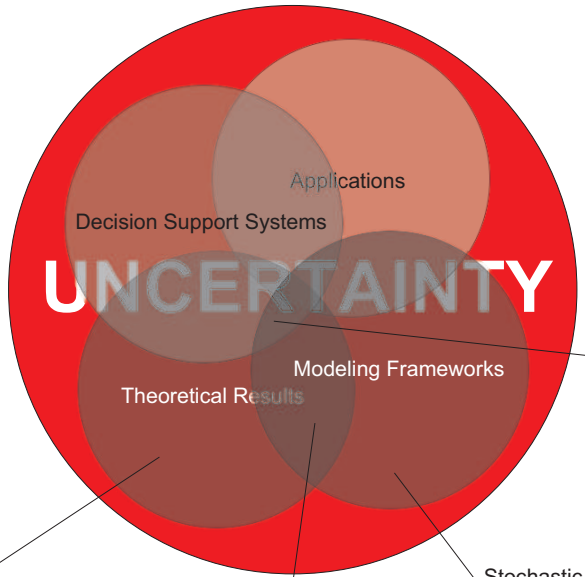
Advantages

- **Seamless** Modeling under Uncertainty!
- **Stochastic OPL** not necessarily linked to CP

Drawbacks

- **Size** of the compiled model
- **Propagation** not fully supported

A Viable Approach for Seamless Stochastic Optimization



S. A. Tarim, S. Manandhar and T. Walsh,
"**Stochastic Constraint Programming: A Scenario-Based Approach**",
Constraints, Vol.11, pp.53-80, 2006.

Stochastic OPL

T. Walsh,
"**Stochastic Constraint Programming**",
Proceedings of the European Conference on AI, 2002.

Stochastic Constraint Programming

Filtering Algorithms for
Global Chance-Constraints

Compilation using Global Chance-Constraints

B. Hnich, R. Rossi, S. A. Tarim and S. Prestwich,
"**Synthesizing Filtering Algorithms for Global Chance-Constraints**",
submitted for possible publication to the *15th International Conference on Principles and Practice of Constraint Programming* (CP-09) Lisbon, Portugal, September 21-24, 2009

Rossi, S. A. Tarim, B. Hnich and S. Prestwich,
"**A Global Chance-Constraint for Stochastic Inventory Systems under Service Level Constraints**", Constraints, Springer-Verlag, Vol. 13(4):490-517, 2008

A Viable Approach for Seamless Stochastic Optimization

Stochastic Constraint Program

Objective:

$$\max \left\{ \sum_{i=1}^{k} r_i X_i - p\mathbb{E}\left[\sum_{i=1}^{k} \mathcal{W}_i X_i - c \right]^+ \right\}$$

Subject to:

$$\Pr\left\{ \sum_{i=1}^{k} \mathcal{W}_i X_i \leq c \right\} \geq \theta$$

Decision variables:

$X_i \in \{0,1\} \qquad \forall i \in 1,\ldots,k$

Stochastic variables:

$\mathcal{W}_i \rightarrow$ item i weight $\forall i \in 1,\ldots,k$

Stage structure:

$V_1 = \{X_1,\ldots,X_k\}$

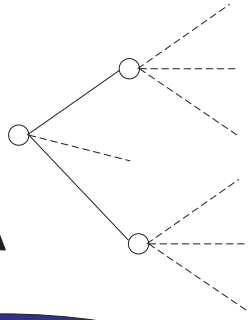
$S_1 = \{\mathcal{W}_1,\ldots,\mathcal{W}_k\}$

$L = [(V_1, S_1)]$

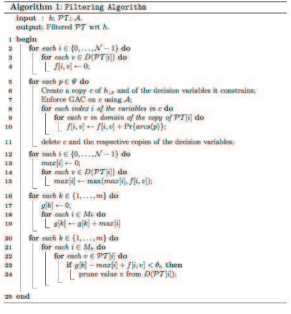
Stochastic OPL Model

```
stoch myrand[onestage]=...;
int nbItems=...;
float c = ...;
float q = ...;
range Items 1..nbItems;
range onestage 1..1;
float W[Items,onestage]^myrand = ...;
float r[Items] = ...;
dvar float+ z;
dvar int x[Items] in 0..1;

maximize
sum(i in Items) x[i]*r[i] - expected(c*z)

subject to{
z >= sum(i in Items) W[i]*x[i] - q;
prob(sum(i in Items) W[i]*x[i] <= q) >= 0.6;
};
```

Solution



Constraint Programming Solver supporting Global Chance-Constraints



Filtering Algorithms for Global Chance-Constraints

A Viable Approach for Seamless Stochastic Optimization

In what follows we shall discuss:

- Constraint Programming
- Stochastic Constraint Programming
- Global Chance-Constraints
- Stochastic OPL.

During the discussion the **SSKP** will be employed as running example.

Introduction

"Constraint programming (CP) represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the **user** states the problem, the **computer** solves it".
Eugene C. Freuder, Constraints, April 1997

Formal Background

A slightly formal definition

A **Constraint Satisfaction Problem** (CSP) is a triple $\langle V, D, C \rangle$.

- $V = \{v_1, \ldots, v_n\}$ is a set of variables
- D is a function mapping each variable v_i to a domain $D(v_i)$ of values
- C is a set of constraints.

A **Constraint Optimization Problem** (COP) consists of a CSP and objective function $f(\widehat{V})$ defined on a subset $\widehat{V}$ of the decision variables in V. The aim in a COP is to find a feasible solution that minimizes (maximizes) the objective function.

An Example

Sample COP: 0-1 KP

- $V = \{x_1, \ldots, x_3\}$
- $D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
- $C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
- $f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

The **optimal solution** for the COP has a value of 25 and corresponds to the assignment is $x_2 = x_3 = 1$ and $x_1 = 0$.

An Example

Sample COP: 0-1 KP

Objects:

r=8 w=8	r=15 w=5	r=10 w=4

Knapsack:

Capacity: 10

An Example

Sample COP: 0-1 KP

Decision:

```
r=8
w=8
```

```
r=15
w=5
```

```
r=10
w=4
```

Knapsack:

```
Capacity: 10
Required: 9
```

Solution Method

Strategy

- **Constraint Programming** proposes to solve CSPs/COPs by associating with each constraint a **filtering algorithm**.
- A **filtering algorithm** removes from decision variable domains **values that cannot belong** to any solution of the CSP/COP.
- **Constraint Propagation** is the **process** that repeatedly calls filtering algorithms until no new deduction can be made.
- **Constraint Solving** interleaves **filtering algorithms** and a **search procedure** (for instance a backtracking algorithm).

An Example

Sample COP: 0-1 KP

- $V = \{x_1, \ldots, x_3\}$
- $D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
- $C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
- $f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

Constraint Propagation initially does not produce **any effect**.

An Example

Sample COP: 0-1 KP

- $V = \{x_1, \ldots, x_3\}$
- $D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
- $C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
- $f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

To build a search tree, we apply the **lexicographic variable and value ordering heuristics** and use **labeling as domain splitting procedure**.

An Example

P_0

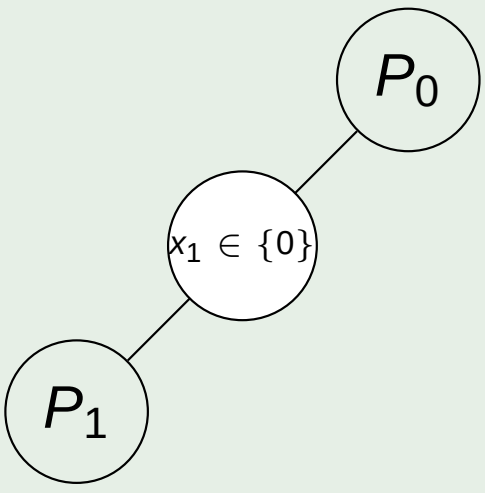
Sample COP: 0-1 KP

$V = \{x_1, \ldots, x_3\}$
$D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
$C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
$f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

Filtered domains at P_0

- $D(x_1) = \{0, 1\}$ $D(x_2) = \{0, 1\}$ $D(x_3) = \{0, 1\}$
  $D(z) = \{0, \ldots, 33\}$

An Example



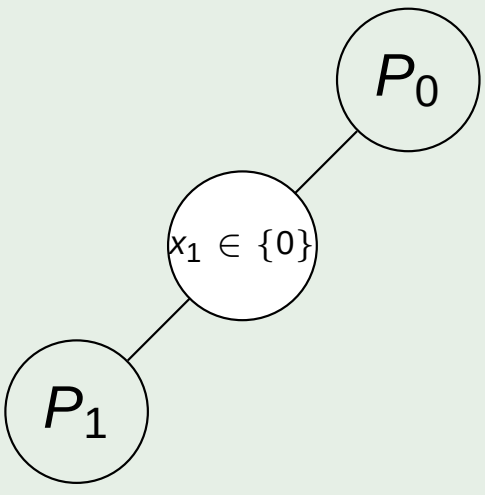
Sample COP: 0-1 KP

$V = \{x_1, \ldots, x_3\}$
$D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
$C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
$f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

Filtered domains at P_1

- $D(x_1) = \{0\}$ $D(x_2) = \{0, 1\}$ $D(x_3) = \{0, 1\}$
  $D(z) = \{0, \ldots, 25\}$

An Example



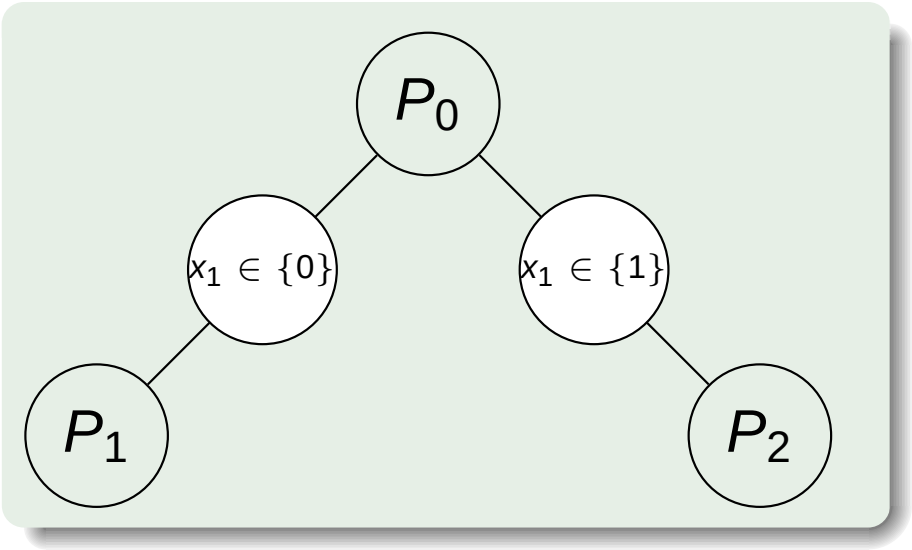
Sample COP: 0-1 KP

$V = \{x_1, \ldots, x_3\}$
$D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
$C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
$f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

Optimal solution associated with P_1

- $D(x_1) = \{0\}$ $D(x_2) = \{1\}$ $D(x_3) = \{1\}$ $D(z) = \{25\}$

An Example



Sample COP: 0-1 KP

$V = \{x_1, \ldots, x_3\}$
$D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
$C = \{8x_1 + 5x_2 + 4x_3 \leq 10\}$
$f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3$

Filtered domains at P_2

- $D(x_1) = \{1\}$ $D(x_2) = \{0\}$ $D(x_3) = \{0\}$ $D(z) = \{8\}$

We backtrack to P_0 and we return the optimal solution with value 25 found in the subtree associated with node P_1.

Global Constraints

Efficiency

Filtering algorithms

- **detect inconsistencies** in a proactive fashion
- **speed up** the search

provided that the time spent in filtering is less then the time saved in terms of search efforts.
A challenging research topic is the **design** of efficient filtering strategies.

Global Constraints

Not only binary relations

In constraint programming is common to find constraints over a **non-predefined number of variables**

- `alldifferent`
- `element`
- `cumulative`
- ...

These constraints are called **global constraints**

- they can be used in a variety of situations
- they are associated with powerful **filtering strategies**
- new **custom** global constraints can be defined

A slightly formal definition

Stochastic Constraint Satisfaction Problem

A Stochastic Constraint Satisfaction Problem (SCSP) is a 7-tuple

$$\langle V, S, D, P, C, \theta, L \rangle.$$

- $V = \{v_1, \ldots, v_n\}$ is a set of decision variables
- $S = \{s_1, \ldots, s_n\}$ is a set of stochastic variables
- D is a function mapping each variable to a domain of potential values
- P is a function mapping each variable in S to a probability distribution for its associated domain
- C is a set of (chance)-constraints, possibly involving stochastic variables
- θ_h is a threshold probability associated to chance-constraint h
- $L = [\langle V_1, S_1 \rangle, \ldots, \langle V_i, S_i \rangle, \ldots, \langle V_m, S_m \rangle]$ is a list of *decision stages*.

By considering an objective function $f(\widehat{V}, \widehat{S})$ we obtain a SCOP.

An Example

Sample SCOP: SSKP

- $V = \{x_1, \ldots, x_3\}$
- $D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
- $S = \{w_1, \ldots, w_3\}$
- $D(w_1) = \{5(0.5), 8(0.5)\}, D(w_2) = \{3(0.5), 9(0.5)\}, D(w_3) = \{15(0.5), 4(0.5)\}$
- $C = \{Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2\}$
- $L = [\langle V, S \rangle]$
- $f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3 - 2\mathbb{E}\max\left[0, \sum_{i=1}^{3} w_i x_i - 10\right]$

Sample SCOP: SSKP

Objects:

r=8 w=5 OR 8	r=15 w=3 OR 9	r=10 w=15 OR 4

Knapsack:

Capacity: 10

Sample SCOP: SSKP

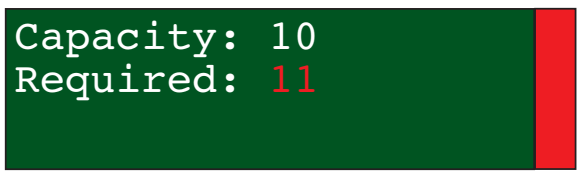
Decision:

r=8 w=5 OR 8	r=15 w=3 OR 9	r=10 w=15 OR 4
Y	Y	N

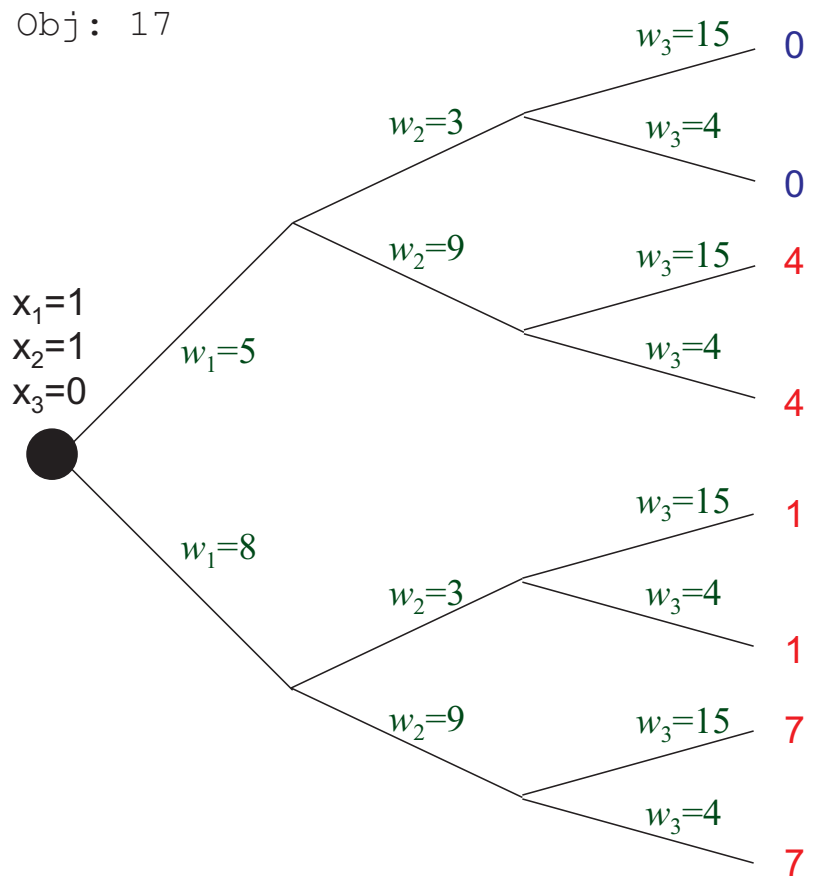
Observation:

r=8 w=8	r=15 w=3	r=10 w=15
Y	Y	N

Knapsack:



Sample SCOP: SSKP



An Example

Sample SCOP: DSKP

- $V = \{x_1, \ldots, x_3\}$
- $D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\}$
- $S = \{w_1, \ldots, w_3\}$
- $D(w_1) = \{5(0.5), 8(0.5)\}, D(w_2) = \{3(0.5), 9(0.5)\}, D(w_3) = \{15(0.5), 4(0.5)\}$
- $C = \{Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2\}$
- $L = [\langle\{x_1\}, \{w_1\}\rangle, \langle\{x_2\}, \{w_2\}\rangle, \langle\{x_3\}, \{w_3\}\rangle]$
- $f(x_1, \ldots, x_3) = \mathbb{E}[8x_1 + 15x_2 + 10x_3] - 2\mathbb{E}\max\left[0, \sum_{i=1}^{3} w_i x_i - 10\right]$

Sample SCOP: DSKP

Objects:

r=8 w=5 OR 8	r=15 w=3 OR 9	r=10 w=15 OR 4

Knapsack:

```
Capacity: 10
```

Sample SCOP: DSKP

Decision stage 1:

r=8 w=5 OR 8	r=15 w=3 OR 9	r=10 w=15 OR 4

Y

Observation:

r=8 w=8

Knapsack:

```
Capacity: 10
Required: 8
```

Sample SCOP: DSKP

Decision stage 2:

```
r=8
w=8
```

```
r=15
w=3 OR 9
```

Y

```
r=10
w=15 OR 4
```

Observation:

```
r=15
w=3
```

Knapsack:

```
Capacity: 10
Required: 11
```

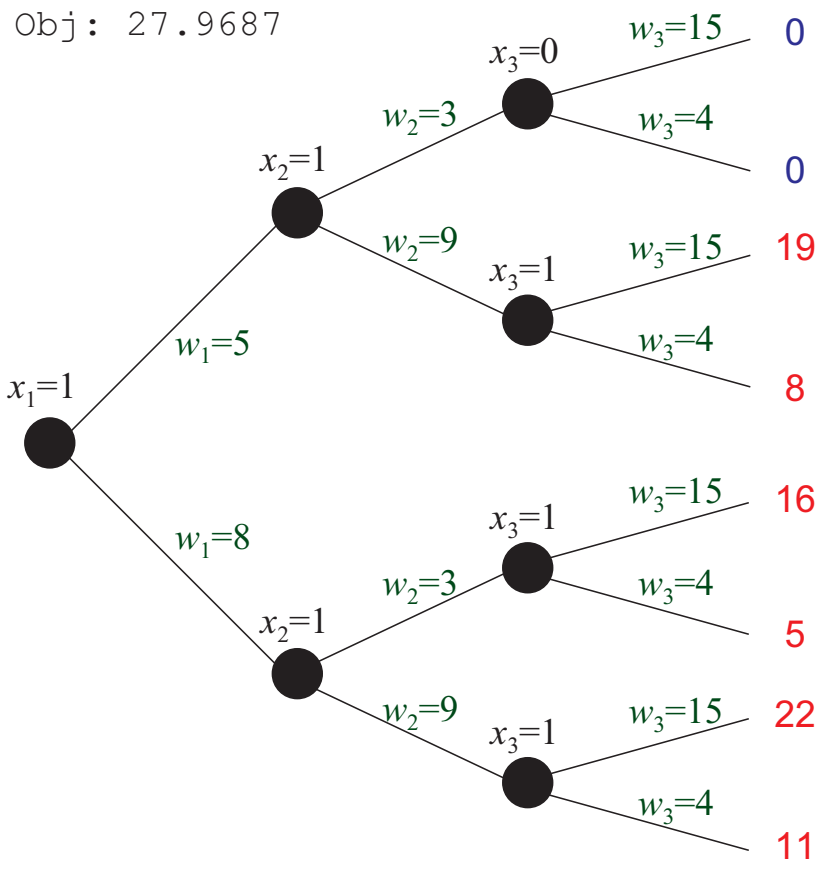
Sample SCOP: DSKP

Decision stage 3:

Observation:

Knapsack:

Sample SCOP: DSKP



Scenario-based Compilation

By using the approach discussed in

S. A. Tarim, S. Manandhar and T. Walsh,
Stochastic Constraint Programming: A Scenario-Based Approach,
Constraints, Vol.11, pp.53-80, 2006

it is possible to compile any SCSP/SCOP down to a deterministic equivalent CSP.

Scenario-based Compilation

SSKP: Compiled Deterministic Equivalent CSP

```
int nbWorlds=8;
range Worlds 1..nbWorlds;
int nbItems=3;
range Items 1..nbItems;
float c = 2;
float W[Worlds,Items] =[[5,3,15],
                        [5,3,4],
                        [5,9,15],
                        [5,9,4],
                        [8,3,15],
                        [8,3,4],
                        [8,9,15],
                        [8,9,4]
                        ];
float Pr[Worlds]=
[0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125];

float r[Items] = [8,15,10];
float q = 10;

var float+ z[Worlds];
var int+ x[Items] in 0..1;

maximize ((sum(i in Items)x[i]*r[i])-c*(sum(j in Worlds)Pr[j]*z[j]))

subject to{
   forall(j in Worlds) z[j]>=(sum(i in Items)W[j,i]*x[i])-q;
   sum(j in Worlds) Pr[j]*(sum(i in Items)W[j,i]*x[i] <= q) >= 0.2;
};
```

An Alternative Compilation Strategy Employing GCCs

SSKP: Compiled Deterministic Equivalent CSP with Global Chance-Constraints

```
int nbWorlds=8;
range Worlds 1..nbWorlds;
int nbItems=3;
range Items 1..nbItems;
float c = 2;
float W[Worlds,Items] =[[5,3,15],
                        [5,3,4],
                        [5,9,15],
                        [5,9,4],
                        [8,3,15],
                        [8,3,4],
                        [8,9,15],
                        [8,9,4]
                        ];
float Pr[Worlds]=
[0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125];

float r[Items] = [8,15,10];
float q = 10;

var float+ z;
var int+ x[Items] in 0..1;

maximize ((sum(i in Items)x[i]*r[i])-c*(max(0,z-q)))

subject to{
   stochLinIneq(x,W,Pr,q,0.2);
   expectedLinEq(x,W,Pr,z);
};
```

Filtering in SCSPs

Also in Stochastic Constraint Programming (SCP) we have

- constraints
- filtering algorithms

In contrast to CP, in SCP constraints divide into

- hard constraints
- chance-constraints

Global Chance-Constraints

Perhaps the most interesting aspect of SCP is that the concept of **global constraint** can be also adopted in a stochastic environment, thus leading to

- Global Chance-Constraints (Rossi et al., 2008)

Filtering in SCSPs

Stochastic Constraint Programming

Global Chance-Constraints

- represent relations among a non-predefined number of **decision** and **random** variables
- implement dedicated filtering algorithms based on
  - **feasibility** reasoning
  - **optimality** reasoning

Global Chance-Constraints performing optimality reasoning are called **Optimization-Oriented Global Chance-Constraints** (Rossi et al., 2008).

Filtering in SCSPs

"Synthesizing Filtering Algorithms for Global Chance-Constraints" (Hnich et al., 2009)

Algorithm 1: Filtering Algorithm

input : h; $\mathcal{PT}$; $\mathcal{A}$.
output: Filtered $\mathcal{PT}$ wrt h.

```
1  begin
2      for each i ∈ {0,...,N−1} do
3          for each v ∈ D(PT[i]) do
4              f[i,v] ← 0;
5      for each p ∈ Ψ do
6          Create a copy c of h_p and of the decision variables it constrains;
7          Enforce GAC on c using A;
8          for each index i of the variables in c do
9              for each v in domain of the copy of PT[i] do
10                 f[i,v] ← f[i,v] + Pr{arcs(p)};
11         delete c and the respective copies of the decision variables;
12     for each i ∈ {0,...,N−1} do
13         max[i] ← 0;
14         for each v ∈ D(PT[i]) do
15             max[i] ← max(max[i], f[i,v]);
16     for each k ∈ {1,...,m} do
17         g[k] ← 0;
18         for each i ∈ M_k do
19             g[k] ← g[k] + max[i]
20     for each k ∈ {1,...,m} do
21         for each i ∈ M_k do
22             for each v ∈ PT[i] do
23                 if g[k] − max[i] + f[i,v] < θ_k then
24                     prune value v from D(PT[i]);
25 end
```

Filtering Algorithms for GCCs

Stochastic Programming Model

$$\Pr\left\{\sum_{i=1}^{k} \mathcal{W}_i X_i \leq c\right\} \geq \theta$$

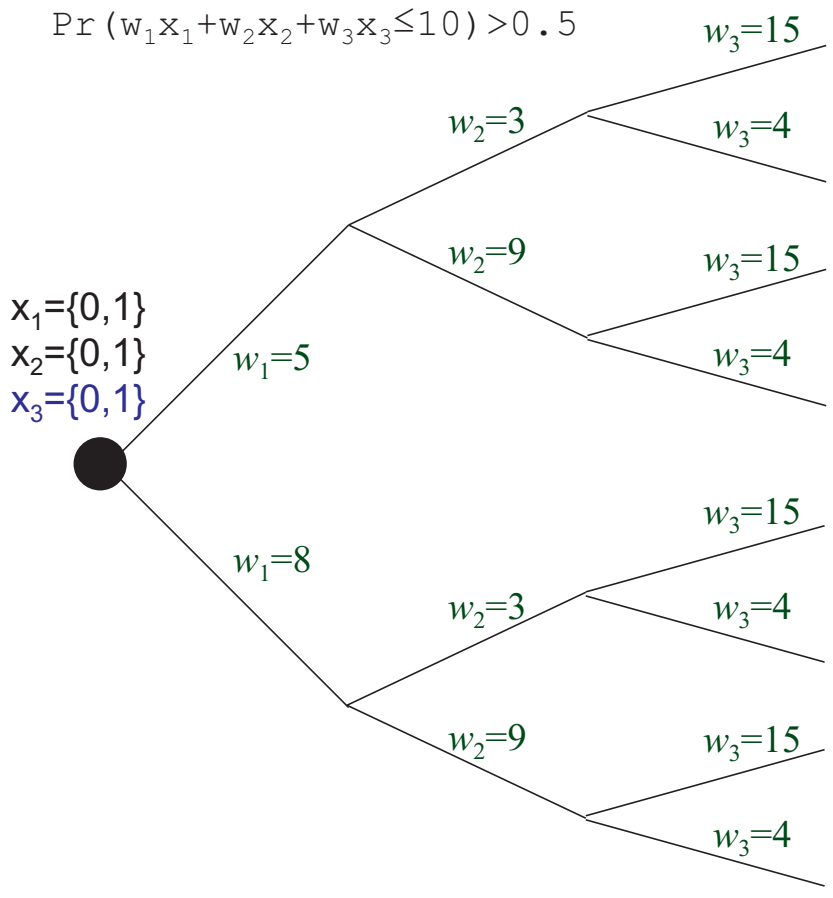
Stochastic Constraint Programming Model

$$C = \{Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2\}$$

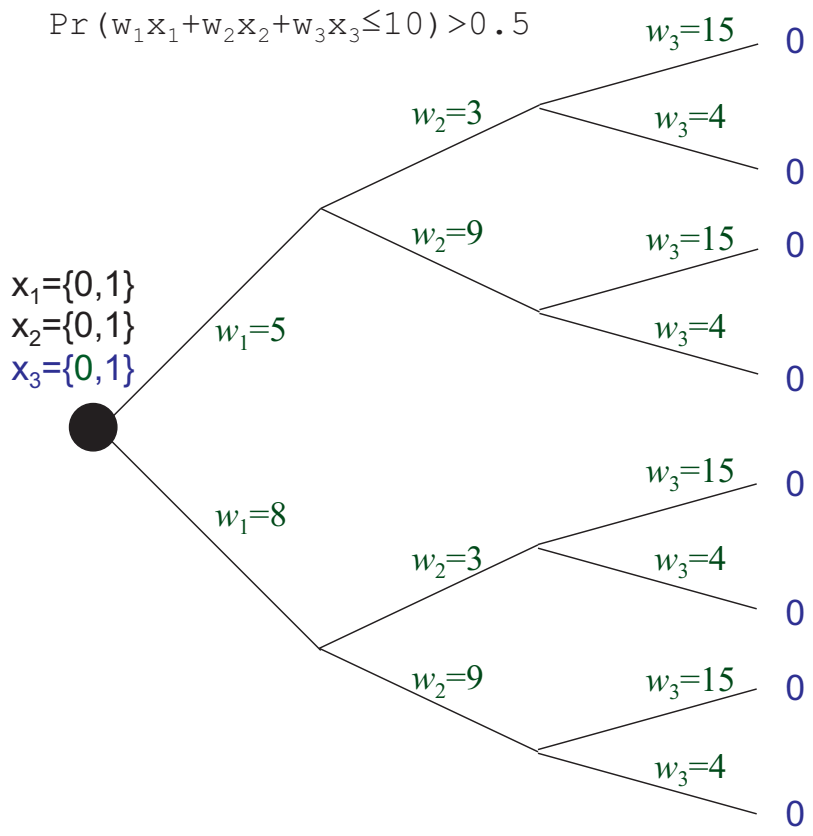
Global Chance-Constraint

```
stochLinIneq(x,W,Pr,q,0.2);
```

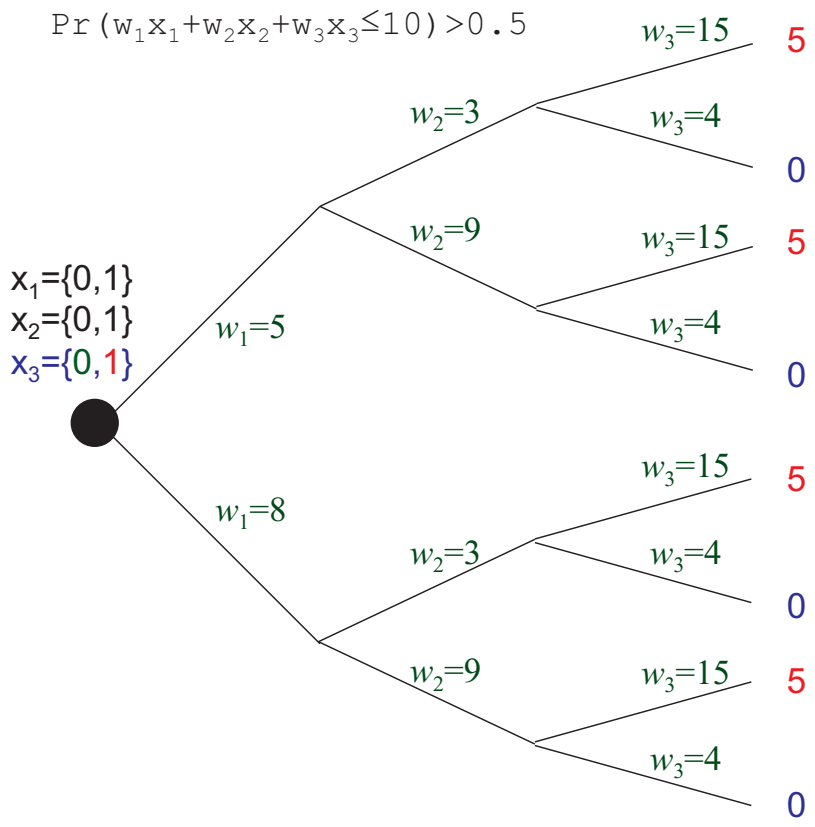
Filtering Algorithms for GCCs



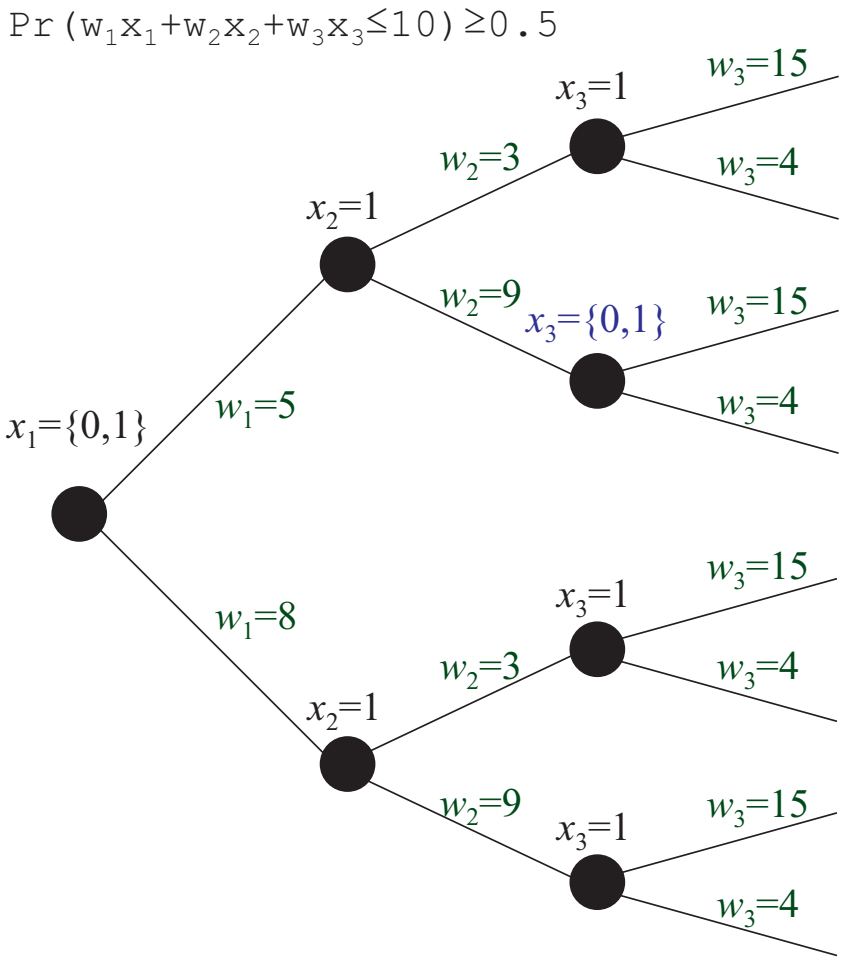
Filtering Algorithms for GCCs



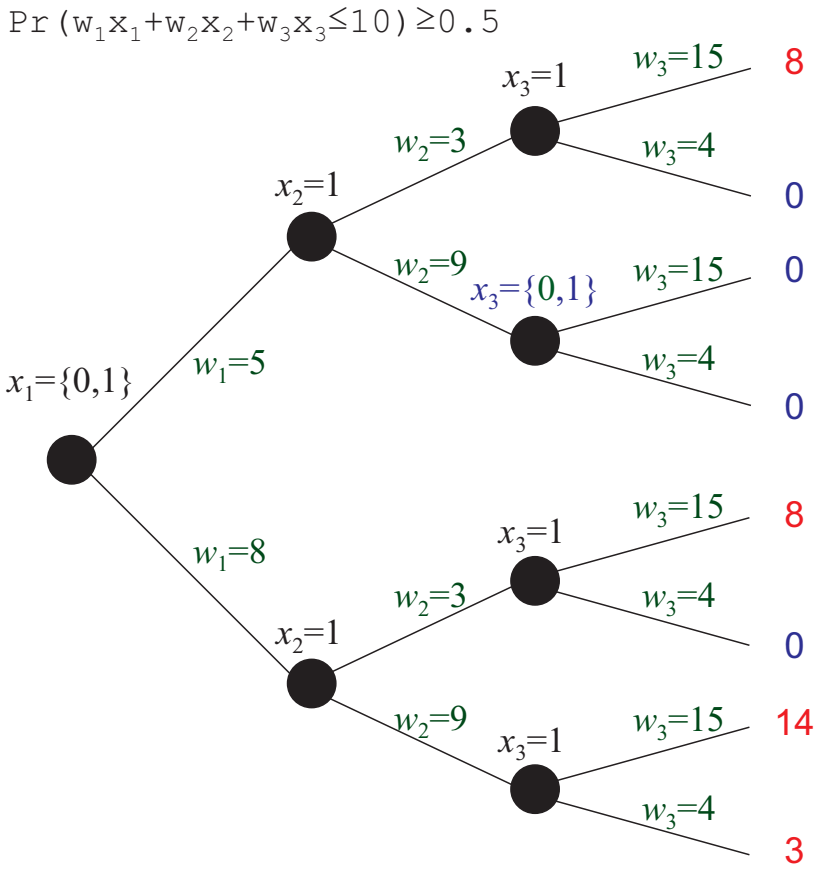
Filtering Algorithms for GCCs



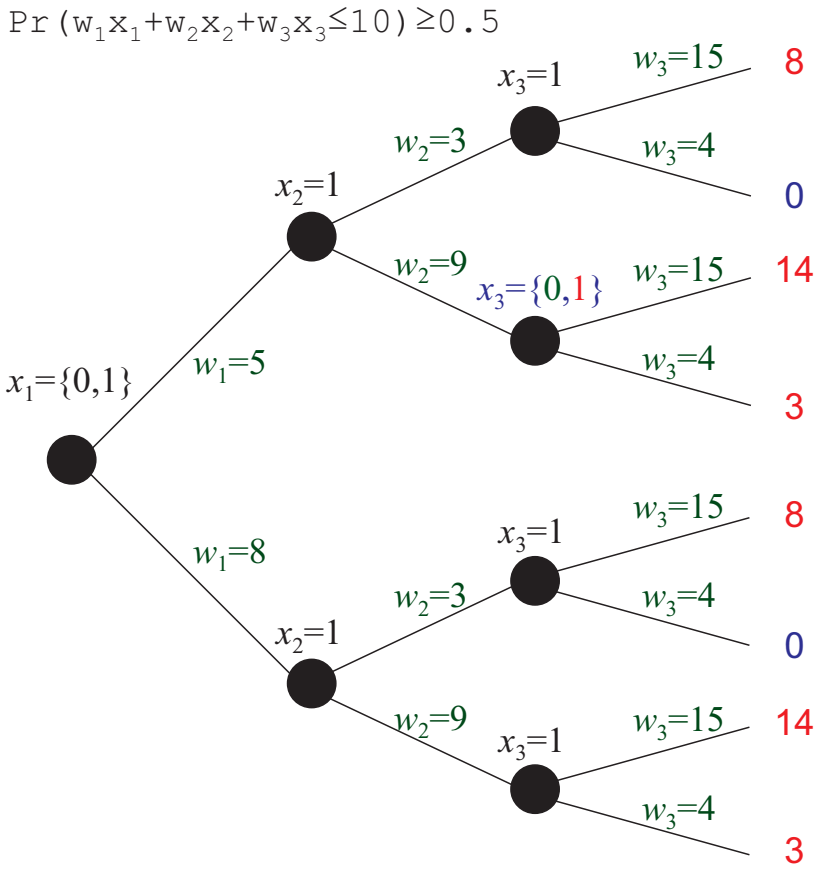
Filtering Algorithms for GCCs



Filtering Algorithms for GCCs



Filtering Algorithms for GCCs



Stochastic OPL

A **language** specifically introduced by Tarim et al. (Tarim et al., 2006) for **modeling decision problems under uncertainty**. It captures several **high level concepts** that facilitate the process of modeling uncertainty:

- stochastic variables (independent or conditional distributions)
- several probabilistic measures for the objective function (expectation, variance, etc.)
- chance-constraints
- decision stages
- ...

Stochastic OPL

SSKP

```
int N = 3;
int c = 10;
int p = 2;
float θ = 0.2
range Object [1..3];
int value[Object] = [8,15,10];
stoch int weight[Object] = [<5(0.5),8(0.5)>,
                    <3(0.5),9(0.5)>,<15(0.5),4(0.5)>];
var int+ X[Object] in 0..1;
stages = [<X,weight>];
var int+ z;

maximize sum(i in Object) X[i]*value[i] - p*z
subject to{
z = max(0,expected(sum(i in Object) X[i]*weight[i] - c));
prob(sum(i in Object) X[i]*weight[i] - c ≤ 0) ≥ θ;
};
```

Stochastic OPL

DSKP

```
int N = 3;
int c = 10;
int p = 2;
float θ = 0.2
range Object [1..3];
int value[Object] = [8,15,10];
stoch int weight[Object] = [<5(0.5),8(0.5)>,
                      <3(0.5),9(0.5)>,<15(0.5),4(0.5)>];
var int+ X[Object] in 0..1;
stages = [<X[1],weight[1]>,<X[2],weight[2]>,<X[3],weight[3]>];
var int+ z;

maximize sum(i in Object) X[i]*value[i] - p*z
subject to{
z = max(0,expected(sum(i in Object) X[i]*weight[i] - c));
prob(sum(i in Object) X[i]*weight[i] - c ≤ 0) ≥ θ;
};
```

A Viable Approach for Seamless Stochastic Optimization

Stochastic Constraint Program

Objective:

$$\max \left\{ \sum_{i=1}^{k} r_i X_i - p\mathbb{E}\left[\sum_{i=1}^{k} \mathcal{W}_i X_i - c \right]^+ \right\}$$

Subject to:

$$\Pr\left\{ \sum_{i=1}^{k} \mathcal{W}_i X_i \leq c \right\} \geq \theta$$

Decision variables:

$X_i \in \{0,1\} \qquad \forall i \in 1,\ldots,k$

Stochastic variables:

$\mathcal{W}_i \rightarrow$ item i weight $\forall i \in 1,\ldots,k$

Stage structure:

$V_1 = \{X_1,\ldots,X_k\}$

$S_1 = \{\mathcal{W}_1,\ldots,\mathcal{W}_k\}$

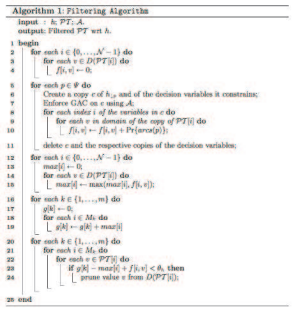
$L = [(V_1, S_1)]$

Stochastic OPL Model

```
stoch myrand[onestage]=...;
int nbItems=...;
float c = ...;
float q = ...;
range Items 1..nbItems;
range onestage 1..1;
float W[Items,onestage]^myrand = ...;
float r[Items] = ...;
dvar float+ z;
dvar int x[Items] in 0..1;

maximize
sum(i in Items) x[i]*r[i] - expected(c*z)

subject to{
z >= sum(i in Items) W[i]*x[i] - q;
prob(sum(i in Items) W[i]*x[i] <= q) >= 0.6;
};
```

Solution

Constraint Programming Solver supporting Global Chance-Constraints



Filtering Algorithms for Global Chance-Constraints

B. Hnich, R. Rossi, S. A. Tarim and S. Prestwich, **Synthesizing Filtering Algorithms for Global Chance-Constraints**, submitted for possible publication to the 15th International Conference on Principles and Practice of Constraint Programming (CP-09) Lisbon, Portugal, September 21-24, 2009

Contribution

A generic approach for **constraint reasoning under uncertainty**.
Works with any existing propagation algorithm!

B. Hnich, R. Rossi, S. A. Tarim and S. Prestwich, **Synthesizing Filtering Algorithms for Global Chance-Constraints**, submitted for possible publication to the 15th International Conference on Principles and Practice of Constraint Programming (CP-09) Lisbon, Portugal, September 21-24, 2009

Drawback

Only implemented for linear inequalities/equalities:

```
stochLinIneq(x,W,Pr,q,0.2);
```

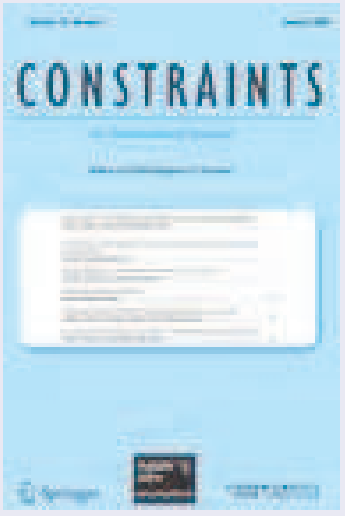
i.e. SSKP $\rightarrow$ $\Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2$

B. Hnich, R. Rossi, S. A. Tarim and S. Prestwich, **Synthesizing Filtering Algorithms for Global Chance-Constraints**, submitted for possible publication to the 15th International Conference on Principles and Practice of Constraint Programming (CP-09) Lisbon, Portugal, September 21-24, 2009

Future work

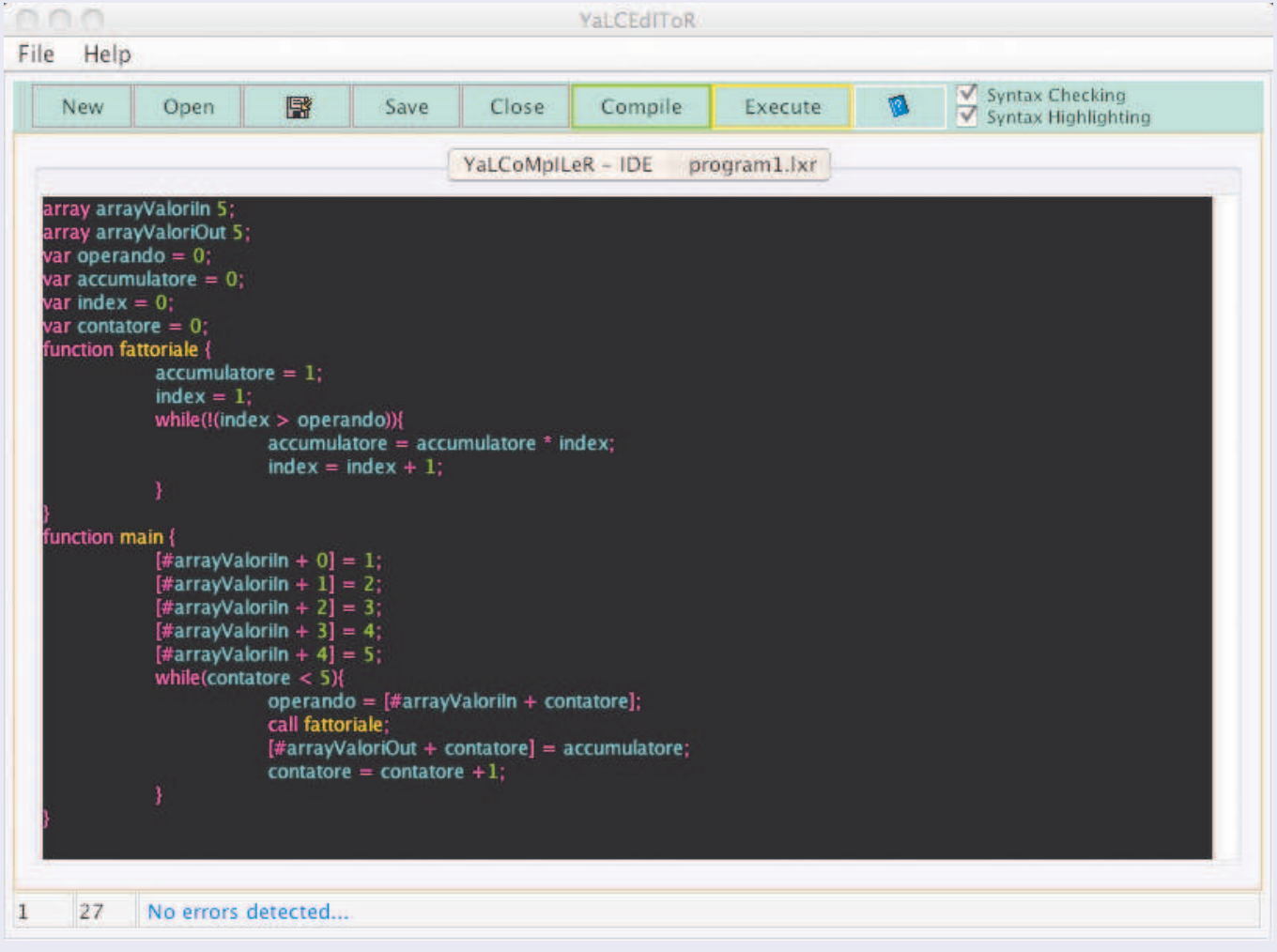
Considering more **global constraints**:

`allDifferent()`
`NValue()`
`Cumulative()`
...



S. A. Tarim, S. Manandhar and T. Walsh,
Stochastic Constraint Programming: A Scenario-Based Approach,
Constraints, Vol.11, pp.53-80, 2006

Language Compiler

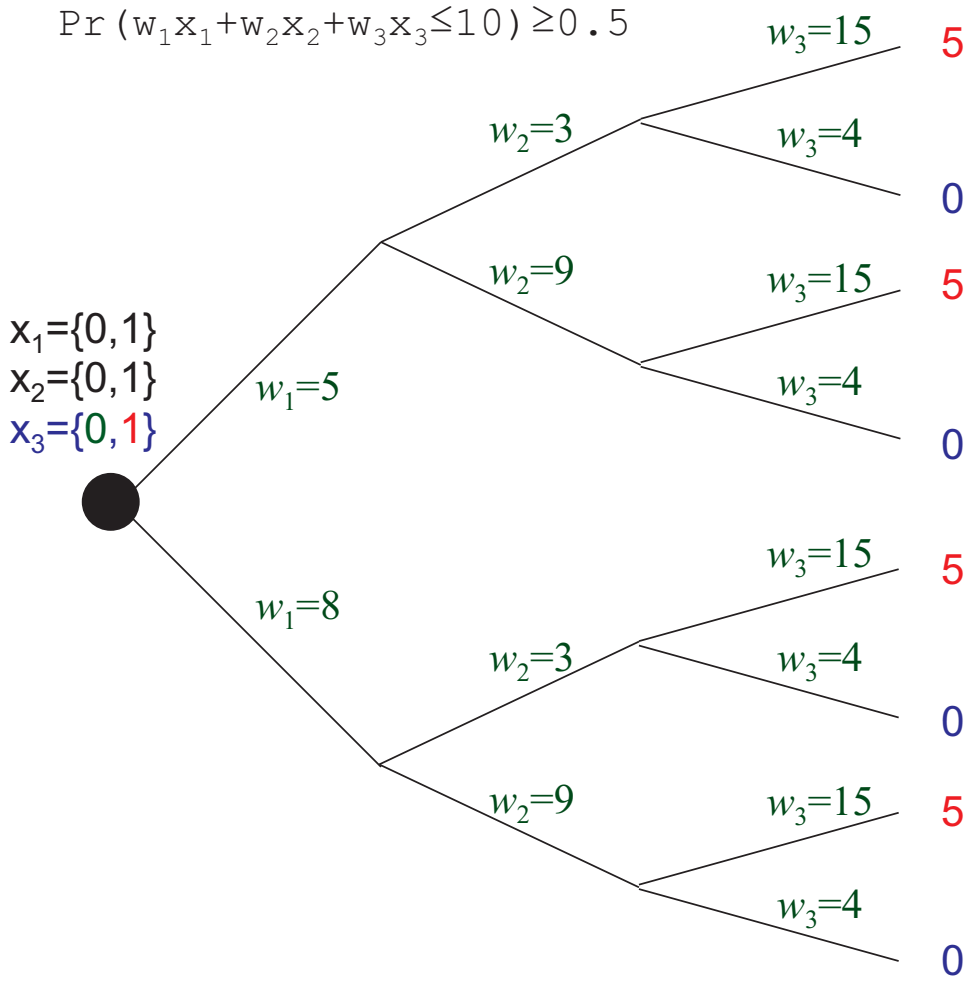


Scenario Reduction

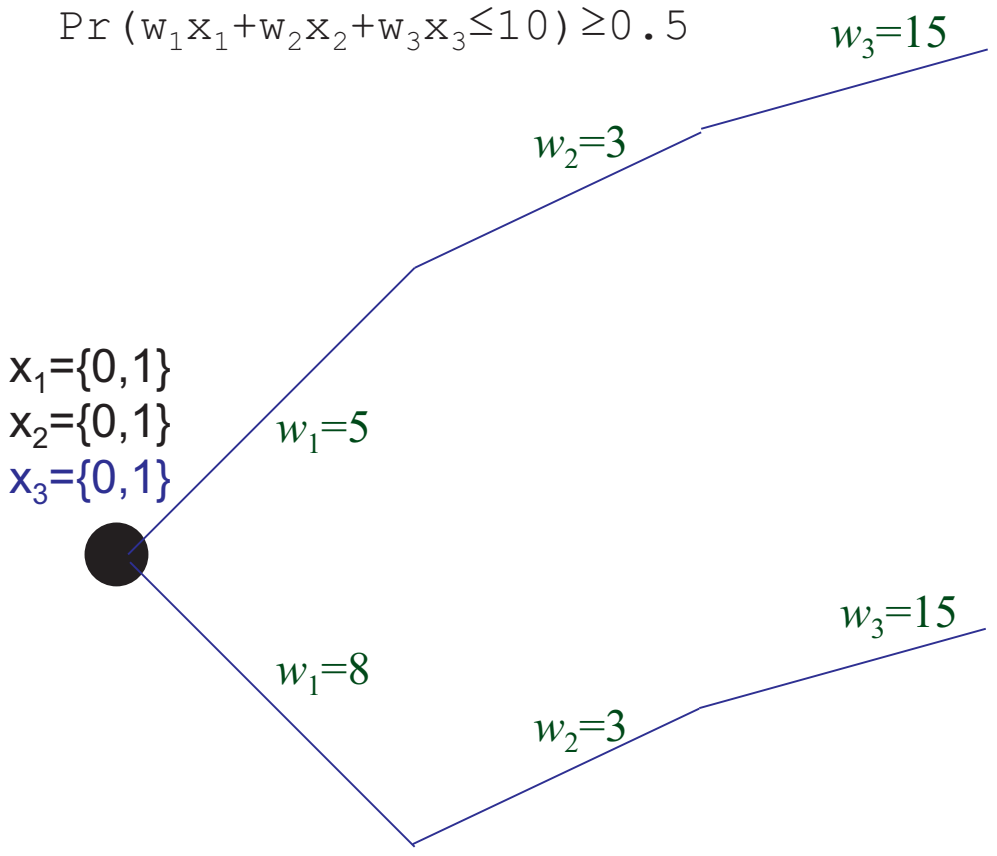
Inspired by:

A. J. Kleywegt, A. Shapiro, T. Homem-De-Mello
The Sample Average Approximation Method for Stochastic Discrete Optimization
SIAM Journal of Optimization, 12:479–502

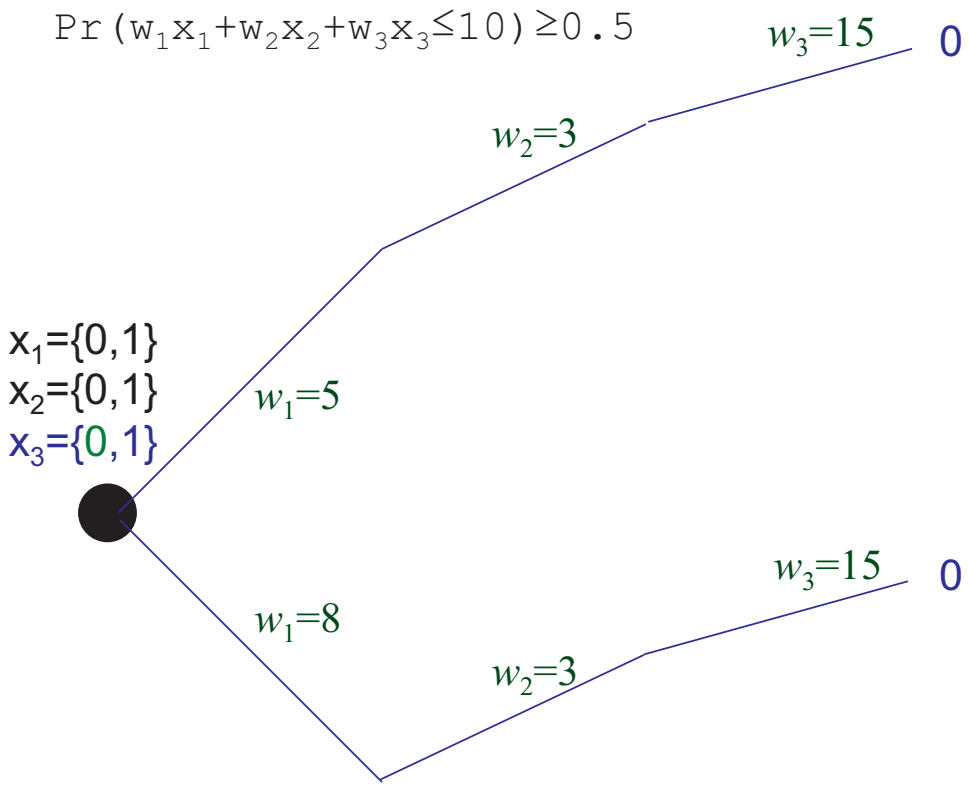
Scenario Reduction



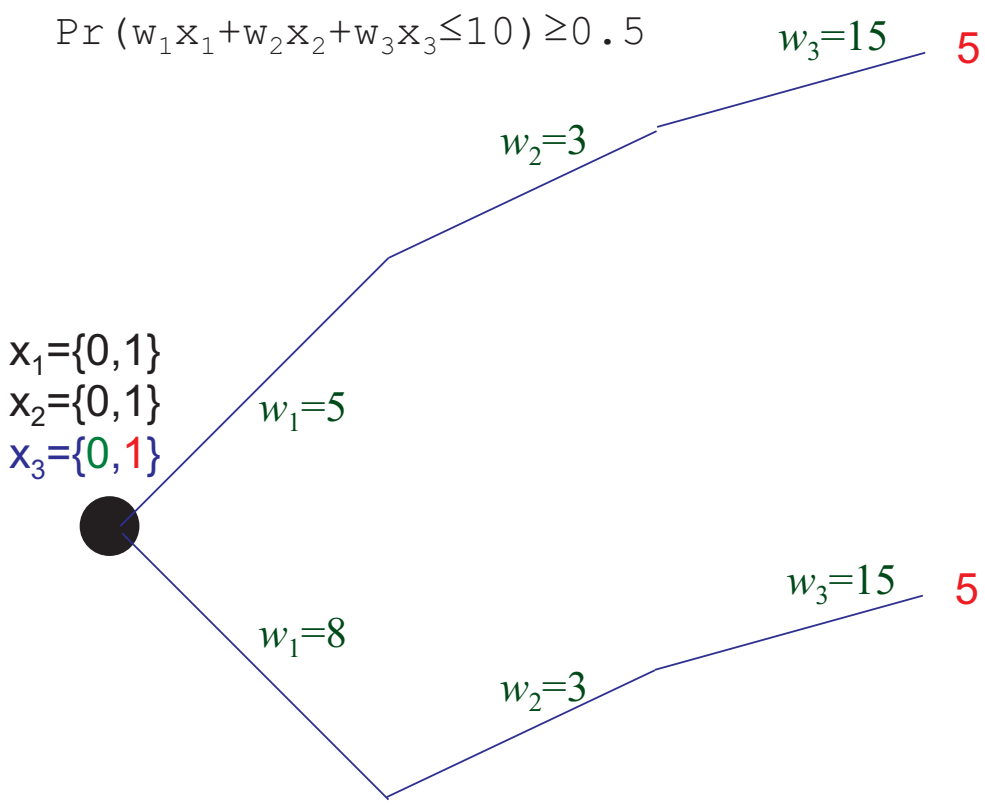
Scenario Reduction



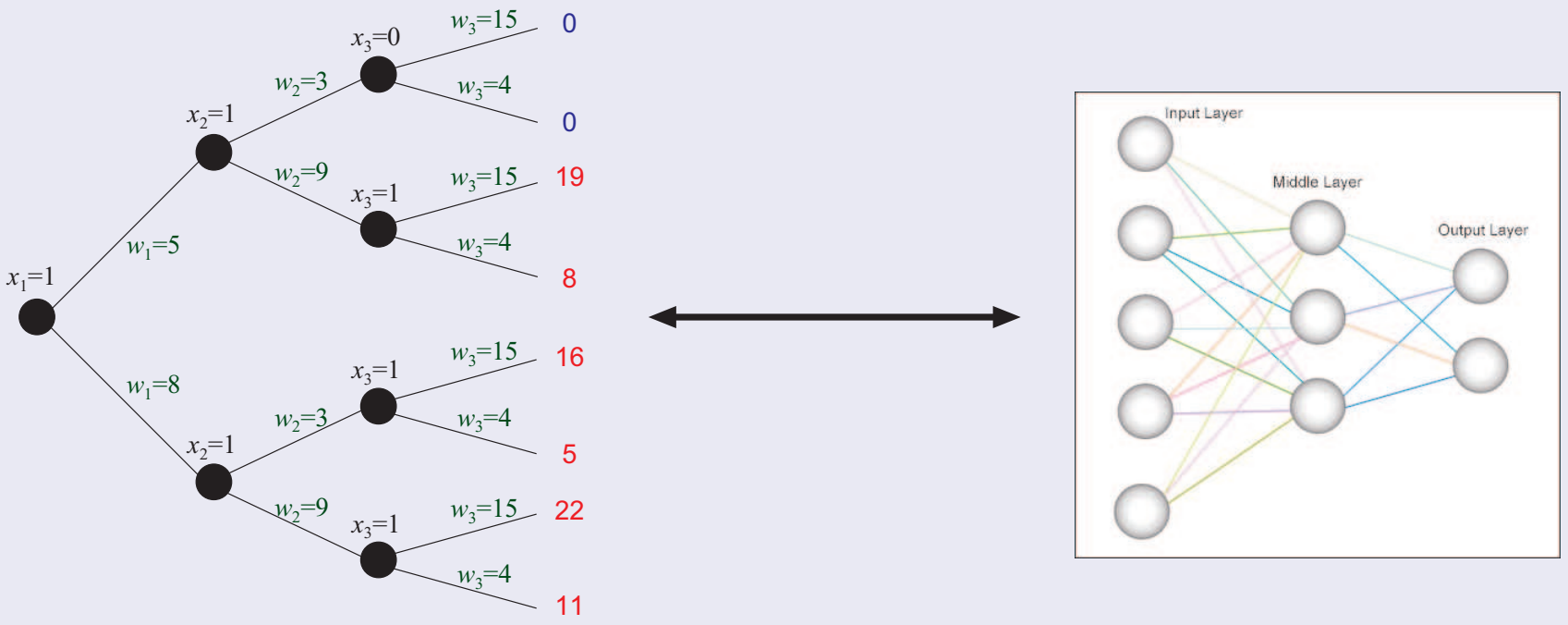
Scenario Reduction



Scenario Reduction



Neuro-evolutionary Stochastic Constraint Programming



S. Prestwich, S. A. Tarim, R. Rossi, B. Hnich,
Evolving Parameterised Policies for Stochastic Constraint Programming,
submitted for possible publication to the 15th International Conference on Principles and Practice of Constraint Programming (CP-09) Lisbon, Portugal, September 21-24, 2009

Food Supply Chain Networks

- Farm - Slaugherhouse allocation (with W. Rijpkema)
- Inventory Control for Perishable Items (with K. Pauls-Worms)

Agrifood Domain

- Land Allocation
- Pest Control

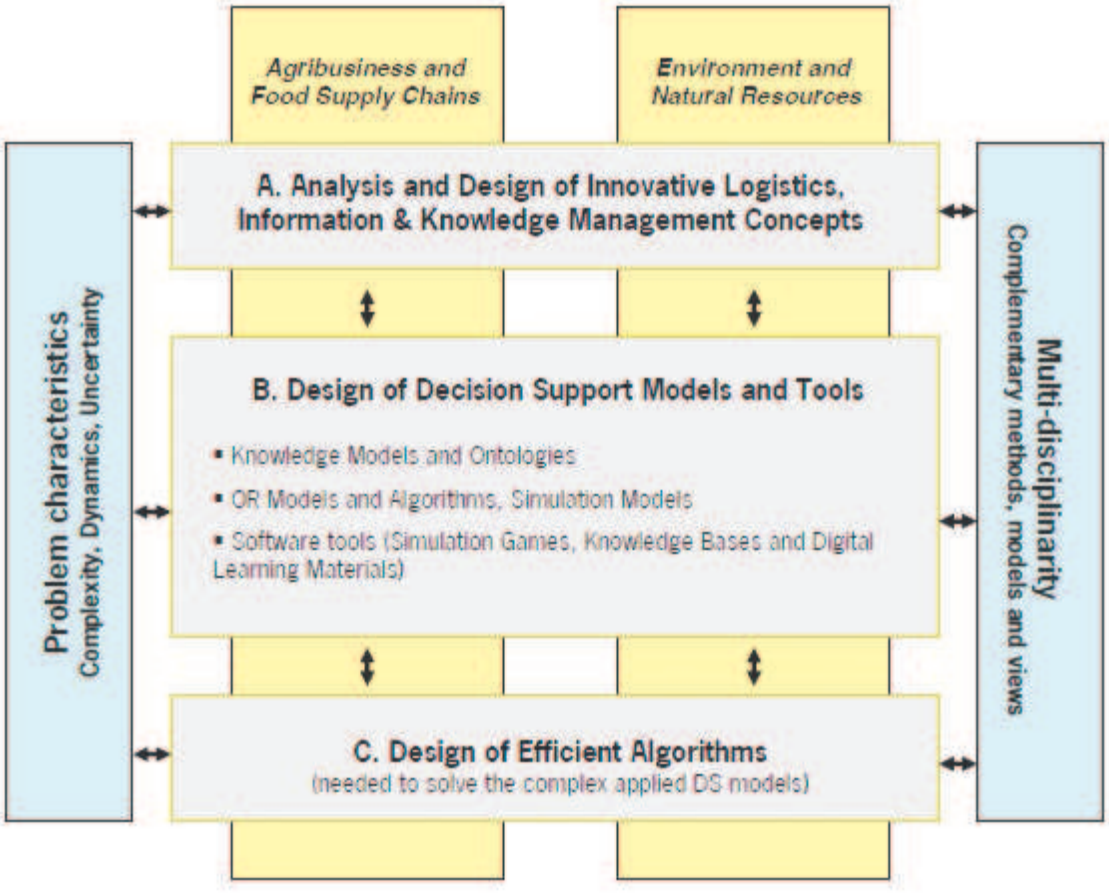
Environmental Domain

- ...

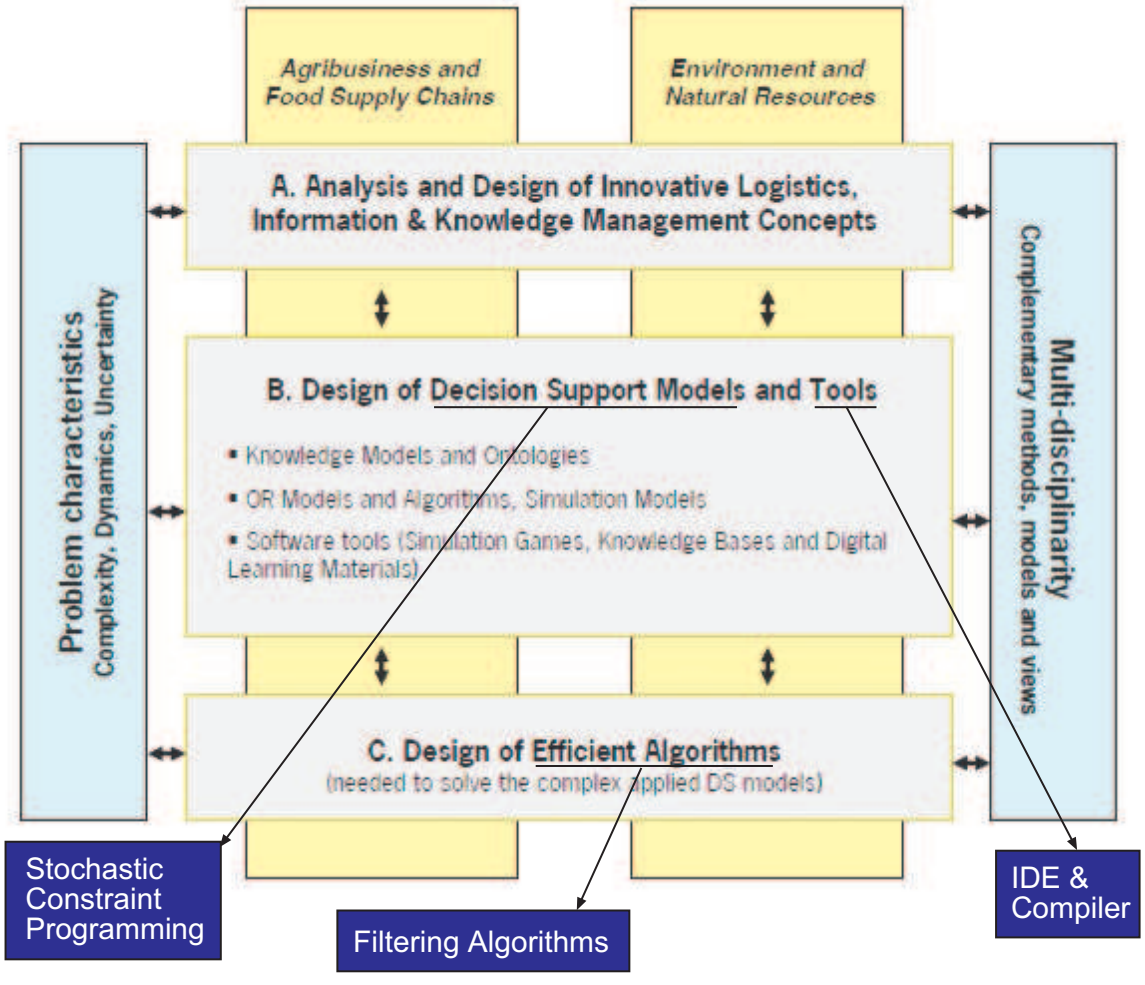
Education

- Decision Science II ???

The LDI research framework



The LDI research framework



Summary

- We discussed a **Framework** for **Modeling Decision Problems under Uncertainty**
  - Stochastic Constraint Programming
  - Global Chance-constraints
  - Stochastic OPL
- We presented some current and possible future **application areas**
- We **positioned** our Framework within the **LDI research framework**

Questions