Synthesizing Filtering Algorithms in Stochastic Constraint Programming

Brahim Hnich\textsuperscript{1}
Roberto Rossi\textsuperscript{2}
S. Armagan Tarim\textsuperscript{3}
Steven D. Prestwich\textsuperscript{4}

\textsuperscript{1}Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey
\textsuperscript{1}LDI, Wageningen UR, the Netherlands
\textsuperscript{3}Dept. of Operations Management, Nottingham University Business School, UK
\textsuperscript{4}Cork Constraint Computation Centre, University College Cork, Ireland

XL Annual Conference of the Italian Operational Research Society, AIRO 2009
A Complete Overview
Decision Making in a Deterministic Setting

Decision Support Systems
- ERP
- CRM
- Inventory Control
- Production Planning
- Transport Scheduling

Applications
- Decision Making
- Constraint Programming
- Stochastic Constraint Programming
- Ongoing Research
- Conclusions

Theoretical Results
- Simplex
- Shortest Path

Modeling Frameworks
- MIP
- CP
- LP

Minimize \( \sum c_j x_j \)
Subject to \( \sum a_{ij} x_j \geq b_i \) for all \( i \)
\( x_j \geq 0 \) for all \( j \)
Decision Making in a Deterministic Setting
Decision Making in a Deterministic Setting

- Decision Support Systems
- Applications
- Modeling Frameworks
- Theoretical Results
- MIP
- CP
- LP
- Simplex
- Shortest Path
- Production Planning
- Transport Scheduling
- ERP
- CRM
- Inventory Control

Mathematical Model:

Minimize: \( \sum_{j} c_j x_j \)

Subject to:

\( \sum_{j} a_{ij} x_j \geq b_i \) \( \forall i \)

\( x_j \geq 0 \) \( \forall j \)
Decision Making Under Uncertainty: A Pervasive Issue
Decision Making Under Uncertainty
Decision Making

Decision Making Under Uncertainty

- Constraint Programming
- Stochastic Constraint Programming
- Ongoing Research
- Conclusions

Introduction

Decision Support Systems

Applications

Modeling Frameworks

Theoretical Results

Simplex

Convex Analysis

Inventory Control

Production Planning

Transport Scheduling

STOCHASTIC OPL

CRM

ERP

Decision Making Under Uncertainty

\[
\min_{x \in S} \{ g(x) = E_{p \sim G(x,W)} \}
\]
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 non-negative. The **maximum weight** that we can carry in the bag is $c$.

**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

Objectives:
\[ \max \sum_{i=1}^{k} r_i x_i \]

Constraints:
\[ \sum_{i=1}^{k} w_i x_i \leq c \]

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 kind of item $i$ has

- a **deterministic profit** $r_i$.
- a **size** $w_i$, which is not known at the time the decision has to be made. The decision maker knows the probability distribution of $w_i$.

A per unit **penalty cost** $p$ has to be paid for exceeding the capacity of the knapsack. The **probability of not exceeding** the capacity of the knapsack should be greater or equal to a given **threshold** $\theta$.

**Objective:** find the knapsack that maximizes the expected profit.
**SSKP: Stochastic Programming Formulation**

**Objective:**

\[
\max \left\{ \sum_{i=1}^{k} r_i X_i - \rho \mathbb{E} \left[ \sum_{i=1}^{k} W_i X_i - c \right] ^+ \right\}
\]

**Subject to:**

\[
\Pr \left\{ \sum_{i=1}^{k} W_i X_i \leq c \right\} \geq \theta
\]

**Decision variables:**

\[X_i \in \{0, 1\} \quad \forall i \in 1, \ldots, k\]

**Stochastic variables:**

\[W_i \rightarrow \text{item } i \text{ weight } \forall i \in 1, \ldots, k\]

**Stage structure:**

\[V_1 = \{X_1, \ldots, X_k\}\]

\[S_1 = \{W_1, \ldots, W_k\}\]

\[L = [\langle V_1, S_1 \rangle]\]
Basic Notions

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 $f(\hat{V})$ defined on a subset $\hat{V}$ of the decision variables in $V$. The aim in a COP is to find a feasible solution that minimizes (maximizes) the objective function.
Basic Notions

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).
### Sample COP: 0-1 KP

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

### Filtered domains at \( P_1 \)

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

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.
A slightly formal definition

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
- \( \theta_h \) 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 \( f(\hat{V}, \hat{S}) \) we obtain a SCOP.
Basic Notions

An Example

Sample SCOP: SSKP

- \( V = \{x_1, \ldots, x_3\} \)
- \( D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\} \)
- \( S = \{w_1, \ldots, w_3\} \)
- \( D(w_1) = \{5(0.5), 8(0.5)\}, D(w_2) = \{3(0.5), 9(0.5)\}, D(w_3) = \{15(0.5), 4(0.5)\} \)
- \( C = \{Pr(w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) \geq 0.2\} \)
- \( L = [\langle V, S \rangle] \)
- \( f(x_1, \ldots, x_3) = 8x_1 + 15x_2 + 10x_3 - 2E \max \left[ 0, \sum_{i=1}^3 w_i x_i - 20 \right] \)
## Basic Notions

### An Example

**Sample SCOP: DSKP**

- \( V = \{x_1, \ldots, x_3\} \)
- \( D(x_i) = \{0, 1\} \quad \forall i \in \{1, \ldots, 3\} \)
- \( S = \{w_1, \ldots, w_3\} \)
- \( D(w_1) = \{5(0.5), 8(0.5)\}, D(w_2) = \{3(0.5), 9(0.5)\}, D(w_3) = \{15(0.5), 4(0.5)\} \)
- \( C = \{Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2\} \)
- \( L = [\langle \{x_1\}, \{w_1\}\rangle, \langle \{x_2\}, \{w_2\}\rangle, \langle \{x_3\}, \{w_3\}\rangle] \)
- \( f(x_1, \ldots, x_3) = \)
  \[
  \mathbb{E}[8x_1 + 15x_2 + 10x_3] - 2\mathbb{E} \max \left[ 0, \sum_{i=1}^{3} w_i x_i - 20 \right]
  \]
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

```plaintext
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;
};
```
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;
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;
}
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**

Stochastic Constraint Program

```
Objective: \( \max \left\{ \sum_{i=1}^{n} r_i x_i - p \sum_{i=1}^{n} W_i x_i - c^i \right\} \)
```

Stochastic OPL Model

```
stoch myrand[onstage]=...;
int nbItems=...;
float c = ...;
float q = ...;
range Items 1..nbItems;
range onstage 1..1;
float W[Items,onstage]*myrand = ...;
float r[Items] = ...;
dvar float s;
dvar int x[Items] in 0..1;
maximize
sum(i in Items) x[i]*r[i] - expected(c^n);

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

Deterministic equivalent model

```
int nbWorlds=...;
range Worlds 1..nbWorlds;
int nbItems=...;
range Items 1..nbItems;
float c = ...;
float W[Worlds,Items] = ...;
float Pr[Worlds]=...;
float r[Items] = ...;
float q = ...;
cvar float s[Worlds];
cvar int x[Items] in 0..1;
maximize
\( \sum_{i} (\text{sum}(j \text{ in Items}) x[i][j] r[i]) - c \cdot \text{sum}(j \text{ in Worlds}) \cdot Pr[j] \cdot x[j][j] \)
subject to:
\( \forall j \text{ in Worlds} \)
\( x[i][j] = \text{sum}(i \text{ in Items}) W[j,i] x[i] \cdot q; \)
\( \text{sum}(j \text{ in Worlds}) \cdot Pr[j] \cdot x[i][j] \cdot x[i][j] \cdot q \) >= 0.2;
```
SSKP: Compiled Deterministic Equivalent CSP

```plaintext
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;
};
```
## Scenario-based Compilation

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

### Drawbacks
- **Size** of the compiled model
- **Constraint Propagation** not fully supported
Solution Methods

An Alternative Approach to Seamless Stochastic Optimization

Stochastic Constraint Program

Objective:
\[ \max \left\{ \sum_{i=1}^{n} r_i X_i \right\} \]
Subject to:
\[ \sum_{i=1}^{n} W_i X_i \leq c \]
Decision variables:
\[ X_i \in \{0, 1\} \]
Stochastic variables:
\[ W_i \rightarrow \text{item weight} \]
Stage structure:
\[ V_i = \{X_1, \ldots, X_n\} \]

Stochastic OPL Model

stoch myrand[onstage]=...;
int nbItems=...
float c = ...
float q = ...
range Items 1..nbItems;
ranged onstage 1..1;
float W[Items, onstage] = myrand = ...;
float r[Items] = ...
dvar float x;
dvar int x[Items] in 0..1;
maximize
\[ \sum_{i=1}^{n} x[i] \cdot r[i] - \text{expected}(x\cdot s) \]
subject to:
\[ z \geq \sum_{i=1}^{n} W[i] \cdot x[i] \cdot q; \]
\[ \text{prob}(\sum_{i=1}^{n} W[i] \cdot x[i] \cdot q \geq 0.6) = \text{...}; \]

Constraint Programming Solver supporting Global Chance-Constraints

Filtering Algorithms for Global Chance-Constraints
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)
Global Chance-Constraints

Filtering in SCSPs

**Stochastic Programming Model**

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

**Global Chance-Constraint**

`stochLinIneq(x,W,Pr,q,0.2);`
SSKP: Compiled Deterministic Equivalent CSP with Global Chance-Constraints

```c
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],
 [8.9,4],
 [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);
};
```
Global Chance-Constraints

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)
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) > 0.5 \]

Search Tree

\[ x_1 = \{0, 1\} \]
\[ x_2 = \{0, 1\} \]
\[ x_3 = \{0, 1\} \]

Solution Tree

\[ w_1 = 5 \]
\[ w_2 = 3 \]
\[ w_3 = 4 \]
\[ w_2 = 9 \]
\[ w_3 = 15 \]
\[ w_1 = 8 \]
\[ w_2 = 3 \]
\[ w_3 = 4 \]
\[ w_2 = 9 \]
\[ w_3 = 15 \]
\[ w_3 = 4 \]
\[ w_3 = 4 \]
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) > 0.5 \]

Search Tree

Solution Tree

\[ x_1 = \{0, 1\} \quad x_2 = \{0, 1\} \quad x_3 = \{0\} \]
Filtering Algorithms for GCCs: An example

\[ \Pr (w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) > 0.5 \]

Search Tree

Solution Tree

\[ x_1 = \{0, 1\} \quad x_2 = \{0, 1\} \quad x_3 = \{0\} \]

\[ x_1 = \{0, 1\} \quad x_2 = \{0, 1\} \quad x_3 = \{0, 1\} \]
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) > 0.5 \]

**Search Tree**

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0, 1\} \)

**Solution Tree**

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0, 1\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0, 1\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0, 1\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0\} \)

- \( x_1 = \{0, 1\} \)
- \( x_2 = \{0, 1\} \)
- \( x_3 = \{0, 1\} \)
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

$$\Pr (w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) > 0.5$$
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) > 0.5 \]

Search Tree

Solution Tree

- \( x_1 = \{1\} \)  \( x_2 = \{0, 1\} \)  \( x_3 = \{0\} \)
- \( x_1 = \{1\} \)  \( x_2 = \{0, 1\} \)  \( x_3 = \{0\} \)
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) > 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr (w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) > 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) > 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) \geq 0.5 \]
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr (w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) \geq 0.5 \]

Search Tree

Solution Tree

- \( x_1 = \emptyset \), \( x_2 = \{1\} \), \( x_3 = \{1\} \)
- \( x_1 = \emptyset \), \( x_2 = \{1\} \), \( x_3 = \{1\} \)
- \( x_1 = \emptyset \), \( x_2 = \{1\} \), \( x_3 = \{0\} \)
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) \geq 0.5 \]
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.5 \]
Global Chance-Constraints

Filtering Algorithms for GCCs: An example

\[ \Pr(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.5 \]
Filtering Algorithms for GCCs: An example

\[ \Pr(w_1 x_1 + w_2 x_2 + w_3 x_3 \leq 10) \geq 0.5 \]

**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*,
*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:
\[
\text{stochLinIneq}(x, W, Pr, q, 0.2);
\]
i.e. SSKP $\rightarrow \text{Pr}(w_1x_1 + w_2x_2 + w_3x_3 \leq 10) \geq 0.2$
Future work

Considering more global constraints:

allDifferent()
NValue()
Cumulative()

...
Integrated Development Environment
Summary

We discussed a Framework for Modeling Decision Problems under Uncertainty

- Stochastic Constraint Programming
- Global Chance-constraints
- Stochastic OPL