On a stochastic knapsack problem

Stefanie Kosuch, Marc Letournel and Abdel Lisser

Université Paris XI - Sud (France)
LRI - GraphComb

Cologne-Twente-Workshop
25^{th}-27^{th} May 2010
Köln, Germany

1 Introduction

1. Introduction

2. Chance-Constrained Knapsack Problem
   - Problem Formulation

The Deterministic Knapsack Problem

The Deterministic Knapsack Problem

- $c > 0$: knapsack weight capacity

The Deterministic Knapsack Problem

- $c > 0$: knapsack weight capacity
- n items

The Deterministic Knapsack Problem

- $c > 0$: knapsack weight capacity
- n items
- $r_i > 0$: reward per weight unit of item i

The Deterministic Knapsack Problem

- $c > 0$: knapsack weight capacity
- n items
- $r_i > 0$: reward per weight unit of item i
- w_i: weight of item i

The Stochastic Knapsack Problem with Random Weights

- $c > 0$: knapsack weight capacity
- n items
- $r_i > 0$: reward per weight unit of item i
- χ_i: independently normally distributed weight of item i

The Stochastic Knapsack Problem with Random Weights

- $c > 0$: knapsack weight capacity
- n items
- $r_i > 0$: reward per weight unit of item i
- χ_i: independently normally distributed weight of item i
- μ_i,σ_i: mean and standard deviation of χ_i

The Stochastic Knapsack Problem with Random Weights

- $c > 0$: knapsack weight capacity
- n items
- $r_i > 0$: reward per weight unit of item i
- χ_i: independently normally distributed weight of item i
- μ_i,σ_i: mean and standard deviation of χ_i

Problem

How to handle possibility of violated capacity constraint?

Stochastic Models

Stochastic Models

- **Two-Stage Model**: decision can be corrected in second stage (at a cost)

Stochastic Models

- **Two-Stage Model**: decision can be corrected in second stage (at a cost)
- **Simple Recourse Model**: violated capacity constraint ⇒ penalty

Stochastic Models

- **Two-Stage Model**: decision can be corrected in second stage (at a cost)
- **Simple Recourse Model**: violated capacity constraint ⇒ penalty
- **Risk Aversion**: given a threshold for the gain, minimize the risk

Stochastic Models

- **Two-Stage Model**: decision can be corrected in second stage (at a cost)
- **Simple Recourse Model**: violated capacity constraint ⇒ penalty
- **Risk Aversion**: given a threshold for the gain, minimize the risk
- **Chance-Constrained**: given a threshold for the risk, maximize the expected gain

Stochastic Models

- **Two-Stage Model**: decision can be corrected in second stage (at a cost)
- **Simple Recourse Model**: violated capacity constraint $\Rightarrow$ penalty
- **Risk Aversion**: given a threshold for the gain, minimize the risk
- **Chance-Constrained**: given a threshold for the risk, maximize the expected gain

Outline

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0, 1\}^n$: decision vector

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

- $p \in (0.5, 1]$

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x\in\{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

- $p \in (0.5, 1]$: prescribed probability

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

- $p \in (0.5, 1]$: prescribed probability
- $1-p \in [0, 0.5)$

Chance-Constrained Knapsack Problem (CCKP)

- $x \in \{0,1\}^n$: decision vector

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$
$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

- $p \in (0.5, 1]$: prescribed probability
- $1 - p \in [0, 0.5)$: risk threshold

Equivalence

Equivalence

$$\mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} = \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]$$

Equivalence

$$\mathcal{P}\{\sum_{i=1}^{n}\chi_i x_i \leq c\} = \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n}\chi_i x_i)\right]$$

- $\mathbb{1}_{\mathbb{R}^+}$: indicator function of the positive real interval (Heaviside function)

Equivalence

$$\mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} = \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]$$

- $\mathbb{1}_{\mathbb{R}^+}$: indicator function of the positive real interval (Heaviside function)

Chance-Constrained Knapsack Problem (CCKP)

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$$

Equivalence

$$\mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} = \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]$$

- $\mathbb{1}_{\mathbb{R}^+}$: indicator function of the positive real interval (Heaviside function)

Expectation-Constrained Knapsack Problem (ECKP)

$$\max_{x \in \{0,1\}^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Problem Solving Method - Scheme

Problem Solving Method - Scheme

- Branch-and-Bound algorithm

Problem Solving Method - Scheme

- Branch-and-Bound algorithm
- Solve linear relaxation to compute upper bounds

Problem Solving Method - Scheme

- Branch-and-Bound algorithm
- Solve linear relaxation to compute upper bounds
- Use stochastic gradient type algorithm to solve linear relaxation

Problem Solving Method - Scheme

- Branch-and-Bound algorithm
- Solve linear relaxation to compute upper bounds
- Use stochastic gradient type algorithm to solve linear relaxation
- Apply "Integration by parts"-method to obtain needed gradient

Outline

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x\in[0,1]^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i\chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in [0,1]^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in [0,1]^n} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Restricted feasible set

Empty knapsack not optimal for *ECKP* $\Rightarrow ||x^*_{rel}||_\infty \geq \frac{1}{n}$

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in X} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Restricted feasible set

Empty knapsack not optimal for *ECKP* $\Rightarrow ||x^*_{rel}||_\infty \geq \frac{1}{n}$

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in X} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Restricted feasible set

Empty knapsack not optimal for $ECKP \Rightarrow ||x^*_{rel}||_\infty \geq \frac{1}{n}$

- $X := \{x \in [0,1]^n | \; ||x||_\infty \geq \frac{1}{n}\}$

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in X} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Convexity of knapsack chance-constraints (A. Prekopa)

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in X} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Convexity of knapsack chance-constraints (A. Prekopa)

Let χ_i follow a log-concave probability distribution. Then a knapsack chance-constraint of the form $\mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$ defines a convex set.

Definition

Continuous Expectation-Constrained Knapsack Problem:

$$\max_{x \in X} \quad \mathbb{E}[\sum_{i=1}^{n} r_i \chi_i x_i]$$

$$s.t. \quad \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$$

Convexity of knapsack chance-constraints (A. Prekopa)

Let χ_i follow a log-concave probability distribution. Then a knapsack chance-constraint of the form $\mathcal{P}\{\sum_{i=1}^{n} \chi_i x_i \leq c\} \geq p$ defines a convex set.

- Example: normal distribution

Outline of the Stochastic Arrow-Hurwicz algorithm

- Solve Lagrangian Dual
  $\longrightarrow$ "non-constrained" stochastic optimization problem

Outline of the Stochastic Arrow-Hurwicz algorithm

- Solve Lagrangian Dual
  $\longrightarrow$ "non-constrained" stochastic optimization problem
- objective function: $\mathcal{L}(x, \lambda) = \mathbb{E}[\ell(x, \lambda, \chi)]$

Outline of the Stochastic Arrow-Hurwicz algorithm

- Solve Lagrangian Dual
  $\longrightarrow$ "non-constrained" stochastic optimization problem
- objective function: $\mathcal{L}(x, \lambda) = \mathbb{E}[\ell(x, \lambda, \chi)]$
- stochastic gradient algorithm

Outline of the Stochastic Arrow-Hurwicz algorithm

- Solve Lagrangian Dual
  $\longrightarrow$ "non-constrained" stochastic optimization problem
- objective function: $\mathcal{L}(x, \lambda) = \mathbb{E}[\ell(x, \lambda, \chi)]$
- stochastic gradient algorithm
  - draw sample $\hat{\chi}$ of random vector χ

Outline of the Stochastic Arrow-Hurwicz algorithm

- Solve Lagrangian Dual
  $\longrightarrow$ "non-constrained" stochastic optimization problem
- objective function: $\mathcal{L}(x, \lambda) = \mathbb{E}[\ell(x, \lambda, \chi)]$
- stochastic gradient algorithm
  - draw sample $\hat{\chi}$ of random vector χ
  - use $\nabla_x \ell(x^{k-1}, \lambda^{k-1}, \hat{\chi})$ (instead of $\nabla_x \mathcal{L}(x^{k-1}, \lambda^{k-1})$) to update x^k

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

Choose $x^0 \in X$, $\lambda^0 \in [0, \infty)$, σ-sequences $(\epsilon^k)_{k \in \mathbb{N}^*}$ & $(\rho^k)_{k \in \mathbb{N}^*}$.

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

Choose $x^0 \in X$, $\lambda^0 \in [0, \infty)$, σ-sequences $(\epsilon^k)_{k \in \mathbb{N}^*}$ & $(\rho^k)_{k \in \mathbb{N}^*}$.
At iteration $k \geq 1$, draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

Choose $x^0 \in X$, $\lambda^0 \in [0, \infty)$, σ-sequences $(\epsilon^k)_{k \in \mathbb{N}^*}$ & $(\rho^k)_{k \in \mathbb{N}^*}$.
At iteration $k \geq 1$, draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
Update x^k and λ^k as follows:

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

Choose $x^0 \in X$, $\lambda^0 \in [0, \infty)$, σ-sequences $(\epsilon^k)_{k\in\mathbb{N}^*}$ & $(\rho^k)_{k\in\mathbb{N}^*}$.
At iteration $k \geq 1$, draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
Update x^k and λ^k as follows:

$$x^k = x^{k-1} + \epsilon^k \nabla_x \ell(x^{k-1}, \lambda^{k-1}, \chi^k)$$

$$\lambda^k = \lambda^{k-1} + \rho^k (p - \mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i^k x_i^k))$$

The Stochastic Arrow-Hurwicz Algorithm

Stochastic Arrow-Hurwicz Algorithm

Choose $x^0 \in X$, $\lambda^0 \in [0, \infty)$, σ-sequences $(\epsilon^k)_{k \in \mathbb{N}^*}$ & $(\rho^k)_{k \in \mathbb{N}^*}$.
At iteration $k \geq 1$, draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
Update x^k and λ^k as follows:

$$x^k = x^{k-1} + \epsilon^k \nabla_x \ell(x^{k-1}, \lambda^{k-1}, \chi^k)$$

$$\lambda^k = \lambda^{k-1} + \rho^k (p - \mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i^k x_i^k))$$

For all $i = 1, ..., n$: If $x_i^k > 1$ set $x_i^k = 1$ / If $x_i^k < 0$ set $x_i^k = 0$
If $\lambda_i^k > 0$ set $\lambda_i^k = 0$

Outline

Problem

- Use $\nabla_x \ell(x^{k-1}, \lambda^{k-1}, \chi^k)$ to update x^k

Problem

- Use $\nabla_x \ell(x^{k-1}, \lambda^{k-1}, \chi^k)$ to update x^k
- *BUT*:

$$\ell(x^{k-1}, \lambda^{k-1}, \chi^k) = \sum_i r_i \chi_i^k x_i^{k-1} - \lambda^{k-1} \left(p - \mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i^k x_i^{k-1}) \right)$$

Problem

- Use $\nabla_x \ell(x^{k-1}, \lambda^{k-1}, \chi^k)$ to update x^k
- *BUT*:

$$\ell(x^{k-1}, \lambda^{k-1}, \chi^k) = \sum_i r_i \chi_i^k x_i^{k-1} - \lambda^{k-1} \left(p - \mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i^k x_i^{k-1}) \right)$$

- $\Rightarrow \ell$ non differentiable

Solution

Solution

- use *Integration by Parts* to reformulate

$$\mathbb{E}[\ell(x,\chi)] = \int_{-\infty}^{\infty} \ell(x,\chi)\,\mathrm{d}\chi$$

Solution

- use *Integration by Parts* to reformulate
  $\mathbb{E}[\ell(x,\chi)] = \int_{-\infty}^{\infty} \ell(x,\chi)\, \mathrm{d}\chi$
- $\rightarrow$ differentiable function $\widetilde{\ell}(x,\hat{\chi})$ such that

$$\mathbb{E}[\widetilde{\ell}(x,\chi)] = \mathbb{E}[\ell(x,\chi)]$$

Solution

- use *Integration by Parts* to reformulate
  $\mathbb{E}[\ell(x,\chi)] = \int_{-\infty}^{\infty} \ell(x,\chi)\, \mathrm{d}\chi$
- $\rightarrow$ differentiable function $\widetilde{\ell}(x,\hat{\chi})$ such that

$$\mathbb{E}[\widetilde{\ell}(x,\chi)] = \mathbb{E}[\ell(x,\chi)]$$

Solution

- use *Integration by Parts* to reformulate
  $\mathbb{E}[\ell(x,\chi)] = \int\limits_{-\infty}^{\infty} \ell(x,\chi)\,\mathrm{d}\chi$
- $\rightarrow$ differentiable function $\widetilde{\ell}(x,\hat{\chi})$ such that

$$\mathbb{E}[\widetilde{\ell}(x,\chi)] = \mathbb{E}[\ell(x,\chi)]$$

- replace $\nabla_x \ell(x^{k-1},\lambda^{k-1},\chi^k)$ by $\nabla_x \widetilde{\ell}(x^{k-1},\lambda^{k-1},\chi^k)$ in stochastic gradient algorithm

Outline

Numerical Problem 1: Divergence in most tests

Numerical Problem 1: Divergence in most tests

$$\mathcal{L}(x,\lambda) \;=\; \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]\right)$$

Numerical Problem 1: Divergence in most tests

$$\mathcal{L}(x,\lambda) \;=\; \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right]\right)$$

Integration by Parts method $\rightarrow$ h-th component of the gradient:

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h + \lambda\left(\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h + \frac{(c - \sum_{i=1}^n \chi_i x_i)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Numerical Problem 1: Divergence in most tests

$$\mathcal{L}(x,\lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right]\right)$$

Integration by Parts method $\rightarrow$ h-th component of the gradient:

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h + \lambda\left(\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h + \frac{(c - \sum_{i=1}^n \chi_i x_i)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Numerical Problem 1: Divergence in most tests

$$\mathcal{L}(x,\lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right]\right)$$

Integration by Parts method $\rightarrow$ h-th component of the gradient:

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h + \lambda\left(\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h + \frac{(c - \sum_{i=1}^n \chi_i x_i)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Solution: Reformulation of the expectation-constraint

Solution: Reformulation of the expectation-constraint

$$\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p \Leftrightarrow \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n} \chi_i x_i - c)\right] \leq 1 - p$$

Solution: Reformulation of the expectation-constraint

$$\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p \Leftrightarrow \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n} \chi_i x_i - c)\right] \leq 1 - p$$

Integration by Parts method $\rightarrow$ h-th component of the gradient:

$$\left(\nabla_x \widetilde{\ell}(x, \lambda, \chi)\right)_h =$$

$$r_h \chi_h - \lambda \left(\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n} \chi_i x_i - c) \frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2} \left(\chi_h - \frac{(\sum_{i=1}^{n} \chi_i x_i - c)}{x_\kappa} \nu_h^\kappa \right) \right)$$

Solution: Reformulation of the expectation-constraint

$$\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p \Leftrightarrow \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n} \chi_i x_i - c)\right] \leq 1 - p$$

Integration by Parts method $\rightarrow$ h-th component of the gradient:

$$\left(\nabla_x \widetilde{\ell}(x, \lambda, \chi)\right)_h =$$

$$r_h \chi_h - \lambda \left(\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n} \chi_i x_i - c) \frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2} \left(\chi_h - \frac{(\sum_{i=1}^{n} \chi_i x_i - c)}{x_\kappa} \nu_h^\kappa \right) \right)$$

Numerical Problem 2: Slow convergence

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}\left(\sum_{i=1}^{n}\chi_i x_i - c\right)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa\sigma_\kappa^2}\left(\chi_h - \frac{\left(\sum_{i=1}^{n}\chi_i x_i - c\right)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}\Big(\sum_{i=1}^{n}\chi_i x_i - c\Big)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa\sigma_\kappa^2}\left(\chi_h - \frac{(\sum_{i=1}^{n}\chi_i x_i - c)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}\left(\sum_{i=1}^{n}\chi_i x_i - c\right)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa\sigma_\kappa^2}\left(\chi_h - \frac{\left(\sum_{i=1}^{n}\chi_i x_i - c\right)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n}\chi_i x_i - c)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h - \frac{(\sum_{i=1}^{n}\chi_i x_i - c)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Solution: Choose κ wisely

- in iter. k: $\kappa \in \{1,\ldots,n\}$ (freely) chosen s.t. $x_\kappa^k > 0$
- Remember: feasible set $X = \{x \in [0,1]^n \mid ||x||_\infty \geq \frac{1}{n}\}$
  $\Rightarrow$ such κ exists if x^k feasible

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}\left(\sum_{i=1}^{n}\chi_i x_i - c\right)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa\sigma_\kappa^2}\left(\chi_h - \frac{(\sum_{i=1}^{n}\chi_i x_i - c)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Solution: Choose κ wisely

- in iter. k: $\kappa \in \{1,\dots,n\}$ (freely) chosen s.t. $x_\kappa^k > 0$
- Remember: feasible set $X = \{x \in [0,1]^n \mid \, ||x||_\infty \geq \frac{1}{n}\}$
  $\Rightarrow$ such κ exists if x^k feasible
- choose κ (if possible) as

Numerical Problem 2: Slow convergence

$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}\left(\sum_{i=1}^{n}\chi_i x_i - c\right)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h - \frac{\left(\sum_{i=1}^{n}\chi_i x_i - c\right)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Solution: Choose κ wisely

- in iter. k: $\kappa \in \{1,\ldots,n\}$ (freely) chosen s.t. $x_\kappa^k > 0$
- Remember: feasible set $X = \{x \in [0,1]^n | \; ||x||_\infty \geq \frac{1}{n}\}$
  $\Rightarrow$ such κ exists if x^k feasible
- choose κ (if possible) as

Numerical Problem 2: Slow convergence

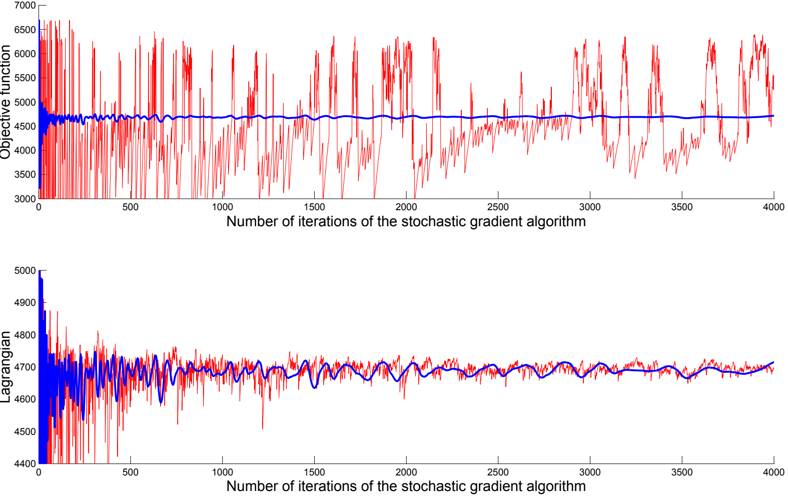
$$\left(\nabla_x \widetilde{\ell}(x,\lambda,\chi)\right)_h =$$

$$r_h\chi_h - \lambda\left(\mathbb{1}_{\mathbb{R}^+}(\sum_{i=1}^{n}\chi_i x_i - c)\frac{(\chi_\kappa - \mu_\kappa)}{x_\kappa \sigma_\kappa^2}\left(\chi_h - \frac{(\sum_{i=1}^{n}\chi_i x_i - c)}{x_\kappa}\nu_h^\kappa\right)\right)$$

Solution: Choose κ wisely

- in iter. k: $\kappa \in \{1,\ldots,n\}$ (freely) chosen s.t. $x_\kappa^k > 0$
- Remember: feasible set $X = \{x \in [0,1]^n | \; ||x||_\infty \geq \frac{1}{n}\}$
  $\Rightarrow$ such κ exists if x^k feasible
- choose κ (if possible) as

$$\kappa = \arg\max_{i=1,\ldots,n}\{x_i^k | x_i^k \geq \frac{1}{n} \wedge (\chi_i^k - \mu_i) > 0\}$$


Objective function
7000
6500
6000
5500
5000
4500
4000
3500
3000
0
500
1000
1500
2000
2500
3000
3500
4000
Number of iterations of the stochastic gradient algorithm
Lagrangian
5000
4900
4800
4700
4600
4500
4400
0
500
1000
1500
2000
2500
3000
3500
4000
Number of iterations of the stochastic gradient algorithm

Technical Problem: Non-convexity

Technical Problem: Non-convexity

- $\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$ defines a convex feasible set, but...

Technical Problem: Non-convexity

- $\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$ defines a convex feasible set, but...
- ... the Lagrangian

$$\mathcal{L}(x, \lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]\right)$$

is not concave.

Technical Problem: Non-convexity

- $\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$ defines a convex feasible set, but...
- ... the Lagrangian

$$\mathcal{L}(x, \lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda \left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]\right)$$

is not concave.

Technical Problem: Non-convexity

- $\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right] \geq p$ defines a convex feasible set, but...
- ... the Lagrangian

$$\mathcal{L}(x, \lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^{n} \chi_i x_i)\right]\right)$$

is not concave. At least not everywhere...

Technical Problem: Non-convexity

- $\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right] \geq p$ defines a convex feasible set, but...
- ... the Lagrangian

$$\mathcal{L}(x,\lambda) = \mathbb{E}\left[\sum_i r_i \chi_i x_i\right] - \lambda\left(p - \mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right]\right)$$

is not concave. At least not everywhere...

CDF involved

$$\mathbb{E}\left[\mathbb{1}_{\mathbb{R}^+}(c - \sum_{i=1}^n \chi_i x_i)\right] = \mathcal{P}\{\sum_{i=1}^n \chi_i x_i \leq c\} = \Phi(c)$$

where Φ is the cumulative distribution function of $\sum_{i=1}^n \chi_i x_i$

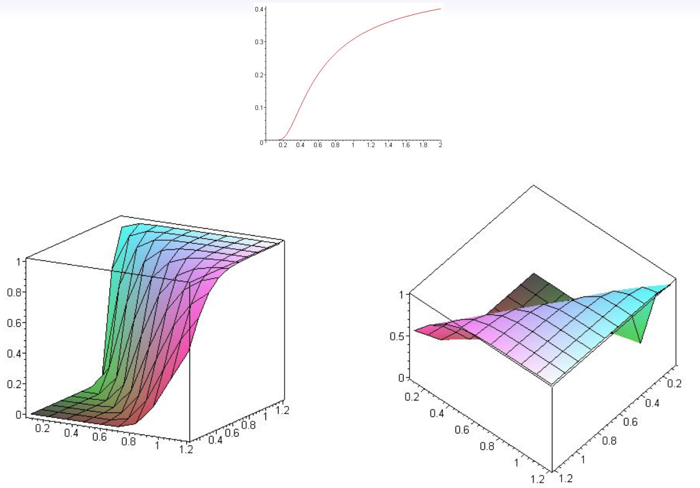


Figure: Constraint function in 1 and 2-dimensional case with realistic constants

Outline

	Arrow-Hurwicz & IP-method		
n	**considered nodes**	**CPU-time (sec) B-and-B**	**Gap**
15	34	0.06	0 %
20	63	0.13	0 %
30	436	1.22	0 %
50	7051	31.34	0 %
75	23911	161.47	0 %
100	98479	1049.18	?
150	*	*	*

* CPU-time exceeds 1h

Table: Numerical results for the (combinatorial) *ECKP*

Conclusion

- **Chance-** (Expectation-) **Constrained Knapsack problem** with random weights

Conclusion

- **Chance-** (Expectation-) **Constrained Knapsack problem** with random weights
- **Branch-and-Bound algorithm** to search binary solution space

Conclusion

- **Chance-** (Expectation-) **Constrained Knapsack problem** with random weights
- **Branch-and-Bound algorithm** to search binary solution space
- Solve **relaxation** to obtain upper bounds using **stochastic gradient algorithm**

Conclusion

- **Chance-** (Expectation-) **Constrained Knapsack problem** with random weights
- **Branch-and-Bound algorithm** to search binary solution space
- Solve **relaxation** to obtain upper bounds using **stochastic gradient algorithm**
- **Integration by Parts** method to overcome non-differentiability

Conclusion

- **Chance**- (Expectation-) **Constrained Knapsack problem** with random weights
- **Branch-and-Bound algorithm** to search binary solution space
- Solve **relaxation** to obtain upper bounds using **stochastic gradient algorithm**
- **Integration by Parts** method to overcome non-differentiability
- Convergence Issues → solved with **more intelligent choices**

Conclusion

- **Chance-** (Expectation-) **Constrained Knapsack problem** with random weights
- **Branch-and-Bound algorithm** to search binary solution space
- Solve **relaxation** to obtain upper bounds using **stochastic gradient algorithm**
- **Integration by Parts** method to overcome non-differentiability
- Convergence Issues $\rightarrow$ solved with **more intelligent choices**
- Able to solve (Binary) Chance-Constrained Knapsack problem with up to 100 items in less than $1h$

Thank you!

Danke!