Genetic Algorithm and Simulation for Chance Constrained Programming problems

Chandra A. Poojari Boby Varghese

CARISMA, School of Information Systems, Computing and Mathematics, Brunel University, London email: Chandra.Poojari@brunel.ac.uk

CARISMA, London

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Outline

1. Increasing interest in making and quantifying risk-based decisions in diverse industrial sectors.
2. We have propose and develop a generic framework combining Genetic algorithm and Simulation.
3. A technique is proposed in order to process constrained optimisation problems using Genetic Algorithms.
4. We define fitness functions that scores a given solution based on its feasibility and optimality.
5. We develop and use a new Statistical library for Simulating random vectors.
6. Statistical analysis are carried out to investigate the performances of the GA and Simulation framework on test problems.

Two principal methods to model Mathematical programming models under uncertainty are:

- Stochastic Programming(SP) ((Dantzig, 1955), (Beale, 1955), (Charnes and Cooper, 1959)), and
- Robust Optimization (RO) ((Ben-Tal and Nemirovski, 2002), (Ghaoui and Lebret, 1997)).

Stochastic Programming

- Allows to model path dependence of the stochastic process within an optimisation model.
- Permits uncountably many states and actions, together with constraints and time-lags.
- Separates the model formulation activity from the solution algorithm.
- Develop plans that hedge against future outcomes.
- Allow the perception of risks to be considered within the decision model.

Applications

- Electric power generation (Murphy et al., 1982),
- Financial planning (Dempster and Consigli, 1998; Dert, 1995; van der Vlerk, 2003),
- Telecommunication network planning (Sen et al., 1994; Tomasgard et al., 1998),
- Supply chain management (MirHassani et al., 2000), Oil industry (Dempster et al., 2000),
- Vehicle manufacturers (Eppen et al., 1989),
- Electricity suppliers (Robinson, 1988; Takriti et al., 1996),
- Environment (Kampas and White, 2003), Transportation (Laporte et al., 1994; Elmaghraby et al., 2001), Chemical processing (Ierapetritou et al., 1996), Military system (Smith, 1999),

Decisions in an SP model

- Anticipative
- Adaptive

Decisions in an SP model

- Anticipative
- Adaptive

Notations

- $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ as the decision vector, $x \in \mathbb{R}^n$,
- $\xi = \{\xi_1, \xi_2, ..., \xi_m\}$ as the stochastic vector, $\xi \in \mathbb{R}^m$, having the expected value vector μ_ξ and the variance-covariance matrix σ_ξ^2.

A Stochastic Programming model

$$
\begin{array}{llcll}
P_{SP} & \text{Max} \;\; f(\mathbf{x}, \xi) & & & \\
& \text{for constraints} & & & \\
& g_i(\mathbf{x}) & \leq & 0 \;\; i = 1, 2, \ldots, p_1 & (1) \\
& g_i(\mathbf{x}, \xi) & \leq & 0 \;\; i = p_1 + 1, p_1 + 2, \ldots p & (2) \\
& h_i(\mathbf{x}) & = & 0 \;\; i = 1, \ldots, q_1 & (3) \\
& h_i(\mathbf{x}, \xi) & = & 0 \;\; i = q_1 + 1, q_1 + 2, \ldots, q & (4)
\end{array}
$$

Let G^d, G^s, H^d, and H^sdenote the index sets for the constraints 1, 2, 3, and 4 respectively. Therefore $|G^d| = p_1$, $|H^d| = q_1$, $|G^s| = p - p_1$, $|H^s| = q - q_1$.

An adaptive decision model

$$P_{SP-Adap} \qquad \begin{array}{rcll} & \text{Max} & (f(\mathbf{x},\xi)|\xi) & \\ & \text{for constraints} & & \\ g_i(\mathbf{x}) & \leq & 0 & i \in G^d \\ (g_i(\mathbf{x},\xi)|\xi) & \leq & 0 & i \in G^s \\ h_i(\mathbf{x}) & = & 0 & i \in H^d \\ (h_i(\mathbf{x},\xi)|\xi) & = & 0 & i \in H^s. \end{array}$$

Anticipative decision model

$$
P_{SP-Anti} \qquad \begin{array}{rcll}
\multicolumn{4}{l}{\text{Max} \quad E_{\xi}[f(\mathbf{x},\xi)]} \\
\multicolumn{4}{l}{\text{for constraints}} \\
g_i(\mathbf{x}) & \leq & 0 & i \in G^d \\
E_{\xi}[g_i(\mathbf{x},\xi)] & \leq & 0 & i \in G^s \\
h_i(\mathbf{x}) & = & 0 & i \in H^d \\
E_{\xi}[h_i(\mathbf{x},\xi)] & = & 0 & i \in H^s.
\end{array}
$$

Stochastic Programming with Recourse

- Decision variables: $(x_1, x_2(\xi))$, where $x_1 \in \Re^{n_1}$ and $x_2 \in \Re^{n_2}$, $(n = n_1 + n_2)$.
- Objective : one component is the cost for making the *here-and-now* decisions, x_1, and the second component is the expected value of the penalty for taking corrective actions, $x_2(\xi)$.
- Algorithms :
  1. Primal and dual decomposition: Slyke and Wets (1969); Birge (1985); Rockafellar and Wets (1991); Mulvey and Ruszczyński (1995),
  2. Approximation techniques: Romisch and Schultz (1991); Shapiro (1991); Shapiro and de Mello (2000); Robinson (1996); Higle and Sen (1996).

Stochastic Programming with Recourse

- Decision variables: $(x_1, x_2(\xi))$, where $x_1 \in \Re^{n_1}$ and $x_2 \in \Re^{n_2}$, $(n = n_1 + n_2)$.
- Objective : one component is the cost for making the *here-and-now* decisions, x_1, and the second component is the expected value of the penalty for taking corrective actions, $x_2(\xi)$.
- Algorithms :
  1. Primal and dual decomposition: Slyke and Wets (1969); Birge (1985); Rockafellar and Wets (1991); Mulvey and Ruszczyński (1995),

2. Approximation techniques: Romisch and Schultz (1991); Shapiro (1991); Shapiro and de Mello (2000); Robinson (1996); Higle and Sen (1996).

Stochastic Programming with Recourse

- Decision variables: $(x_1, x_2(\xi))$, where $x_1 \in \Re^{n_1}$ and $x_2 \in \Re^{n_2}$, $(n = n_1 + n_2)$.
- Objective : one component is the cost for making the *here-and-now* decisions, x_1, and the second component is the expected value of the penalty for taking corrective actions, $x_2(\xi)$.
- Algorithms :
  1. Primal and dual decomposition: Slyke and Wets (1969); Birge (1985); Rockafellar and Wets (1991); Mulvey and Ruszczyński (1995),
2. Approximation techniques: Romisch and Schultz (1991); Shapiro (1991); Shapiro and de Mello (2000); Robinson (1996); Higle and Sen (1996).

Chance Constrained Programming

- A particular variable/function lie within a target range with a certain probability.
- Prékopa (1993), discusses several approaches for solving chance-constrained models including gradient methods and the use of penalty functions.
- Hillier (1967) proposed a procedure for approximating chance constraints by linear constraints.
- Weintraub and Vera (1991) proposed a cutting plane approach for solving the deterministic equivalent of a CCP for the case of normally distributed random constrained coefficients.
- Ruszczyński (2002) reformulates the CCP problems as a Mixed Integer Programming problems.

Alternate Objective Functions

1. A function that maximizes the expected value of the objective function (the **E**-model),

$$\text{Max} \quad E_{\xi}\left[f(\mathbf{x}, \xi)\right]$$

2. A function that minimizes the generalized mean square of the objective function (the **V**-model),

$$\text{Min} \quad \left[f(\mathbf{x}, \xi) - E_{\xi}\left[f(\mathbf{x}, \xi)\right]\right]^2$$

and,

3. A function that maximizes the probability of satisfying an aspiration level of the objective function (the **P**-model),

$$\text{Max} \quad Pr\left[\ f(\mathbf{x}, \xi)\ \right] \left\{\begin{matrix}\geq \\ = \\ \leq\end{matrix}\right\} u$$

Alternate Objective Functions

1. A function that maximizes the expected value of the objective function (the **E**-model),

$$\text{Max} \quad E_{\xi}\left[f(\mathbf{x},\xi)\right]$$

2. A function that minimizes the generalized mean square of the objective function (the **V**-model),

$$\text{Min} \quad \left[f(\mathbf{x},\xi) - E_{\xi}\left[f(\mathbf{x},\xi)\right]\right]^2$$

and,

3. A function that maximizes the probability of satisfying an aspiration level of the objective function (the **P**-model),

$$\text{Max} \quad Pr\left[\ f(\mathbf{x},\xi)\ \right]\left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\}u$$

Alternate Objective Functions

1. A function that maximizes the expected value of the objective function (the **E**-model),

$$\text{Max} \qquad E_{\xi}\left[f(\mathbf{x},\xi)\right]$$

2. A function that minimizes the generalized mean square of the objective function (the **V**-model),

$$\text{Min} \qquad \left[f(\mathbf{x},\xi) - E_{\xi}\left[f(\mathbf{x},\xi)\right]\right]^2$$

and,

3. A function that maximizes the probability of satisfying an aspiration level of the objective function (the **P**-model),

$$\text{Max} \qquad Pr\left[\ f(\mathbf{x},\xi)\ \right]\left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\}u$$

Alternate Constraint formulation

1. Satisfying the expected value of constraints.

$$\begin{aligned} E_\xi[g_i(\mathbf{x},\xi)] &\leq 0, \qquad & i \in G^s \\ E_\xi[h_i(\mathbf{x},\xi)] &= 0, \qquad & i \in H^s \end{aligned}$$

2. Satisfying the constraints probabilistically.
    1. Individual Chance constraint
    2. Joint Chance constraint

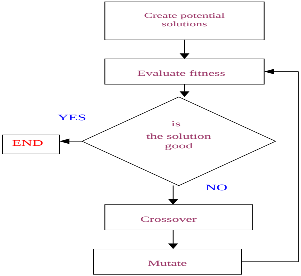
Alternate Constraint formulation

1. Satisfying the expected value of constraints.

$$\begin{aligned} E_{\xi}[g_i(\mathbf{x},\xi)] &\leq 0, & i &\in G^s \\ E_{\xi}[h_i(\mathbf{x},\xi)] &= 0, & i &\in H^s \end{aligned}$$

2. Satisfying the constraints probabilistically.
    1. Individual Chance constraint
    2. Joint Chance constraint

Genetic Algorithm



The C++ Classes

- *Objective function*: Single/Multiple functions, Deterministic/Stochastic.
- *Constraints*: Deterministic/Stochastic, Individual/Joint Chance constraints. The stochastic constraints, $g_i(\mathbf{x}, \xi)$ and $h_i(\mathbf{x}, \xi)$ can have one of the two mathematical structures
$$\begin{cases} \zeta_1(\mathbf{x}) + \tau\xi & \zeta_1 : \Re^n \rightarrow \Re, \tau \in \Re \\ \zeta_2(\mathbf{x}, \xi) & \zeta_2 : \Re^{n+m} \rightarrow \Re \end{cases}$$
- *Variables*: Pure integer ($\{0,1\}^n$), General integer (Z^n), continuous ($\Re^n$), and mixed integer ($\{0,1\}^{n_1} \bigcup Z^{n_2} \bigcup \Re^{n_3}$, $n = n_1 + n_2 + n_3$)

The C++ Classes

- *Objective function*: Single/Multiple functions, Deterministic/Stochastic.
- *Constraints*: Deterministic/Stochastic, Individual/Joint Chance constraints. The stochastic constraints, $g_i(\mathbf{x}, \xi)$ and $h_i(\mathbf{x}, \xi)$ can have one of the two mathematical structures
$$\begin{cases} \zeta_1(\mathbf{x}) + \tau\xi & \zeta_1 : \Re^n \rightarrow \Re, \tau \in \Re \\ \zeta_2(\mathbf{x}, \xi) & \zeta_2 : \Re^{n+m} \rightarrow \Re \end{cases}$$
- *Variables*: Pure integer ($\{0,1\}^n$), General integer (Z^n), continuous ($\Re^n$), and mixed integer ($\{0,1\}^{n_1} \bigcup Z^{n_2} \bigcup \Re^{n_3}$, $n = n_1 + n_2 + n_3$)

The C++ Classes

- *Objective function*: Single/Multiple functions, Deterministic/Stochastic.
- *Constraints*: Deterministic/Stochastic, Individual/Joint Chance constraints. The stochastic constraints, $g_i(\mathbf{x}, \xi)$ and $h_i(\mathbf{x}, \xi)$ can have one of the two mathematical structures
$$\begin{cases} \zeta_1(\mathbf{x}) + \tau\xi & \zeta_1 : \Re^n \longrightarrow \Re, \tau \in \Re \\ \zeta_2(\mathbf{x}, \xi) & \zeta_2 : \Re^{n+m} \longrightarrow \Re \end{cases}$$
- *Variables*: Pure integer ($\{0, 1\}^n$), General integer (Z^n), continuous ($\Re^n$), and mixed integer ($\{0, 1\}^{n_1} \bigcup Z^{n_2} \bigcup \Re^{n_3}$, $n = n_1 + n_2 + n_3$)

The features supported by of the framework.

	Problem components				
	Variables	**Constraints**		**Objective**	
Type	continuous	deterministic		deterministic	
	integer	stochastic	expectation	stochastic	expectation
	mixed integer		ICC & JCC		probabilistic
	binary				
	linear				
	non-linear				

Computation of the constraints and the objective functions

- $\Omega = \{\xi^1, \xi^2, \xi^i \dots \xi^{|\Omega|}\}$,
- ξ^i's $= \{\xi^i_1, \xi^i_2, \dots \xi^i_m\}$
- $\mathbf{x}^*$ be the solution to the problem, $P_{2SP-Anti}$

Computation of the constraints and the objective functions

- $\Omega = \{\xi^1, \xi^2, \xi^i \ldots \xi^{|\Omega|}\}$,
- ξ^i's $= \{\xi^i_1, \xi^i_2, \ldots \xi^i_m\}$
- $\mathbf{x}^*$ be the solution to the problem, $P_{2SP-Anti}$

Computation of the constraints and the objective functions

- $\Omega = \{\xi^1, \xi^2, \xi^i \ldots \xi^{|\Omega|}\}$,
- ξ^i's $= \{\xi^i_1, \xi^i_2, \ldots \xi^i_m\}$
- $\mathbf{x}^*$ be the solution to the problem, $P_{2SP-Anti}$

Computation of the constraints and the objective functions

- **E model**

$$E_\xi[\,\cdot\,] \quad = \quad \begin{cases} \frac{\sum_{j=1}^{|\Omega|} g_i\left(\mathbf{x}^*,\xi^j\right)}{|\Omega|}, & \forall \;\; g_i \in \bar{G}^s \\ \frac{\sum_{j=1}^{|\Omega|} h_i\left(\mathbf{x}^*,\xi^j\right)}{|\Omega|} & \forall \;\; h_i \in \bar{H}^s. \end{cases}$$

- **V model**

$$E_\xi[f^2(\mathbf{x},\xi)] - E_\xi^2[f(\mathbf{x},\xi)]$$

Computation of the constraints and the objective functions

- **E model**

$$E_{\xi}[\,\cdot\,] = \begin{cases} \frac{\sum_{j=1}^{|\Omega|} g_i\left(\mathbf{x}^*, \xi^j\right)}{|\Omega|}, & \forall \;\; g_i \in \bar{G}^s \\ \frac{\sum_{j=1}^{|\Omega|} h_i\left(\mathbf{x}^*, \xi^j\right)}{|\Omega|} & \forall \;\; h_i \in \bar{H}^s. \end{cases}$$

- **V model**

$$E_{\xi}[f^2(\mathbf{x}, \xi)] - E_{\xi}^2[f(\mathbf{x}, \xi)]$$

Computation of the constraints and the objective functions

$n_{\hat{g}_{i,r}}$ be the number of random vectors that satisfy the probabilistic inequality constraint

$$Pr[\,\cdot\,] \;=\; \frac{n_{\hat{g}_{i,r}}}{|\Omega|}$$

Computation of the constraints and the objective functions

$n_{\hat{g}_{i,r}}$ be the number of random vectors that satisfy the probabilistic inequality constraint

$$Pr[\,\cdot\,] \quad = \quad \frac{n_{\hat{g}_{i,r}}}{|\Omega|}$$

Outline

Notations

- V denote the set of individuals in the population,
- a given individual in the population is denoted by v,
- K be the population size,
- v_k denotes k^{th} individual in the population
- M be the total number of generations that the population evolves, and
- V_m denotes the m^{th} generation.

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Parameters

- Genetic Algorithm = Steady State
- Size of the population (K) = 10 × n
- Number of generation (M) = 20 × K
- Chromosome representation = floating point
- Probability of crossover (p_{cross}) =0.7
- Probability of mutation (p_{mut}) = 0.01
- Proportion replaced (p_{repl}) = 0.5

Settings

- *Initialisation*: uniform random numbers or approximate solution
- *Selection*: tournament selection
- *Crossover*: one-point crossover
- *Mutation*: Flip mutator
- *Stopping criteria*: *M* generations

Settings

- *Initialisation*: uniform random numbers or approximate solution
- *Selection*: tournament selection
- *Crossover*: one-point crossover
- *Mutation*: Flip mutator
- *Stopping criteria*: M generations

Settings

- *Initialisation*: uniform random numbers or approximate solution
- *Selection*: tournament selection
- *Crossover*: one-point crossover
- *Mutation*: Flip mutator
- *Stopping criteria*: M generations

Settings

- *Initialisation*: uniform random numbers or approximate solution
- *Selection*: tournament selection
- *Crossover*: one-point crossover
- *Mutation*: Flip mutator
- *Stopping criteria*: M generations

Settings

- *Initialisation*: uniform random numbers or approximate solution
- *Selection*: tournament selection
- *Crossover*: one-point crossover
- *Mutation*: Flip mutator
- *Stopping criteria*: M generations

Penalty Function for Constrained optimisation

The penalty function is computed as follows:

$$\rho(v,i) = \begin{cases} max\{0, \alpha - Pr[\,\cdot\,]\} & \text{for constraints of the type } Pr[\,\cdot\,] \geq \alpha \\ max\{0, Pr[\,\cdot\,] - \alpha\} & \text{for constraints of the type } Pr[\,\cdot\,] \leq \alpha \\ max\{0, E_\xi[g(v,\xi)]\} & \\ max\{0, |E_\xi[h(v,\xi)]|\} & \\ max\{0, g(v)\} & \\ max\{0, |h(v)|\} & \end{cases}$$

$\rho_{max}(i) = max_{v \in V_m}\{\rho(v,i)\}$

Penalty Function for Constrained optimisation

The penalty function is computed as follows:

$$\rho(v,i) = \begin{cases} max\{0, \alpha - Pr[\,\cdot\,]\} & \text{for constraints of the type } Pr[\,\cdot\,] \geq \alpha \\ max\{0, Pr[\,\cdot\,] - \alpha\} & \text{for constraints of the type } Pr[\,\cdot\,] \leq \alpha \\ max\{0, E_\xi[g(v,\xi)]\} & \\ max\{0, |E_\xi[h(v,\xi)]|\} & \\ max\{0, g(v)\} & \\ max\{0, |h(v)|\} & \end{cases}$$

$\rho_{max}(i) = max_{v \in V_m}\{\rho(v,i)\}$

Penalty Function for Constrained optimisation

Scoring Functions

- Feasibility Scoring function(s)
- Optimality Scoring function

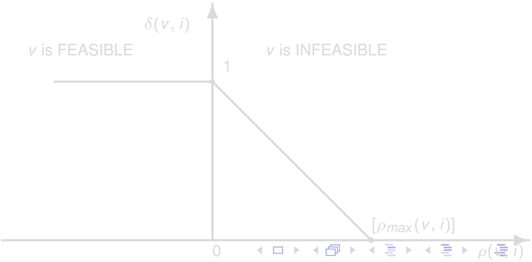
Penalty Function for Constrained optimisation

Feasibility scoring function

Degree of constraint satisfaction

$$\delta(v,i) = \begin{cases} 1 & \text{if constraint } i \text{ is satisfied,} \\ \frac{\rho max(i) - \rho(v,i)}{\rho max(i)} & \text{otherwise,} \end{cases}$$

therefore $\delta(v,i) : V \times \Gamma \rightarrow [0,1]$.



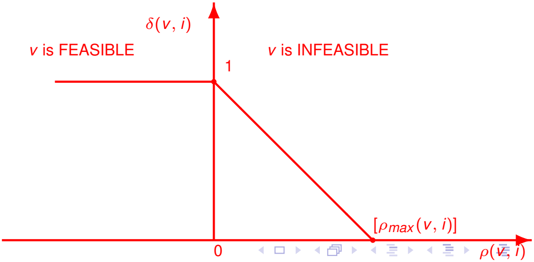
Penalty Function for Constrained optimisation

Feasibility scoring function

Degree of constraint satisfaction

$$\delta(v,i) = \begin{cases} 1 & \text{if constraint } i \text{ is satisfied,} \\ \frac{\rho max(i) - \rho(v,i)}{\rho max(i)} & \text{otherwise,} \end{cases}$$

therefore $\delta(v,i) : V \times \Gamma \longrightarrow [0,1]$.



Penalty Function for Constrained optimisation

Feasibility scoring functions

- *Additive*

$$\nu_{\oplus}(v) = \frac{\sum_{i \in \Gamma} \delta(v, i)}{\Gamma},$$

- *Multiplicative*

$$\nu_{\otimes}(v) = \prod_{i \in \Gamma} \delta(v, i).$$

Penalty Function for Constrained optimisation

Feasibility scoring functions

- *Additive*

$$\nu_{\oplus}(v) = \frac{\sum_{i \in \Gamma} \delta(v,i)}{\Gamma},$$

- *Multiplicative*

$$\nu_{\otimes}(v) = \prod_{i \in \Gamma} \delta(v,i).$$

Optimality scoring function

$\gamma(v)$ = objective function value for a given individual v,
Define γ_{max} and γ_{min}

Degree of optimality satisfaction

$$\theta(v) = \begin{cases} \frac{\gamma(v)}{\gamma_{max}} & \text{for maximisation problems} \\ \frac{\gamma_{min}}{\gamma(v)} & \text{for minimisation problems} \end{cases}$$

Optimality scoring function

$\gamma(v)$ = objective function value for a given individual v,
Define γ_{max} and γ_{min}
Degree of optimality satisfaction

$$\theta(v) = \begin{cases} \frac{\gamma(v)}{\gamma_{max}} & \text{for maximisation problems} \\ \frac{\gamma_{min}}{\gamma(v)} & \text{for minimisation problems} \end{cases}$$

Fitness Function

$$\Psi(\nu_*(v),\theta(v)) = \nu_*(v)\times\theta(v)$$

$$\Psi_\lambda(\nu_*(v),\theta(v)) = \nu_*(v)^{(1-\lambda_m)}\times\theta(v)^{\lambda_m}$$

, $\lambda\in\{\lambda_{min},\lambda_{max}\}$

$$\lambda_{m+1} = \lambda_m - \frac{\lambda_\triangle}{M}$$

where $\lambda_1 = \lambda_{max}$ and $\lambda_\triangle = \frac{(\lambda_{max}-\lambda_{min})}{M}$.

Fitness Function

$$\Psi(\nu_*(v),\theta(v)) = \nu_*(v) \times \theta(v)$$

$$\Psi_\lambda(\nu_*(v),\theta(v)) = \nu_*(v)^{(1-\lambda_m)} \times \theta(v)^{\lambda_m}$$

$, \lambda \in \{\lambda_{min}, \lambda_{max}\}$

$$\lambda_{m+1} = \lambda_m - \frac{\lambda_\triangle}{M}$$

where $\lambda_1 = \lambda_{max}$ and $\lambda_\triangle = \frac{(\lambda_{max} - \lambda_{min})}{M}$.

Fitness Function

$$\Psi(\nu_*(v), \theta(v)) = \nu_*(v) \times \theta(v)$$

$$\Psi_\lambda(\nu_*(v), \theta(v)) = \nu_*(v)^{(1-\lambda_m)} \times \theta(v)^{\lambda_m}$$

, $\lambda \in \{\lambda_{min}, \lambda_{max}\}$

$$\lambda_{m+1} = \lambda_m - \frac{\lambda_\triangle}{M}$$

where $\lambda_1 = \lambda_{max}$ and $\lambda_\triangle = \frac{(\lambda_{max} - \lambda_{min})}{M}$.

Outline

1. Introduction
2. Alternate formulations of CCP models
3. An overview of Genetic Algorithm
4. Computational Framework
    - GA Controls
    - Simulation
5. Description of the test problems
    - Analysis of the framework
6. Discussion and Conclusion

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times\ |J^s| \times\ K \times\ 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times |J^s| \times K \times 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times |J^s| \times K \times 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times |J^s| \times K \times 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times |J^s| \times K \times 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times |J^s| \times K \times 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega|\times\ |J^s|\times\ K\times\ \ 20\times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Settings

- Any probability distribution, Mixed distribution (Sum/Product),
- Stochastic functions

$$J^s = \begin{cases} H^s \bigcup G^s & \text{if the objective determinist} \\ H^s \bigcup G^s \bigcup \text{Objective function} & \text{otherwise.} \end{cases}$$

- Evaluating a population $|\Omega| \times \ |J^s| \times \ K \times \ 20 \times K$
- These random vectors are used to evaluate the stochastic optimisation problem for each of the N runs of the simulation.

Simulation Settings

Parameter	Value
$\lvert\Omega\rvert$	300
Number of runs (N)	100

Table: Control parameters for the Simulation.

Substituting, we get

$$6 \times 10^5 \times n^2 \times |J^s|,$$

where n is the cardinality of the solution vector.

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Test Problems

1. *Fractional programming problem*: deterministic optimisation problem,
2. *Newsboy problem*: Individual Chance constraint programming model with the right-hand side as a random variable,
3. *Feeder-mix and the Kilosa farmer*: Individual Chance constraint with the coefficients as the random variables,
4. *Pension fund*: an Expected Value problem, an Individual Chance constrained problem, and a Joint chance constrained problem,
5. *Nonlinear and Non-convex optimisation problem,*
6. *Probabilistic objective and individual chance constraints.*

Problem 1 : Fractional programming problem

$$P_{FP} \qquad \text{Max} \qquad \frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3} + \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}$$

for constraints

$$\begin{aligned} x_1 + x_2 - x_3 &\leq 1, \\ -x_1 + x_2 - x_3 &\leq -1, \\ 12x_1 + 5x_2 + 12x_3 &\leq 34.8, \\ 12x_1 + 12x_2 + 7x_3 &\leq 29.1, \\ -6x_1 + x_2 + x_3 &\leq -4.1, \\ x_i \geq 0, i &= 1, 2, 3. \end{aligned}$$

Problem 2: The News vendor problem

$$
\begin{array}{llrcl}
P_{NB} & \text{Max} & \pi x - cx & & \\
& & \text{for constraints} & & \\
& & P\{h(x-\xi) \leq 5\} & \geq & 0.90 \quad (\textit{Wastage Cost}) \\
& & P\{\pi(\xi - x) \leq 27\} & \geq & 0.90 \quad (\textit{Shortage Cost})
\end{array}
$$

Problem 3: Feed Mixer Problem

Van de Panne and Popp presented a CCP formulation of feed mixer problem.

$$
\begin{array}{llrcll}
P_{FM} & \text{Max} & 24.55x_1 + 26.75x_2 + 39.00x_3 + 40.50x_4 & & & \\
 & & 2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 & \geq & 5, & (Fat\ \ constraint) \\
 & & Pr\left[\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 \geq 21\right] & \geq & p, & (Protein\ \ constraint) \\
 & & x_1 + x_2 + x_3 + x_4 & = & 1, & \\
 & & x_1, x_2, x_3, x_4 & \geq & 0 &
\end{array}
$$

where

- p (=0.95) equals the probability level,
- ξ_1, ξ_2, ,ξ_3, and ξ_4, have normal distributions $\mathcal{N}(12, 0.53)$, $\mathcal{N}(11.9, 0.44)$, $\mathcal{N}(41.8, 4.5)$, and $\mathcal{N}(52.1, 0.79)$ respectively.

Problem 4: Pension Fund problem

The pension fund of a company has to meet its liabilities for the next T years. The liabilities shall be covered by investing an initial capital B in P different types of bonds.
γ_i denote the cost for bond i, $i \in P$,
β_j denote the payments for the year j, $j \in T$,
and, α_{ij} denote the yield per bond of type i in the year j.
The liquidity constraint is for the year j :

$$\underbrace{B - \sum_{i=1}^{|P|} \gamma_i x_i}_{\substack{\text{cash after}\\ \text{buying bonds}}} + \underbrace{\sum_{k=1}^{j} \sum_{i=1}^{|P|} \alpha_{ik} x_i}_{\substack{\text{cumulative}\\ \text{yields of bonds}}} - \underbrace{\sum_{k=1}^{j} \beta_k}_{\substack{\text{cumulative}\\ \text{payment}}} \geq 0$$

Problem 4: Pension Fund problem

$a_{ij} = \sum_{k=1}^{j} \alpha_{ik} - \gamma_i$, $b_j = \sum_{k=1}^{j} \beta_k - B$. The liquidity constraint for the terminal time-period:

$$\sum_{i=1}^{|P|} a_{i,T} x_i - b_T .$$

The deterministic optimisation problem is

$$\begin{array}{lcccc} Max & \displaystyle\sum_{i=1}^{|P|} a_{i,T} x_i - b_T & & & \\ & \text{for constraints} & & & \\ & \displaystyle\sum_{i=1}^{|P|} a_{ij} x_i & \geq & b_j & j = 1, \ldots T \end{array}$$

Problem 4: Pension Fund problem

Assume that the liability streams are random variables η_j, expected value = β_j (the deterministic payments), and standard deviation = $500j$ (the standard deviation increases with time). The cumulative payment at the end of j years, $\xi_j = \sum_{k=1}^{j} \eta_k$.
The optimisation problems with individual chance constraint is

$$
\begin{array}{lllll}
P_{PF-ICC} & \text{Max} & \displaystyle\sum_{i=1}^{|P|} a_{i,T} x_i & & \\
& Pr\left\{ \displaystyle\sum_{i=1}^{|P|} a_{ij} x_i \geq \xi_j \right\} & \geq & p & (j = 1, \ldots, T)
\end{array}
$$

Problem 4: Pension Fund problem

Problem can be extended by imposing the constraint that the liabilities must be satisfied *jointly* for all the time periods with 95% probability.

$$P_{PF-JCC} \qquad \text{Max} \qquad \sum_{i=1}^{|P|} a_{i,T} x_i$$

$$Pr\{\sum_{i=1}^{|P|} a_{ij} x_i \geq \xi_j \;\; (j = 1, \ldots, T)\} \quad \geq \quad p.$$

Problem 5: Kilosa farmer problem

Let
x_m= acreage of maize in hectacres;
x_s= acreage of sorghum in hectacres;
ξ_m = random yield per hectacre of maize (in 100 Kgs);
ξ_s = random yield per hectacre of sorghum (in 100 Kgs);
ξ = random total rainfall during the growing season (mm);
ϵ_m = white noise in the yield of maize;
ϵ_s = white noise in the yield of sorghum.
It is known that

1. 100 kgs of maize contains 2.8×10^5 Kcal and 6.4 kg of protein
2. 100 kgs of sorghum contains 2.8×10^5 Kcal and 8 kg of protein.

Problem 5: Kilosa farmer problem

Regression analysis applied to empirical data leads to the following model for the random yields.

$$\begin{aligned} \xi_m &= 0.020\xi - 1.65 + \epsilon_1 \\ \xi_s &= 0.008\xi + 5.92 + \epsilon_2 \end{aligned}$$

where $\xi \sim \mathcal{N}(515.5, \sqrt{18769})$, $\epsilon_1 \sim \mathcal{N}(0, \sqrt{100})$, $\epsilon_2 \sim \mathcal{N}(0, \sqrt{100})$

Problem 5: Kilosa farmer problem

The optimisation problem is :

$$\begin{array}{llcl} P_{KF} & \text{Min} & x_m + x_s & \\ & Pr\{2.8\xi_m x_m + 2.8\xi_s x_s \geq 44\} & \geq & \alpha_c \\ & Pr\{6.4\xi_m x_m + 8.0\xi_s x_s \geq 89\} & \geq & \alpha_p . \end{array}$$

Problem 6: Non-Convex and Non-Linear problem

$$
\begin{array}{lrcl}
P_{NCNL} & \text{Min} -9x_1^2 + 10x_1x_2 - 50x_1 + 8x_2 + 460 & & \\
 & \text{for constraints} & & \\
 & Pr\{x_1 - 0.2768x_2^2 + .235x_2 \leq 3.718 + \xi_1\} & \geq & \alpha \\
 & Pr\{x_1 + 0.019x_2^3 - 0.446x_2^2 + 3.98x_2 \leq 15.854 + \xi_2\} & \geq & \alpha
\end{array}
$$

where $\xi_1 \sim \mathcal{N}(\mu_{\xi_1}, \sigma_{\xi_1})$ and $\xi_2 \sim \mathcal{N}(\mu_{\xi_2}, \sigma_{\xi_2})$, and α = 2.5%. For $\xi_1 \sim \mathcal{N}(0, 0.01)$ and $\xi_2 \sim \mathcal{N}(0, 0.02)$, there are two optimal solutions to the problem, namely {5,3} and {3,0}. While for, $\xi_1 \sim \mathcal{N}(0, 0.01)$ and $\xi_2 \sim \mathcal{N}(-5, 0.02)$, there is only one optimal solution to the problem, namely {3,0}. We define two variants of the problem P_{NCNL}.

Problem 6: Non-Convex and Non-Linear problem

$$
\begin{array}{lrcl}
P_{NCNL_1} & \text{Min } -9x_1^2 + 10x_1x_2 - 50x_1 + 8x_2 + 460 & & \\
& \text{for constraints} & & \\
& Pr\{x_1 - 0.2768x_2^2 + .235x_2 \leq 3.718 + \xi_1\} & \geq & 0.025 \\
& Pr\{x_1 + 0.019x_2^3 - 0.446x_2^2 + 3.98x_2 \leq 15.854 + \xi_2\} & \geq & 0.025
\end{array}
$$

where $\xi_1 \sim \mathcal{N}(0, 0.01)$, $\xi_2 \sim \mathcal{N}(0, 0.02)$, and

Problem 6: Non-Convex and Non-Linear problem

$$
\begin{array}{llll}
P_{NCNL_2} & \text{Min } -9x_1^2 + 10x_1x_2 - 50x_1 + 8x_2 + 460 & & \\
& \text{for constraints} & & \\
& Pr\{x_1 - 0.2768x_2^2 + .235x_2 \leq 3.718 + \xi_1\} & \geq & 0.025 \\
& Pr\{x_1 + 0.019x_2^3 - 0.446x_2^2 + 3.98x_2 \leq 15.854 + \xi_2\} & \geq & 0.025
\end{array}
$$

where $\xi_1 \sim \mathcal{N}(0, 0.01)$, $\xi_2 \sim \mathcal{N}(-5, 0.02)$.

Problem 7: Probabilistic Objective

$$\begin{aligned} \text{Max } Pr\{\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 \geq 2.27\} &\geq 0.90 \\ \text{for constraints} & \\ Pr\{\eta_1 x_1^2 + \eta_2 x_2^2 + \eta_3 x_3^2 < 8\} &\geq 0.80 \\ Pr\{\tau_1 x_1^3 + \tau_2 x_2^3 + \tau_3 x_3^3 < 15\} &\geq 0.85 \end{aligned}$$

where $\xi_1 \sim \mathcal{U}(1,2)$, $\xi_2 \sim \mathcal{N}(2,1)$, $\xi_3 \sim Exp(1)$, $\eta_1 \sim \mathcal{U}(2,3)$, $\eta_2 \sim \mathcal{N}(2,1)$, $\eta_3 \sim Exp(2)$, $\tau_1 \sim \mathcal{U}(3,4)$, $\tau_2 \sim \mathcal{N}(3,1)$, $\tau_3 \sim Exp(3)$.

AMD Athlon processor of speed 1.40 Ghz and 256 MB RAM and Windows XP operating system

The Pseudocode

Algorithm 5.1: GAANDSIMULATION()

```
Initialise $K$, $M$, $|\Omega|$, $p_{cross}$, $p_{mut}$, $p_{repl}$, $\lambda_{min}$, $\lambda_{max}$
$m \leftarrow 1$, $\lambda_m \leftarrow \lambda_{max}$
Generate the initial population $V_m$
while $m \leq M$
    for all $v \in V_m$
        comment: Evaluate the constraints.
        Calculate $\rho(v, i)$, $\delta(v, i)$
        Calculate $\nu(v)$
        comment: Evaluate the objective.
        Calculate $\theta(v)$
        Calculate $\Phi(\nu_*, \theta)$
    Perform one-point crossover by selecting and mating two individuals.
    Perform gaussian mutation.
    Replace a fraction of the individuals in the current population.
    $m \leftarrow m + 1$
    Update $\lambda_m$
```

Outline

1. Introduction
2. Alternate formulations of CCP models
3. An overview of Genetic Algorithm
4. Computational Framework
    - GA Controls
    - Simulation
5. Description of the test problems
    - Analysis of the framework
6. Discussion and Conclusion

Statistical Analysis

1. Do the representation and the framework provide consistent solutions to the CCP problems ?
2. Do the additive and multiplicative feasibility scoring functions influence differently ?
3. Does the strategy of varying the importance to the feasibility and optimality scoring functions over generations have significant influence over the quality of the solution ?

Statistical Analysis

1. Do the representation and the framework provide consistent solutions to the CCP problems ?
2. Do the additive and multiplicative feasibility scoring functions influence differently ?
3. Does the strategy of varying the importance to the feasibility and optimality scoring functions over generations have significant influence over the quality of the solution ?

Statistical Analysis

1. Do the representation and the framework provide consistent solutions to the CCP problems ?
2. Do the additive and multiplicative feasibility scoring functions influence differently ?
3. Does the strategy of varying the importance to the feasibility and optimality scoring functions over generations have significant influence over the quality of the solution ?

Solutions values

Problem	Objective	Solution
Fractional Programming	2.471	{1,0,0}
News Boy	5.39	{49}
Feeder-Mix	29.89	{0.6359,0,0.3127,0.0515}
Pension Fund (Exp)	127,332	{31.1,55.5,147.3 }
Pension Fund (ICC)	103,925	{62.8,72.6,101.1 }
Pension Fund (JCC)	98,160	{65.8, 83.7,86.2}
Kilosa Farmer	1.81	{1.81,0}
Non-Linear (1)	159	{5,3}
Non-Linear (2)	229	{3,0}
Prob. Obj	2.27	{1.458, 0.490, 0.811}

Table: The values of the objective function and the solutions for the test problems.

Consistency of the solution

	Problem									
	P_{FP}	P_{NB}	P_{FM}	P_{PF-EXP}	P_{PF-ICC}	P_{PF-JCC}	P_{KF}	P_{NCNL_1}	P_{NCNL_2}	$P_{ProbObj}$
$H_0^{\oplus,0}$	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*
$H_0^{\otimes,0}$	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*	*CA*
$H_0^{\oplus,\lambda}$	*CR*	*CR*	*CA*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*
$H_0^{\otimes\lambda}$	*CR*	*CR*	*CA*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*

Table: The Null Hypothesis (at 5% level of significance) for the consistency of the solution.

The system does provide a consistent solution.

Impact of the feasibility scoring function

Our null hypothesis is that there is no difference between the two feasibility scoring functions.

	Problem								
	P_{FP}	P_{NB}	P_{FM}	P_{PF-EXP}	P_{PF-ICC}	P_{PF-JCC}	P_{KF}	P_{NCNL-1}	P_{NCNL-2}
$H_0^{\oplus;\otimes}$	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*
$H_0^{\oplus,\lambda;\otimes,\lambda}$	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*	*CR*

Table: The Null Hypothesis (at 5% level of significance) for the impact of the feasibility scoring function.

Impact of the feasibility scoring function

- (Surprisingly) cannot reject the null hypothesis that they are the same,
- Population converges in fewer generations on using multiplicative scoring function,
- Multiplicative scoring function not appropriate for problems having large numbers of constraints and/or a narrow feasible region.

Impact of the varying fitness function

Our null hypothesis is that there is no difference between the two two variants of the fitness functions.

	Problem								
	P_{FP}	P_{NB}	P_{FM}	P_{PF-EXP}	P_{PF-ICC}	P_{PF-JCC}	P_{KF}	P_{NCNL-1}	P_{NCNL-2}
$H_0^{\oplus,0;\oplus,\lambda}$	CA	CA	CR	CR	CR	CR	CR	CR	CR
$H_0^{\otimes,0;\otimes,\lambda}$	CA	CA	CA	CR	CR	CR	CR	CR	CR

Table: The Null Hypothesis (at 5% level of significance) for the impact of the varying fitness function.

The two methods for calculating the fitness act differently.

Problem	Average time in seconds			
	Additive Penalty		Multiplicative Penalty	
	with λ update	without λ update	with λ update	without λ update
P_{FP}	.26	.24	.26	.24
P_{NB}	3.53	3.53	3.52	3.52
P_{FM}	96.38	96.42	96.39	96.46
P_{PF-EXP}	.5	.5	.5	.5
P_{PF-ICC}	213.9	213.47	260.37	218.8
P_{PF-JCC}	184.82	184.23	191	189.74
P_{KF}	49.52	49.55	49.52	49.51
P_{NCNL_1}	12.61	12.41	12.57	12.43
P_{NCNL_2}	12.56	12.57	12.58	12.56
$P_{ProbObj}$	104.98	102.81	104.92	102.61

Table: Average time for a single run.

Problem	Objective function value			
	Additive Penalty		Multiplicative Penalty	
	95%	99%	95%	99%
P_{FP}	5.33 to 5.41	5.31 to 5.42	5.30 to 5.38	5.29 to 5.39
P_{NB}	5.78 to 6.11	5.71 to 6.18	5.72 to 6.10	5.65 to 6.18
P_{FM}	40.04 to 45.46	38.93 to 46.57	39.96 to 44.31	39.07 to 45.21
P_{PF-EXP}	124213 to 126055	123835 to 126433	124746 to 126298	124428 to 126616
P_{PF-ICC}	98636.8 to 100171	98322 to 100486	9767.1 to 100810	97033.1 to 101453
P_{PF-JCC}	96644.7 to 97907.7	96385.6 to 98166.8	96247.4 to 97906.6	95907 to 98247
P_{KF}	1.67 to 2.3	1.53 to 2.43	1.79 to 2.25	1.69 to 2.34
P_{NCNL_1}	134 to 172	126 to 180	135 to 169	128 to 176
P_{NCNL_2}	201.2 to 245	192.08 to 254.11	203 to 243	194.67 to 251.32
$P_{ProbObj}$	2.56 to 2.9	2.49 to 2.97	2.36 to 2.92	2.24 to 3.03

Table: Confidence intervals on the objective function value using the fitness function $\Psi(\cdot)$.

Problem	**Solution vectors**	
	Additive Penalty	
	95%	99%
P_{FP}	{.81,.82,0} to {.82,.86,0}	{.81,.74,0} to {.82,.86,0}
P_{NB}	{ 48.33} to {59.86}	{ 45.96} to {62.23}
P_{FM}	{-.07,-.07,.44,.42} to {.1,.105,.66,.59}	{-.11,-.10,.40,.38} to {.13,.14,.70,.63}
P_{PF-EXP}	{34.49,52.02,143.28} to {40.15,55.15,145.78}	{33.33,51.37,142.76} to {41.32,55.8,146.29}
P_{PF-ICC}	{65.87,72.26,90.81} to {68.28,77.05,96.56}	{65.37,71.26,89.63} to {68.78,78.03,97.74}
P_{PF-JCC}	{66.43,80.92,85.26} to {68.78,83.38,86.44}	{65.95,80.42,85.02} to {69.27,83.88,86.68}
P_{KF}	{-.56,-.49} to {2.49,.92}	{-1.19,-.78} to {3.11,1.21}
P_{NCNL_1}	{4.69, 2.97} to {5.00,3.39}	{4.67,2.93} to {5.03,3.43}
P_{NCNL_2}	{3.08,-0.042} to {3.82,0.94}	{3.01,-0.13} to {3.89,1.03}
$P_{ProbObj}$	{-.42,-.13,-.04} to {2.44,1.77,1.95}	{-1,-.52,-.45} to {3.03,2.16,2.36}

Problem	**Solution vectors**	
	Multiplicative Penalty	
	95%	99%
P_{FP}	{.81,.82,0 } to {.81,.85,0}	{.82,.82,0} to {.82,.86,0}
P_{NB}	{47.43} to { 60.20}	{44.81} to {62.81}
P_{FM}	{-.04,-.03,.46,.42} to {.04,.05,.64,.58}	{-.06,-.05,.42,.39 } to {.06,.07,.68,.62 }
P_{PF-EXP}	{33.58,52.22,144.31} to {39.28,55.47,145.71}	{32.41,51.55,144.03} to {40.45,56.13,146}
P_{PF-ICC}	{63.37,70.21,92.42} to {70.67,74.43,98.14 }	{61.87,69.35,91.24} to {72.17,75.29,99.32}
P_{PF-JCC}	{65.80,79.78,85.48} to {69.10,83.12,86.96}	{65.12,79.09,85.18} to {69.78,83.80,87.26}
P_{KF}	{1.14,.07} to {2.37,1.90}	{.89,-.29} to {2.63,2.28}
P_{NCNL_1}	{4.79,2.88} to {5.06,3.27}	{4.77,2.85} to {5.08,3.30}
P_{NCNL}	{2.92,-0.031} to {3.54,0.87}	{2.87,-0.12} to {3.59,0.96}

Problem	Objective function value			
	Additive Penalty		Multiplicative Penalty	
	95%	99%	95%	99%
P_{FP}	2.40 to 2.48	2.39 to 2.50	2.41 to 2.48	2.39 to 2.49
P_{NB}	5.38 to 5.44	5.37 to 5.45	5.39 to 5.47	5.38 to 5.49
P_{FM}	38.62 to 40.23	38.29 to 40.56	38.04 to 40.36	37.57 to 40.83
P_{PF-EXP}	121495 to 130552	119636 to 132411	121210 to 130511	119301 to 132419
P_{PF-ICC}	94655.5 to 106052	92317.4 to 108390	91426.7 to 101513	89357.3 to 103583
P_{PF-JCC}	96151 to 98639.8	95640.3 to 99150.5	96644.7 to 98256.5	96314 to 98587.2
P_{KF}	1.8 to 2	1.76 to 2.04	1.8 to 2	1.75 to 2.04
P_{NCNL_1}	143 to 162	139.04 to 165.95	145 to 161.2	141.6 to 164.57
P_{NCNL_2}	214.2 to 235.12	209.84 to 239.47	216.3 to 234.9	212.4 to 238.7
$P_{ProbObj}$	2.24 to 2.99	2.08 to 3.14	2.21 to 3	2.04 to 3.16

Table: Confidence intervals on the objective function value using the fitness function $\Psi_\lambda(\cdot)$.

Problem	**Solution vectors**	
	Additive Penalty	
	95%	99%
P_{FP}	{.94,-.03,-.05} to { 1,.05,.1 }	{.94,-.05,-.08} to {1,.06,.13 }
P_{NB}	{48.29} to {50.1}	{47.91} to {50.48}
P_{FM}	{0,.01,.59,.29} to {0.02,.06,.66,.39}	{0, 0.01,.57,.26} to {.02,.07,.68,.41}
P_{PF-EXP}	{29.15,50.67,138.42} to {37.55,57.37,154.31}	{27.43,49.30,135.16} to {39.28,58.74,157.58}
P_{PF-ICC}	{64.60,64.27,89.33} to {73.56,75.13,106 }	{62.76,62.03,85.92} to {75.40,77.36,109.42}
P_{PF-JCC}	{65.93,79.30,85.22} to {69.21,83.60,87.78}	{65.26,78.42,84.70} to {69.88,84.48,88.30}
P_{KF}	{-.579,-0.513} to {2.51,0.961}	{-1.213,-0.816} to {3.15,1.26}
P_{NCNL_1}	{4.86,2.91}to {5.08,3.10}	{4.84, 2.90} to {5.10,3.12}
P_{NCNL_2}	{2.84,0.13} to {3.49,0.48}	{2.71,0.061} to {3.62,0.56}
$P_{ProbObj}$	{-.27,-.32,-.34} to {3.6,1.8,1.69}	{-1.06,-.76,-.76} to {4.4,2.24,2.11}

	Multiplicative Penalty	
	95%	99%
P_{FP}	{.94,-.02,-.05} to {1,.04,.10 }	{.92,-.04,-.09} to {1.02,.05,.138 }
P_{NB}	{48.31 } to {50.48}	{47.87} to {50.92}
P_{FM}	{-0.04,-.04,.54,.30} to {.07,.08,.70,.40}	{-.07,-.07,.5,.29 } to {.1,.11,.73,.42 }
P_{PF-EXP}	{30.02,50.80,138.15} to {36.86,56.91,154.45}	{28.62,49.54,134.80} to {38.27,58.17,157.80}
P_{PF-ICC}	{65.87,66.44,85.65} to {71.95,72.80,102.16}	{64.62,65.14,82.26}to {73.2,74.1,105.54 }
P_{PF-JCC}	{65.85,80.06,85.57} to {68.99,83.19,87.37}	{65.21,79.43,85.21} to {69.63,83.83,87.73 }
P_{KF}	{1.13,.12} to {2.36,1.89}	{.88,-.23} to {2.61,2.26}
P_{NCNL_1}	{4.90,2.90} to {5.06,3.08}	{4.89,2.88} to {5.07,3.09}
P_{NCNL_2}	{2.78,-0.164} to {3.24,0.40}	{2.73,-0.28} to {3.29, 0.48}
$P_{ProbObj}$	{-.03,-.04,-.02} to {1.26,1.13,.93}	{-.29,-.28,-.22} to {1.53,1.36,1.12}

Table: Confidence intervals on the decision variables using the fitness function $\Psi_{\lambda}(\cdot)$.

Summary

- GA for non-linear and non-convex optimisation problems, Simulation for generating the random variables,
- Extend GA for constrained optimisation,
- Verified the versatility of the combined framework.

Future work

- Stochastic Integer Programming problems,
- Parallel optimisation,
- Variance reduction,
- Coupling with a modelling system.

E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *J. Royal Statistical Society*, Series B:173–184, 1955.

A. Ben-Tal and A. Nemirovski. Robust optimization methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

J.R. Birge. Decomposition and partitioning methods for multi-stage stochastic linear programs. *Operations Research*, 33:989–1007, 1985.

A. Charnes and W.W. Cooper. Chance constrained programming. *Management Science*, 5:73–79, 1959.

G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

M.A.H. Dempster and G. Consigli. The calm stochastic programming model for dynamic asset-liability management. In *World Wide Asset and Liability Modelling*, pages 464–500. Cambridge University Press, 1998.

M.A.H. Dempster, N. Hicks Pedrón, E.A. Medova, J.E. Scott, and A Sembos. Planning logistic operations in the oil industry. *Journal of Operational Research Society*, 51(11): 1271–1288, 2000.

C.L. Dert. *Asset Liability Management for Pension Funds: A Multistage Chance Constrained Programming Approach*. PhD thesis, Erasmus University, Rotterdam, The Netherlands, 1995.

S. E. Elmaghraby, H. Soewandi, and Ming-Jong Yao. Chance-constrained programming in activity networks: A critical evaluation. *European Journal of Operational Research*, 131(1):440–458, June 2001.

G.D. Eppen, R.K. Martin, and L. Schrage. A scenario approach to capacity planning. *Operational Research*, 37(4):517–525, 1989.

L. El Ghaoui and H. Lebret. Robust solutions to least-square problems with uncertain data matrices. *SIAM Journal of Matrix Analysis Application*, 18:10351064, 1997.

J.L. Higle and S. Sen. *Stochastic Decomposition : A Statistical Method for large-scale Stochastic Linear Programming*. Kluwer Academic Publishers, University of Arizona, 1996.

F.S. Hillier. Chance-constrained programming with 0-1 or bounded continuous decision variables. *Management Science*, 14:34–57, 1967.

M.G. Ierapetritou, J. Acevedo, and E.N. Pistikopoulos. An optimization approach for process engineering problems under uncertainty. *Computers and Chemical Engineering*, 20:703–709, 1996.

Athanasios Kampas and Ben White. Probabilistic programming for nitrate pollution control: Comparing different probabilistic constraint approximations. *European Journal of Operational Research*, 147(1):217–228, May 2003.

G. Laporte, F.V. Louveaux, and L. Van Hamme. Exact solution to a location problem with stochastic demands. *Transportation Science*, 28:95–103, 1994.

S.A. MirHassani, C.A. Lucas, G. Mitra, E. Messina, and C.A. Poojari. Computational solution of capacity planning models under uncertainty. *Parallel Computing*, 26:511–538, 2000.

J.M. Mulvey and A. Ruszczyński. A new scenario decomposition method for large-scale stochastic optimisation. *Operational Research*, 43(3):477–490, 1995.

F.H. Murphy, S. Sen, and A. L. Soyster. Electric utility capacity expansion planning with uncertain load forecasts. *AIIE Transaction*, 14:52–59, 1982.

A. Prékopa. Programming under probabilistic constraint and maximizing a probability under constraints. Rrr 35-93, Center for Operations Research, Rutgers university, P.O. box 5062, New Brunswick, New Jersey, December 1993.

J. R. Robinson. Loaded questions: New approaches to utlity forecasting. *Energy policy*, 16(1):58–68, 1988.

S.M. Robinson. Analysis of sample path optimisation. *Mathematics of Operations Research*, (21):513–528, 1996.

R.T. Rockafellar and R.J.B. Wets. Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

W. Romisch and R. Schultz. Distribution sensitivity in stochastic programming. *Mathematical Programming*, (50):197–226, 1991.

Andrzej Ruszczyński. Probabilistic programs with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming*, (93):195–215, 2002.

S. Sen, R.D. Doverspike, and S. Cosares. Network planning with random demand. *Telecommunication Systems*, 3:11–30, 1994.

A. Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, (30):169–186, 1991.

A. Shapiro and T. Homem de Mello. On rate of convergence of monte carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11(1):70–86, 2000.

R. Van Slyke and R. J-B. Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

Jeffrey Smith. Optimizing platform survivability using a chance constrained linear program. In *US Army Ground Vehicle Survivability Symposium*, Survivability/lethality analysis directorate, munitions and Platform branch, White sands Missile range, NM, March 1999. U.S. Army Research Laboratory.

S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.

A. Tomasgard, S. Dye, S.W. Wallace, J.A. Audestad, and L. Stougie. Modelling aspects of distributed processing in telecommunication networks. *Annals of Operations Research*, 82:161–184, 1998.

M.H. van der Vlerk. Integrated chance constraints in an alm model for pension funds. Technical Report 03A21, University of Groningen, Research Institute SOM (Systems, Organisations and Management), 2003.

A. Weintraub and J. Vera. A cutting plane approach for chance constrained linear program. *Operations Research*, 39(5): 770–776, 1991.