AORDA®
American Optimal Decisions, Inc.

Breakthrough technologies in risk management and optimization

Introduction to Portfolio Safeguard (PSG)

Overview: Portfolio Safeguard (PSG)

- Portfolio Safeguard (PSG) is a product of American Optimal Decisions, Inc. (AOD) a privately held company based in Gainesville, Florida, USA.

- AOD provides consulting services and software solutions in risk management, financial engineering, and investment strategies.

- PSG is a high level decision-support tool for the design and solving different kinds of risk management and optimization problems (linear, quadratic, convex, nonconvex, mixed integer, etc.).

- Our team includes professionals in risk management and prominent researchers in optimization and numerical methods.

PSG Environments

PSG Operates in 3 Environments:

- Run-File
- MATLAB
- C++

- Run-File is a text based tool for solving optimization problems prepared as *.txt files in General PSG Format.

- MATLAB Environment includes a set of m-functions for generating and solving optimization problems in MATLAB. Also, it includes GUI Toolbox.

- C++ Environment contains classes for incorporating PSG codes to external programming systems.

PSG Functions

PSG contains an extensive set of pre-coded Risk, Stochastic, Deterministic and Scenario functions. The most popular groups of functions:

- VaR
- Probability
- CVaR
- CDaR
- Partial Moment
- Error
- Cardinality, etc.

Pre-coded PSG functions allows for a quick and efficient operating with them. User can define additional functions.

PSG Solvers

- Four main solvers for optimization problems (based on different optimization techniques):
  - VAN
  - CAR
  - TANK
  - BULDOZER
- Three additional solvers based on Gurobi optimization package:
  - VANGRB
  - CARGRB
  - HELI

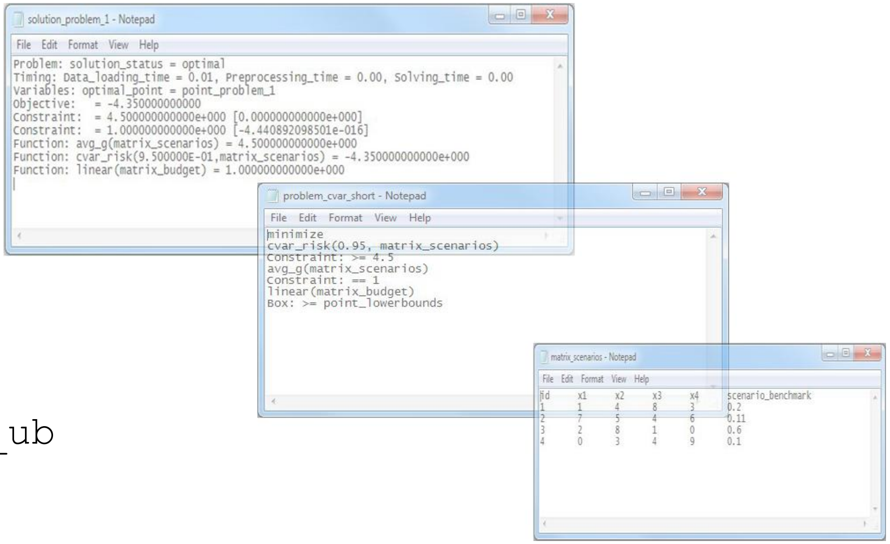
Problem Statement and Data

Optimization Problem in PSG includes two parts: **Problem Statement** and **Data**.

- **Problem Statement** is a text written in PSG Script (type of optimization, objective function, constraints, box of variables, solver): simple and clear format.

Example of Problem Statement:

```
maximize
 linear(matrix_r)
Constraint: <= 0.02
 var_risk(0.95, matrix_B)
Constraint: == 1
 linear(matrix_C)
Box: >= point_lb, <= point_ub
Solver: car
```

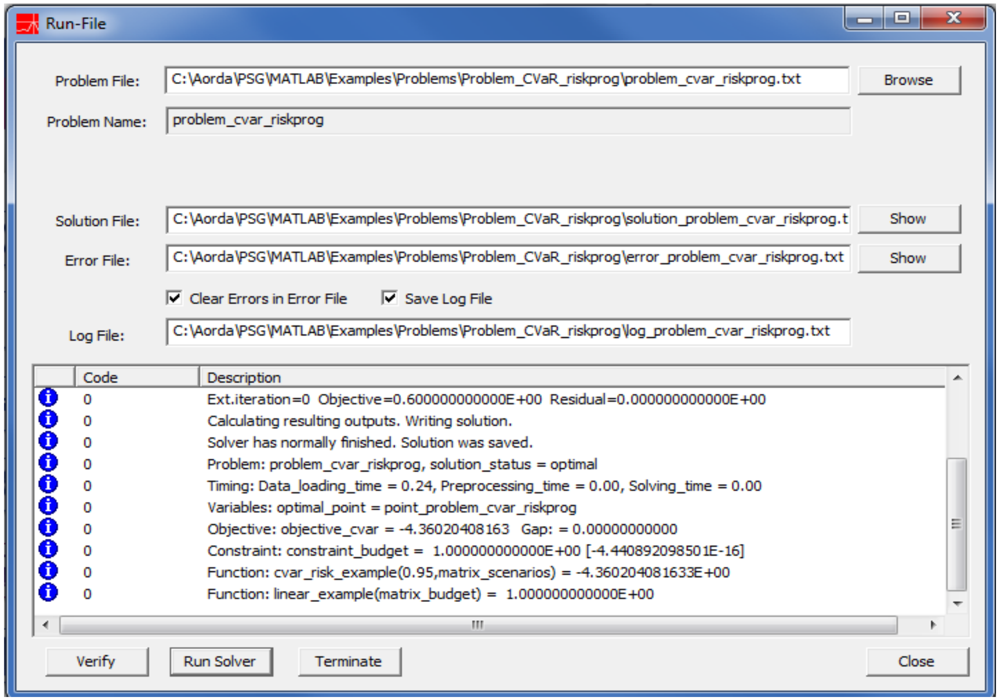


- Four types of **Data Objects: matrix**, **point**, **vector**, **parameter**.

  Run-File data are stored in separate *.txt files. In MATLAB data is stored in array of structures.

Run-File

Run-File is a PSG tool for solving optimization problem located in a set of files in General (Text) Format. Start Run-File.exe and chose a problem in the directory.



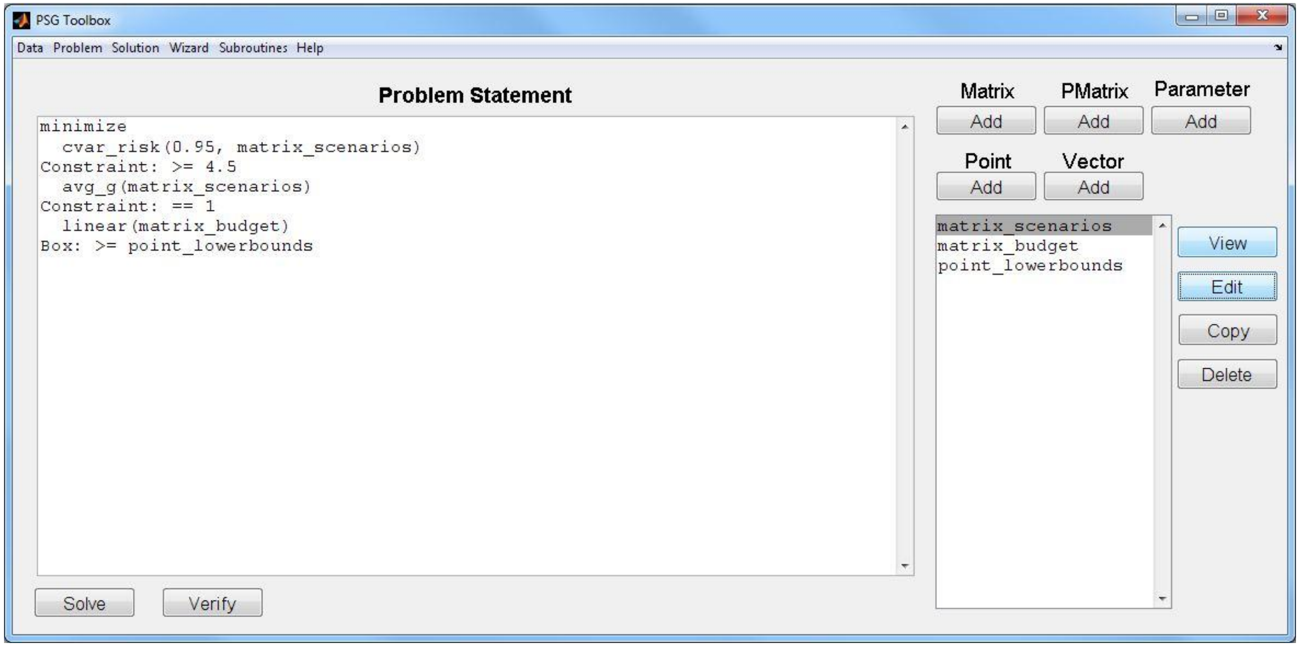
Main operations in Run-File: verify and solve problem, show files with solution and errors.

PSG MATLAB Toolbox

PSG MATLAB Toolbox is multi-window tool for creation and solving optimization problems in MATLAB Environment.

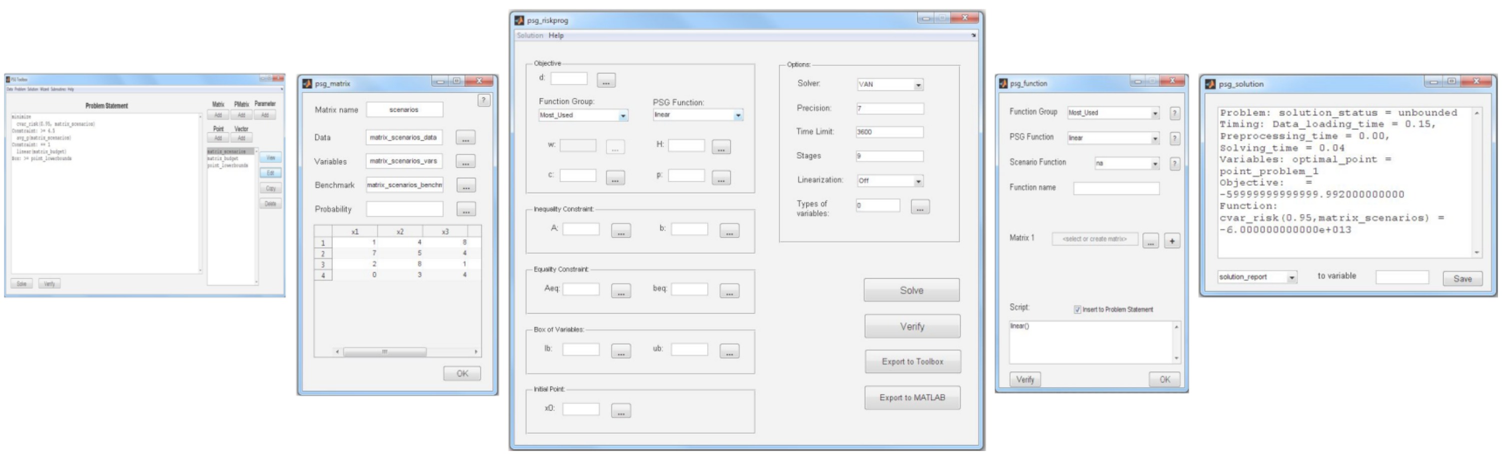
Toolbox window includes two parts:

- Left part for typing and editing problem statement
- Right part for working with PSG Data Objects:



Main operations: PSG MATLAB Toolbox

- Creating and Editing a problem with Wizard for choosing PSG functions
- Verifying and solving PSG problem
- Loading or Saving problem with solution in *.mat files
- Windows for Special PSG Subroutines riskprog and riskconstrprog
- Exporting and importing from text format
- Saving PSG problem to MATLAB workspace
- Creating user subroutines



MATLAB Special Subroutines

Special MATLAB subroutines for different optimization problems.

riskprog	risk minimization with linear and box constraints
riskconstrprog	risk minimization with constraint on risk function
riskratioprog	risk ratio maximization
riskparam, riskconstrparam	optimization for a set of parameters
functionvalue, fuctionsensitivity, functionincrements	calculation of values, sensitivities, and increments of PSG functions

MATLAB Special Subroutines

Examples of subroutines:

- `[xout, fval, status, output] = riskprog('cvar_risk', 0.95, H, c, [], [], [], []Aeq, beq, lb);`
- `[xout, fval, status, output] = riskconstrprog('cvar_risk', 'avg', 0.95, H, c, [], [], H, c, [], [], r, [], [], Aeq, beq, lb);`
- `[val] = functionvalue('cvar_risk', 0.65, H, c, p, a);`

Possibility to create user subroutines for optimization problems.

Mathematical Problem Statement

Finds a minimum for a problem specified by

$$\min_x \left[risk + d^T x \right]$$

subject to

$$A \cdot x \le b$$
$$Aeq \cdot x = beq$$
$$lb \le x \le ub$$

where
A, Aeq are matrices;
d, x are column vectors;
b, beq, lb, ub are column vectors or scalars;
risk function is a linear combination of PSG risk functions, PSG deterministic functions, or a PSG utility functions (see List of PSG functions for riskprog).

Syntax

```
xout = riskprog (risk, w, H, c, p, d, A, b)
xout = riskprog (risk, w, H, c, p, d, A, b, Aeq, beq)
xout = riskprog (risk, w, H, c, p, d, A, b, Aeq, beq, lb, ub)
xout = riskprog (risk, w, H, c, p, d, A, b, Aeq, beq, lb, ub, x0)
xout = riskprog (risk, w, H, c, p, d, A, b, Aeq, beq, lb, ub, x0, options)
[xout, fval] = riskprog (…)
[xout, fval, exitflag] = riskprog (…)
[xout, fval, exitflag, output] = riskprog (…)
```

```
%Define input arguments:
H=[1,4,8,3; 7,5,4,6; 2,8,1,0; 0,3,4,9];
c=[0.2; 0.11; 0.6; 0.1];
Aeq=[1, 1, 1, 1];
beq=1;
r=-4.5;
lb=[0; 0; 0; 0];

%Solve optimization problems:
[xout, fval, status, output] = riskconstrprog('cvar_risk', 'avg',...
    0.95, H, c, [], [], H, c, [], [], r, [], [], Aeq, beq, lb);

%Display results:
disp(' ');
disp('Results: ');
%Display status of optimization problem:
disp(sprintf('status of optimization problem = %s', status));
%Display solving time:
disp(sprintf('solving time = %g', output.solving_time));
%Display objective:
disp(sprintf('objective = %g', fval));
%Display function:
disp(sprintf('cvar_risk= %g', output.frval1));
disp(sprintf('avg= %g', output.frval2))
%Display optimal point:
disp('optimal point = ');
disp(xout');
```

Load data from mat file:

```
load('ThisData', 'Hmatr', 'dmatr', 'Aeqmatr', 'beqvec', 'lbvec');
```

Optimize problem:

```
[xout, fval, exitflag, output] = riskprog('cvar_risk', 0.95, Hmatr, [], [], [], dmatr', -0.00105,
Aeqmatr, beqvec, lbvec, [], '');
```

Describe input and output arguments of PSG subroutine `riskprog`:

Input arguments	Meaning
'cvar_risk'	Name of Risk function.
0.95	Confidence level in CVaR Risk.
Hmatr	Matrix of Scenarios (matrix_prior_scenarios).
dmatr'	Matrix for Linear function in constraint on expected return (constraint_return).
-0.00105	Lower bound of constraint return.
Aeqmatr	Matrix for linear function in the budget constraint (constraint_budget).
beqvec	Lower bound of budget constraint.
lbvec	Lower bound for variables.

Output argument	Meaning
xout	Optimal point.
fval	Optimal value of objective.
exitflag	Status of solution.
output	Output string.

MATLAB m-functions

PSG includes a set of m-functions for creating and solving optimization problems.

Main groups of functions:

- Operations with PSG Problems:

  tbpsg_run - solving problems, **tbpsg_vars** - showing all problem inputs, **tbpsg_create_user_subroutine** - creating user subroutine.

- Creation of PSG Data Objects:

  tbpsg_matrix_pack - pack PSG Matrix,
  tbpsg_point_pack - pack PSG Point, etc.

- Extract Solution Results:

  tbpsg_solution_struct - create structure with a solution report,
  tbpsg_objective - returns objective value, etc.

MATLAB m-functions

Example of problem generation and solving:

```
%Define input arguments:
H=[1,4,8,3; 7,5,4,6; 2,8,1,0; 0,3,4,9];
c=[0.2; 0.11; 0.6; 0.1];
Aeq=[1, 1, 1, 1];
lb=[0; 0; 0; 0];
vars={'x1', 'x2', 'x3', 'x4'};
alpha = 0.95;

%Create problem statement:
problem_statement = sprintf('%s\n',...
'minimize',...
`  cvar_risk(parameter_alpha, matrix_scenarios)',...
'Constraint: == 1',...
`  linear(matrix_budget)',...
'Box: >= point_lowerbounds');

%Pack PSG Data Objects:
toolboxstruc_arr(1) = tbpsg_matrix_pack('scenarios',H,vars,c);
toolboxstruc_arr(2) = tbpsg_matrix_pack('budget',Aeq,vars);
toolboxstruc_arr(3) = tbpsg_point_pack('lowerbounds',lb,vars);
toolboxstruc_arr(4) = tbpsg_parameter_pack('alpha',alpha);

%Optimize problem:
[solution_str, outargstruc_arr] = tbpsg_run(problem_statement,toolboxstruc_arr);

%Extract optimal point data:
point_data = tbpsg_optimal_point_data(solution_str, outargstruc_arr);
```

Case Studies

Examples of solved problems with codes and data are at www.aorda.com.

PSG can solve various problems in different application areas:

- **Financial Engineering** (Mortgage Pipeline Hedging, Portfolio Optimization with Mixed CVaR Profile, Portfolio Optimization with Nonlinear Transaction Costs, and others).
- **Advanced Statistics** (Estimation of CVaR through Explanatory Factors with Super Quantile Regression, Spline Regression, Support Vector Machines Based on Tail Risk Measures, Data Envelopment Analysis,and others).
- **Stochastic Programming** (Stochastic Two Stage Linear Problem, Supply Chain Planning Problem, and others).
- **Logistics** (Optimal Allocation of Stock Levels and Stochastic Customer Demands to a Capacitated Resource, Stochastic Multicommodity Network Flow Problem, and others).
- **Medical Applications** (Optimizing Intensity-Modulated Radiation Therapy Treatment Planning Problem).
- **Mechanics** (Optimization of Parameters of Beam Excitation Waveform).

Case Studies

High speed for large data sets.

Case Study	# of Variables	# of Scenarios	Solving Time (sec)
Hedging Portfolio of Options	121	45000	0.47
Spline Approximation	120	4371	3.82
Portfolio Optimization with Drawdown Constraints on Multiple Paths	31	12925	0.02
Relative Entropy Minimization	100,000	3	1.1
Support Vector Machines Based on Tail Risk Measures	1025	1,000	0.08
Structuring step up CDO	5	500,000	1.46
Classification of Loan Application s	20	380465	1.45
Optimal Position Liquidation with CVaR Constraints	10,050	2,000	6.10

Forthcoming Developments

New features:

- R Environment (solving problems with PSG in R)
- Windows based Shell for a quick and simple operation with optimization problems
- New classes of PSG functions
- Composition of PSG functions
- MIP for general convex and nonconvex problems.
- Second Order Conic programming

Thank You!

Download and Try PSG:

Request a Trial

Learn more:

PSG Help

Case Studies

Contact us:

support@aorda.com

Request a Call