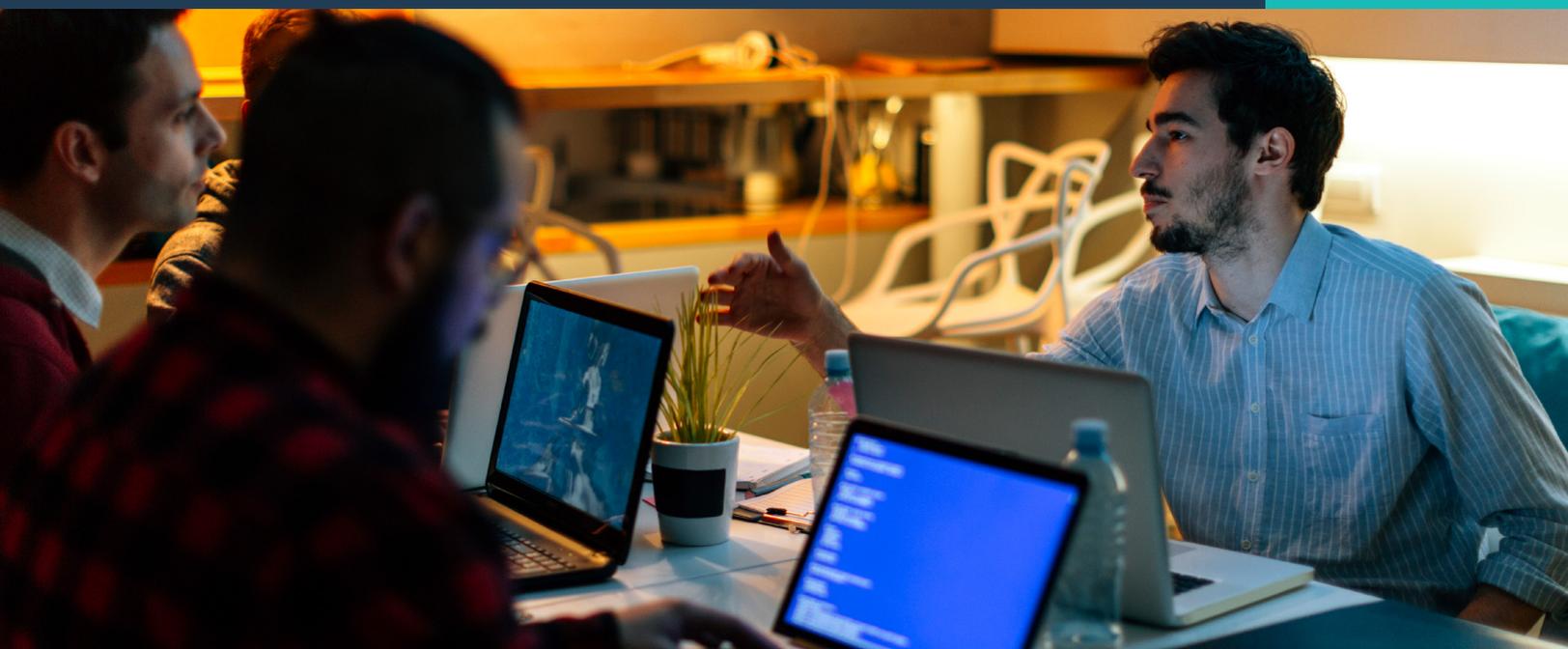


# SAS® Optimization on SAS® Viya™

Find optimal solutions to complex business and planning problems faster than ever



## Key Benefits

- **Quickly solve complex optimization problems.** This solution takes advantage of SAS® Viya™, a new distributed, in-memory platform, to deliver optimization modeling results at break-through speeds. So you can find optimal solutions to difficult problems faster.
- **Empower users with language options.** Python, Java, R and Lua programmers can experience the power of this product without having to learn SAS. Now they can access trusted and tested SAS Optimization algorithms from other languages.
- **Drive better decision making.** SAS Optimization offers state-of-the-art methods for mathematical optimization that are integrated with a full suite of data preparation, exploration, analytics and reporting capabilities. This unified environment enables organizations to identify and apply the best responses to complex, real-world problems.

## Overview

SAS Optimization provides a powerful array of optimization modeling capabilities and solution techniques that enable organizations to effectively consider more alternative scenarios to determine the best course of action. With SAS Optimization, you can:

- Identify actions that will produce the best results, while operating within resource and other relevant constraints.
- Determine the optimal allocation of resources, and select the best way to achieve organizational goals.
- Add structure, consistency, adaptability and repeatability to decision-making processes.
- Avoid the hassles of dealing with niche software packages.

```

1 libname mycaslib cas caslib=casuser;
2
3 proc optmodel;
4 set ASSETS;
5 num return {ASSETS};
6 num cov {ASSETS, ASSETS} init 0;
7 read data mycaslib.pf_means into ASSETS=[_n_] return;
8 read data mycaslib.pf_cov into [asset1 asset2] cov [asset2,asset1]=cov;
9 num risklimit init 0.00025;
10 num minThreshold init 0.1;
11 num numTrials = 20;
12
13 /* declare MIP problem for fixed set of assets */
14 set ASSETS_THIS;
15 var AssetPropVar {ASSETS_THIS} >= minThreshold <= 1;
16 max ExpectedReturn =
17 sum {i in ASSETS_THIS} return[i] * AssetPropVar[i];
18 con RiskBound:
19 sum {i in ASSETS_THIS, j in ASSETS_THIS} cov[i,j] * AssetPropVar[i] * AssetPropVar[j] <= riskLimit;
20 con TotalPortfolio:
21 sum {asset in ASSETS_THIS} AssetPropVar[asset] = 1;
22
23 num infinity = constant('BIG');
24 num best_objective init -infinity;
25 set INCUMBENT;
26 num sol {INCUMBENT};
27
28 num overall_start;
29 overall_start = time();
30 set TRIALS = 1..numTrials;
31 num start {TRIALS};
32 num finish {TRIALS};
33 call streaminit(4);
34 cofor {trial in TRIALS} do;

```

Figure 1: The SAS Studio programming environment in SAS Optimization.

## Capabilities

### Algebraic, symbolic optimization modeling

A powerful, intuitive algebraic optimization modeling language with specialized syntax and constructs enables you to build optimization models with clarity and precision. This makes it easier to review models for initial validation, make adjustments and run models with new data.

### A single language supports a wide range of optimization models and constraint programming

With SAS Optimization, you need to learn only one set of statements and commands to build and solve a wide range of optimization models, including linear, network, mixed integer linear, nonlinear and quadratic, as well as constraint satisfaction problems. As analysts adjust formulations to address evolving requirements, the constraints and/or objectives can change from linear to nonlinear expressions and vice versa. With this common language, you'll have no problem implementing the most appropriate algorithm.

### Powerful optimization solvers and presolvers

SAS Optimization provides a suite of optimization solvers that is streamlined for simplicity and tuned for performance. Aggressive presolvers reduce effective problem size. This enables you to tackle large problems and solve them more quickly.

- Linear programming solvers include primal and dual simplex; network simplex; and interior point with crossover.
- A parallel branch-and-bound, mixed integer programming solver has cutting planes, heuristics, conflict search and option tuning. A concurrent mode is also available.
- General nonlinear optimization solvers include active set and interior point; concurrent solve; and the multistart algorithm.

- A state-of-the-art quadratic solver is tailored for large-scale optimization problems.
- A constraint programming solver enables you to address constraint satisfaction problems using domain reduction/constraint propagation and a choice of search strategies (look-ahead and backtracking).

## Network flow optimization

Network algorithms, accessible from both PROC OPTMODEL and PROC OPTNETWORK, enable you to investigate the characteristics of networks and find the best answers to network-oriented problems. Optimization and diagnostic algorithms include shortest path, maximum flow, minimum-cost flow, traveling salesman problem, connected and biconnected components, clique and cycle enumeration, minimum spanning tree, linear assignment and more.

## Multistart algorithm helps identify better solutions to nonlinear problems

The multistart algorithm for nonconvex nonlinear optimization increases your chance of finding a globally optimal solution from among many locally optimal solutions. This iterative algorithm selects multiple starting points and begins optimization in parallel from each. The best solution from all starting points is reported.

## Decomposition algorithm

The decomposition algorithm (automated Dantzig-Wolfe) works well for large, appropriately structured linear and mixed integer linear optimization problems. It decomposes the overall problem into a set of component problems each with an exclusive set of decision variables. Parallel solution of the subproblems is coordinated with the overall solution process, reducing time to solution significantly.

## Open, cloud-enabled, in-memory platform

SAS Optimization runs on SAS Viya, an open, scalable, distributed in-memory processing platform. It provides a fast, multiuser environment for concurrent access to data in memory. The platform is enhanced with fault tolerance for high availability. SAS Viya also provides an open analytics coding environment. Whether it's SAS, Python, Java, R or Lua, analytical professionals can access the power of SAS using their language of choice. And with public REST APIs, you can add the power of SAS Analytics to other applications. All analytical assets are managed within a common environment to provide a single, governed model inventory across applications.

## Learn More

Find out more at [sas.com/optimization](https://sas.com/optimization).

To contact your local SAS office, please visit: [sas.com/offices](https://sas.com/offices)

