

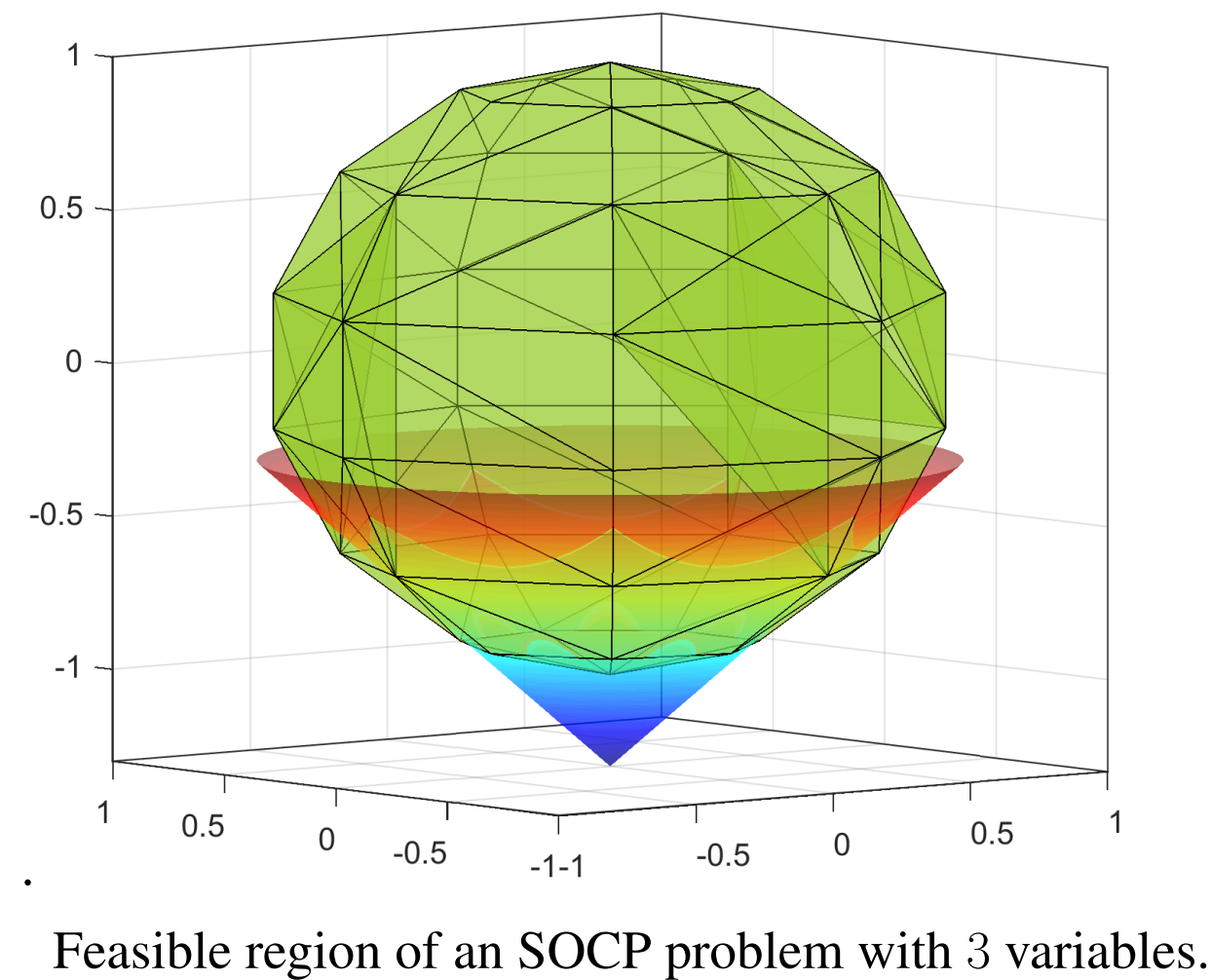
What is SOCP?

Second-order cone programming (SOCP) is convex optimization which extends linear programming (LP) with second-order (Lorentz or the ice cream) cones. It appears in a **broad range of applications** from **engineering**, control theory and **quantitative finance** to quadratic programming and **robust optimization**. It has become an important tool for financial optimization due to its **powerful nature**. Interior point methods (IPM) are the most popular approaches to solve SOCP problems due to their theoretical polynomial complexity and **practical performance**. The NAG SOCP solver (e04pt) uses an IPM to solve a problem in the standard form:

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{subject to } l_A \leq Ax \leq u_A, \\ & \quad l_x \leq x \leq u_x, \\ & \quad x \in \mathcal{K}, \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $l_A, u_A \in \mathbb{R}^m$, $c, l_x, u_x \in \mathbb{R}^n$ are the problem data, and $\mathcal{K} = \mathcal{K}^{n_1} \times \dots \times \mathcal{K}^{n_r} \times \mathbb{R}^{n_l}$ where \mathcal{K}^{n_i} is a second-order cone defined as

$$\mathcal{K}_q^{n_i} := \left\{ x = (x_1, \dots, x_{n_i}) \in \mathbb{R}^{n_i} : x_1^2 \geq \sum_{j=2}^{n_i} x_j^2, x_1 \geq 0 \right\}.$$



Versatility of SOCP and SOCP-representable problems

SOCP is widely used in quantitative finance due to its **flexibility and versatility** to handle a large variety of problems with different kinds of constraints, such as stochastic and robust portfolio optimization. Here is a list of problems and constraints that can be **transformed into equivalent SOCPs**. Quantitative finance professionals could use these elements to build more complex and more realistic models other than linear and quadratic programming.

- Convex quadratic constraint:

$$\frac{1}{2}x^T Px + q^T x + r \leq 0.$$

- The p -norm constraints:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \leq t,$$

where $p \geq 1$, e.g., absolute value $|x| \leq t$, l_1 -norm $\|x\|_1 \leq t$ and Euclidean norm $\|x\|_2 \leq t$.

- Minimize sum of norm: $\min \sum_{i=1}^n \|x_i\|_2$.

- Minimize maximum of norm: $\min \max_{1 \leq i \leq r} \|x_i\|_2$.

- Probability constraint:

$$\text{Prob}(a^T x \leq b) \geq \eta,$$

where a is an independent Gaussian random vector and $\eta \geq 0.5$.

- Constraints involving power functions, for example:

$$x_1^5 x_2^2 x_3^3 x_4^6 \geq 1 \text{ and } -x_1^{1/5} x_2^{2/5} x_3^{1/5} \leq x_4, x_i \geq 0.$$

- Quadratic/linear fractional problem:

$$\begin{aligned} & \text{minimize } \sum_{i=1}^p \frac{\|F_i x + g_i\|^2}{a_i^T x + b_i}, \\ & \text{subject to } a_i^T x + b_i > 0, i = 1, \dots, p. \end{aligned}$$

Performance of the NAG SOCP solver

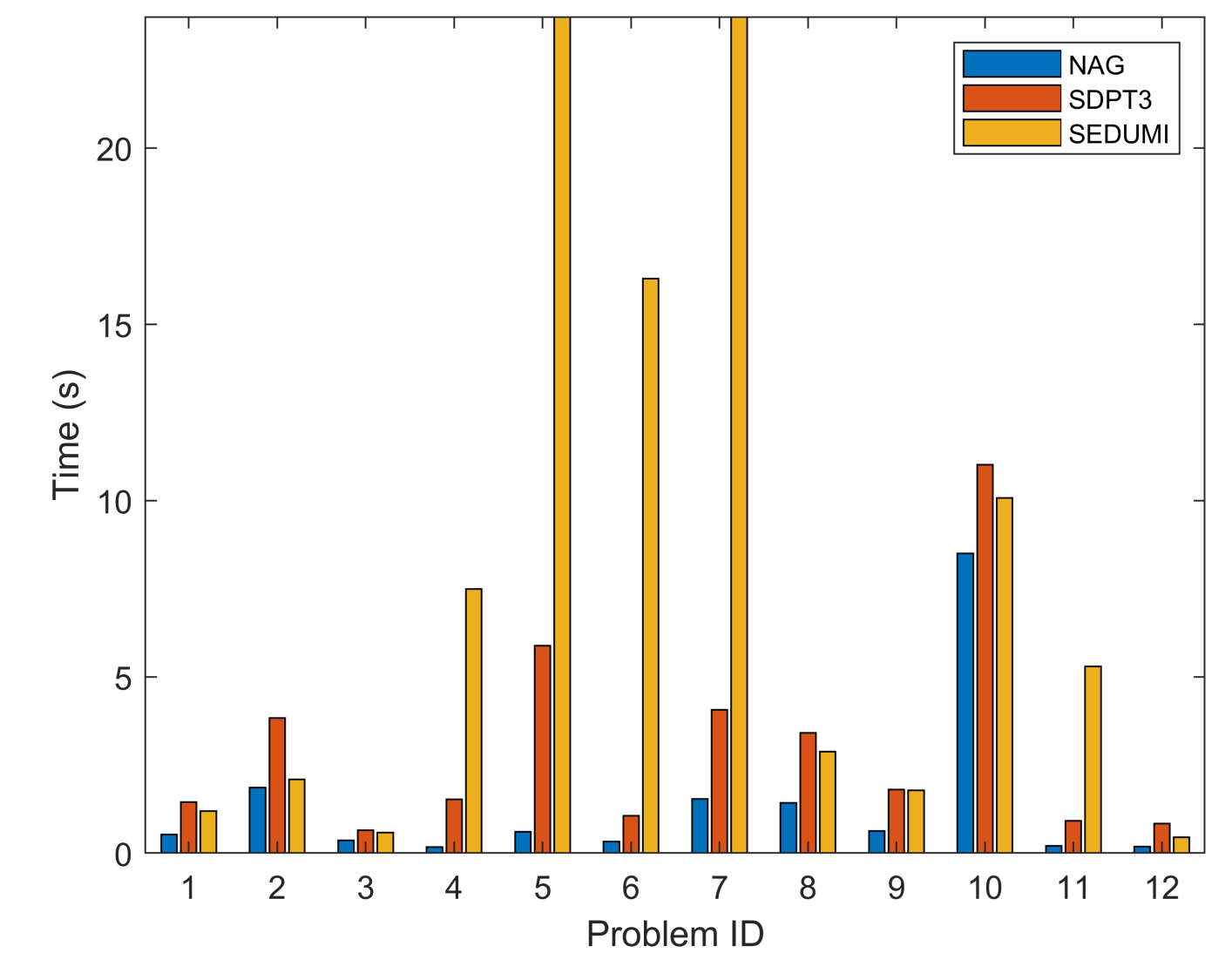
We compare our solve times for 18 DIMACS Challenge problems to the well-known solvers SEDUMI and SDPT3 (default options, accuracy 10^{-8} , single-threaded mode).

	NAG	SEDUMI	SDPT3
Problems solved	100%	77.78%	83.33%

Table 1: Statistics on solvers' status after run, percentage of successful solve attempts.

Prob. class	No. of prob.	Avg. n	Avg. m	Avg. nc
nb	4	3098.75	321	906
nql	3	86102	49440	12300
qssp	3	99486	49471	24871
sched	8	17712.5	8448.5	1.5

Table 2: DIMACS problem statistics. 18 test problems in 4 classes. (n number of variables, m number of linear constraints, nc number of quadratic cone constraints.)



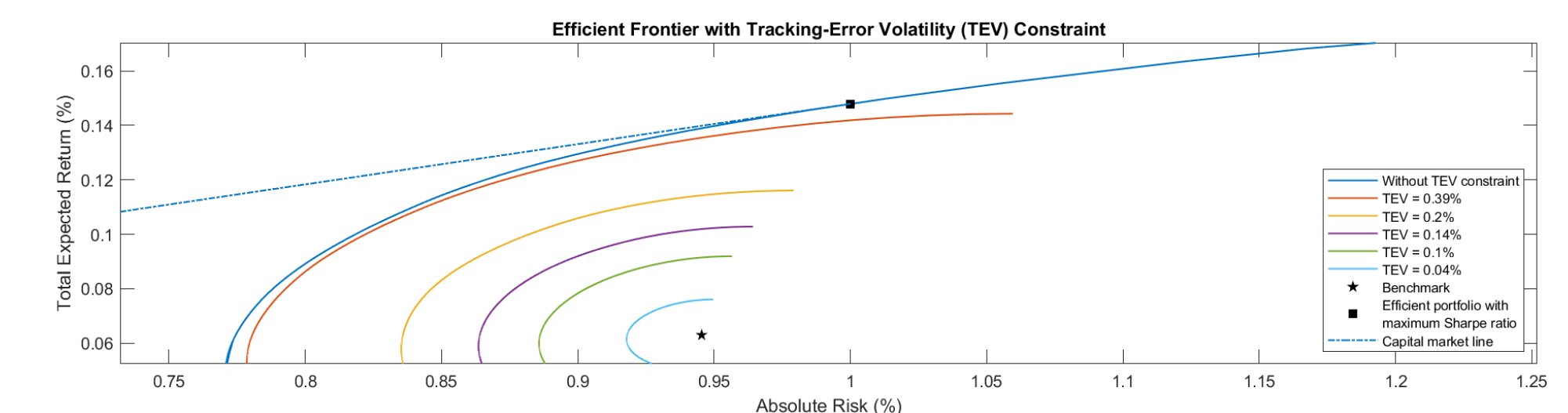
Comparison of time on 12 DIMACS problems that all three solvers managed to solve successfully.

The NAG SOCP solver is more robust and outperforms both SEDUMI and SDPT3 on all test cases in DIMACS problem dataset in terms of efficiency and accuracy.

Case study: Portfolio optimization with tracking-error constraints

The following model explores the risk and return relationship of **active portfolios** subject to **tracking-error volatility (TEV)**. Let r be the vector of expected returns and V the covariance matrix for asset returns (estimated from daily data for 30 stocks in DJIA from March 2018 to March 2019). Randomly generating a benchmark portfolio b , we solve the following optimization:

$$\begin{aligned} & \text{minimize } -r^T x + \mu(b+x)^T V(b+x) \\ & \text{subject to } e^T x = 0, \\ & \quad x + b \geq 0, \\ & \quad x^T V x \leq tev, \end{aligned}$$



where e is the vector of all ones, μ controls the trade-off between excess return and absolute risk and tev is the threshold on TEV. Note that without the absolute risk in the objective, the problem reduces to excess-return optimization. However, *Roll (1992)* noted this *classical model leads to the unpalatable result that the active portfolio has systematically higher risk than the benchmark* and is not optimal. Therefore, by taking the absolute risk into account the **QCQP model (solved by SOCP) improves the performance of active portfolio**.

SOCP solver in the NAG Library was used to solve the above model. Each efficient frontier in the figure was generated by solving 2000 SOCPs. The whole process took around 4 mins (less than 0.02s per solving).

Various financial models can be significantly enhanced by using the NAG SOCP solver.