Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2018, 6, 1111-1119
http://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Solution of Stochastic Quadratic Programming with Imperfect Probability Distribution Using Nelder-Mead Simplex Method

Xinshun Ma*, Xin Liu

Department of Mathematics and Physics, North China Electric Power University, Baoding, China
Email: *maxs@sina.com

How to cite this paper: Ma, X.S. and Liu, X. (2018) Solution of Stochastic Quadratic Programming with Imperfect Probability Distribution Using Nelder-Mead Simplex Method. *Journal of Applied Mathematics and Physics*, **6**, 1111-1119.
https://doi.org/10.4236/jamp.2018.65095

Received: April 16, 2018
Accepted: May 28, 2018
Published: May 31, 2018

Abstract

Stochastic quadratic programming with recourse is one of the most important topics in the field of optimization. It is usually assumed that the probability distribution of random variables has complete information, but only part of the information can be obtained in practical situation. In this paper, we propose a stochastic quadratic programming with imperfect probability distribution based on the linear partial information (LPI) theory. A direct optimizing algorithm based on Nelder-Mead simplex method is proposed for solving the problem. Finally, a numerical example is given to demonstrate the efficiency of the algorithm.

Keywords

Stochastic Quadratic Programming, LPI, Nelder-Mead

1. Introduction

Stochastic programming is an important method to solve decision problems in random environment. It was proposed by Dantzig, an American economist in 1956 [1]. Currently, the main method to solve the stochastic programming is to transform the stochastic programming into its own deterministic equivalence class and using the existing deterministic planning method to solve it. According to different research problems, stochastic programming mainly consists of three problems: distribution problem, expected value problem, and probabilistic constraint programming problem. Classic stochastic programming with recourse is a type of expected value problem, modeling based on a two-stage decision-making method. It is a method by making decisions before and after observing the value of a variable. With regard to the theory and methods of two-stage stochastic programming, a very systematic study has been conducted and many important solutions have been proposed [2]. In these methods, the dual decomposition L-shape algorithm established in the literature [3] is the most effective algorithm for solving two-stage stochastic programming. It is based on the duality theory, and the algorithm converges to the optimal solution by determining the feasible cutting plane and optimal cutting, and solving the main problem step by step. This method is essentially an external approximation algorithm that can effectively solve the large-scale problems that occur after the stochastic programming is transformed into deterministic mathematical programming. Abaffy and Allevi present a modified version of the L-shaped method in [4], used to solve two-stage stochastic linear programs with fixed recourse. This method can apply class attributes and special structures to a polyhedron process to solve a certain type of large-scale problems, which greatly reduces the number of arithmetic operations.

While the stochastic programming is transformed into the corresponding equivalence classes, it is generally a nonlinear equation. In recent years, with the introduction of new theories and methods for solving nonlinear equations, especially the infinite dimensional variational inequality theories and the application of smoothness techniques that have received widespread attention in recent years [5] [6] [7], a stochastic programming solution method based on nonlinear equation theory is proposed. Chen X. expressed the two-stage stochastic programming as a deterministic equivalence problem in the literature [8], and transformed it into a nonlinear equation problem by introduced Lagrange multiplier. By using the B-differentiable properties of nonlinear functions, a Newton method for solving stochastic programming is proposed. Under certain conditions, the global convergence and local super-linear convergence of the algorithm are proved.

In general, stochastic programming is based on the complete information about probability distribution, but in practical situation, due to the lack of historical data and statistical theory, it is impossible to obtain complete information of the probability distribution, and can only get partial information in fact. In order to solve this problem, literature [9] and [10] based on fuzzy theory, under the condition that the membership function of certain parameters of the probability distribution is known, the method of determining the two-stage recourse function is given , two-stage and multi-stage stochastic programming problems are distributed and discussed. In [11], based on the linear partial information (LPI) theory of Kofler [12], a class of two-stage stochastic programming with recourse is established, and an L-shape method based on quadratic programming is given. Based on the literature [8] and literature [11], this paper establishes a two-stage stochastic programming model under incomplete probability distribution information based on LPI theory, and presents an improved Nelder-Mead solution method. Experiment shows the algorithm is effective.

2. The Model of Stochastic Quadratic Programming with Imperfect Probability Distribution

Let (Ω,Σ,P) be a probability space, $h=h(\omega)\in R^m$ is a stochastic vector in this space, $C\in R^{n\times n}$ be symmetric positive semi-definite and $H\in R^{m\times m}$ be symmetric positive definite. We consider following problem:

$$\begin{aligned} \min_{x\in R^n,y\in R^m} \quad & \frac{1}{2}x^{\mathrm{T}}Cx+c^{\mathrm{T}}x+f(y) \\ \text{s.t.} \quad & Ax\le b \\ & Tx=y \end{aligned} \tag{1}$$

where

$$f(y)=\max_{P\in\pi} E_P\left(g(y,\omega)\right) \tag{2}$$

$$\begin{aligned} g(y,\omega_i)=\max_{z\in R^m} \quad & -\frac{1}{2}z^{\mathrm{T}}Hz+z^{\mathrm{T}}\left(h(\omega_i)-y\right) \\ \text{s.t.} \quad & Wz\le q \end{aligned} \tag{3}$$

Here $x\in R^n, y\in R^m$ are decision variables, $c\in R^n, A\in R^{r\times n}, b\in R^r, T\in R^{m\times n}$, $q\in R^{m_1}, W\in R^{m_1\times m}$ are fixed matrices, $\omega\in R^{m_2}$ is a random vector with support $\Omega\subseteq R^{m_2}$, $P=(p_1,p_2,\cdots,p_N)^{\mathrm{T}}$ is the probability distributions of limited sample set $\Omega=(\omega_1,\cdots,\omega_N)$, that is $p_i=prob(\{\omega=\omega_i\})$. Assumed that the probability distribution of random variable has the following linear partial information:

$$\pi=\left\{P\in R^N \mid BP\le d,\sum_{i=1}^{N}p_i=1; p_i\ge 0, i=1,\cdots,N\right\}$$

Here $d\in R^s, B\in R^{s\times N}$ are fixed matrices, assumed the set of probability distributions π is a polyhedron. Thus the two-stage function can be written as

$$f(y)=\max_{P\in\pi} E_P\left(g(y,\omega)\right)=\max_{P\in\pi}\sum_{i=1}^{N}p_i g(y,\omega_i) \tag{4}$$

We call Equations (1)-(3) stochastic quadratic programming with recourse models under LPI.

Chen established a similar stochastic quadratic programming model in [8], but assumed that the probability distribution is completely known, that is the "Max" symbol in Equation (2) does not appear. The above model is a new stochastic quadratic programming model based on LPI theory to solve the stochastic programming problem with incomplete information probability distribution. Since $g(y,\omega_i)$ is the convex function about y ([8]), $f(y)$ is also the convex function about y (see [13]), and then the problems (1)-(3) essentially belong to the convex programming problem. Obviously the recourse function is not differential, so the Newton method proposed in [8] is no longer applicable. In order to solve this problem, we design a solution based on the improved Nelder-Mead method. The experimental results show that the method is effective.

3. Modified Nelder-Mead

The Nelder-Mead method (NM) [14] was originally a direct optimization algorithm developed for solving the nonlinear programming, NM algorithm belongs to the modified polyhedron method in nature. It searches for the new solution by reflecting the extreme point with the worst function value through the centroid of the remaining extreme points. Experimental shows, compared to random search, the algorithm can find the optimal solution more efficiently. The NM algorithm does not require any gradient information of the function during the entire optimization procedures, it can handle problems for which the gradient does not exist everywhere. NM allows the simplex to rescale or change its shape based on the local behavior of the response function. When the newly-generated point has good quality, an extension step will be taken in the hope that a better solution can be found. On the other hand, when the newly-generated solution is of poor quality, a contraction step will be taken, restricting the search on a smaller region. Since NM determines its search direction only by comparing the function values, it is insensitive to small inaccuracies in function values.

The classic NM method has several disadvantages in the search process. First, the convergence speed of the algorithm depends too much on the choice of initial polyhedron. Indeed, a too small initial simplex can lead to a local search, consequently the NM may convergent to a local solution. Second, NM might perform the shrink step frequently and in turn reduce the size of simplex to the greatest extent. Consequently, the algorithm can converge prematurely at a non-optimal solution.

Chang [15] propose a new variant of Nelder-Mead, called Stochastic Nelder-Mead simplex method (SNM). This method seeks the optimal solution by gradually increasing the sample size during the iterative process of the algorithm, which not only can effectively save the calculation time, but also can increase the adaptability of the algorithm to prevent premature convergence of the algorithm. This article refers to the design idea of [15] and adds an adaptive random search process to solve problems (1)-(3) in the NM algorithm. The specific process of the algorithm is described as follows:

Firstly, by attaching a Lagrange multiplier vector λ, convex problems (1)-(3) can be written as an unconstrained problem:

$$\theta(\mu) = \min \frac{1}{2} x^{\mathrm{T}} P x + c^{\mathrm{T}} x + f(y) + \lambda^{\mathrm{T}}(Ax - b) \tag{5}$$

$\mu = (x, \lambda)^{\mathrm{T}}$, let $\mu^1, \mu^2, \cdots, \mu^{n+1}$ be the $n+1$ points of n-dimensional space of R^n, which are not on the same plane. Let μ^h, μ^s, μ^l represent the points that have the highest, second highest, and lowest estimates of function values, μ^c is the centroid of all vertices other than μ^h, $\mu^c = \frac{1}{n} \sum_{i \neq h} \mu^i$.

Reflection: since μ^h is the vertex with the higher value among the vertices, we can expect to find a lower value at the reflection of μ^h in the opposite face formed by all vertices μ^i except μ^h. Generate a new point μ^r, by reflecting μ^h through μ^c according to

$$\mu^r = \mu^c + \alpha(\mu^c - \mu^h) \text{ with } (\alpha > 0).$$

If the reflected point is better than the second worst, but not better than the best, *i.e.* $\theta\left(\mu^l\right) < \theta\left(\mu^r\right) < \theta\left(\mu^s\right)$, then obtain a new simplex by replacing the worst point μ^h with the reflected point μ^r.

Expansion: if the reflected point is the best point so far, we can expect to find interesting values along the direction from μ^c to μ^r, that is if $\theta\left(\mu^r\right) < \theta\left(\mu^l\right)$, then search direction is favorable, compute the expansion point using

$$\mu^e = \mu^c + \beta\left(\mu^c - \mu^h\right) \text{ with } (\beta > 1).$$

If the expanded point is better than the reflected point, that is $\theta\left(\mu^e\right) < \theta\left(\mu^r\right)$, then replace μ^h by μ^e, otherwise, obtain a new simplex by replacing the worst point μ^h with the reflected point μ^r.

Contraction: here it is certain that $\theta\left(\mu^r\right) > \theta\left(\mu^s\right)$, in this case, we can expect that a better value will be inside the simplex formed by all the vertices, then the simplex contracts.

1) If $\theta\left(\mu^s\right) < \theta\left(\mu^r\right) < \theta\left(\mu^h\right)$, the contracted point is determined by $\mu^p = \mu^c + \gamma\left(\mu^r - \mu^c\right)$ with $0 \le \gamma < 1$, if $\theta\left(\mu^p\right) < \theta\left(\mu^h\right)$, the contraction is accepted, Replaced μ^h by μ^p.

2) If $\theta\left(\mu^r\right) \ge \theta\left(\mu^h\right)$, the contracted point is determined by $\mu^p = \mu^c + \gamma\left(\mu^h - \mu^c\right)$, if $\theta\left(\mu^p\right) < \theta\left(\mu^h\right)$, the contraction is accepted. Replaced μ^h by μ^p.

Shrink: although a failed contraction is much rarer, it may happen in some case, In that case, generally we contract towards the lowest point in the expectation of finding a simpler landscape. Replace all points except the best point μ^l with $\mu^i = \mu^l + \delta\left(\mu^i - \mu^l\right)$. This article uses the following process: when contraction fails, using a random search process to generate new points based on fitness of function. Let fitness function be $F\left(\mu^i\right) = -\theta\left(\mu^i\right) + M$, where M is a fully large number, calculate the probability of obtaining μ^i by $F\left(\mu^i\right) \big/ \sum_{i=1}^{n+1} F\left(\mu^i\right)$. According to the roulette algorithm, get a new point by randomly searched in the neighborhood of the point corresponding to the probability interval. The neighborhood $\delta\left(\mu^i\right)$ of μ^i is defined as

$$\delta\left(\mu^i\right) = \left\{\mu : \left\|\mu - \mu^i\right\| \le \min\left\{\left\|\mu^i - \mu^j\right\|, \forall j \ne i\right\}\right\}$$

Algorithm termination condition: There are different criteria to determine the termination conditions of the NM algorithm in practice, in this paper, we use

$$\left|\frac{\theta\left(\mu^h\right) - \theta\left(\mu^l\right)}{\theta\left(\mu^l\right)}\right| \le \varepsilon,$$

as our convergence criterion.

Parameters choice: The polyhedron transform in the NM algorithm mainly includes four parameters, α for reflection, β for expansion and γ for contraction, assumed they satisfy the following constraints:

$$\alpha > 0, \beta > 1, \beta > \alpha, 0 < \gamma < 1.$$

Algorithm

Choose $\alpha > 0, \beta > 1, \beta > \alpha, 0 < \gamma < 1$, Convergence criterion $\varepsilon > 0$.

Step 1 calculate the function values of n+1 points, rank all points and identify $\mu_k^h, \mu_k^s, \mu_k^l$, find μ_k^c, the centroid of all vertices other than μ_k^h, generating a new point μ_k^r by reflecting point μ_k^h through μ_k^c according to reflection rule $\mu_k^r = \mu_k^c + \alpha\left(\mu_k^c - \mu_k^h\right)$;

Step 2 if $\theta\left(\mu_k^r\right) < \theta\left(\mu_k^l\right)$, then the reflection point is expanded using the expansion rule $\mu_k^e = \mu_k^c + \beta\left(\mu_k^c - \mu_k^h\right)$, if $\theta\left(\mu_k^e\right) < \theta\left(\mu_k^r\right)$, then replace μ_k^h by μ_k^e, otherwise, let μ_k^r replaced μ_k^h, return to Step 4;

Step 3 if $\theta\left(\mu_k^l\right) < \theta\left(\mu_k^r\right) < \theta\left(\mu_k^s\right)$, then let μ_k^r replaced μ_k^h, go to Step 4, otherwise return to Step 5;

Step 4 if the convergence criterion is met, stop the iteration, otherwise, return to Step 1;

Step 5 if $\theta\left(\mu_k^r\right) \geq \theta\left(\mu_k^s\right)$, then the simplex contracts.

(i) if $\theta\left(\mu_k^s\right) < \theta\left(\mu_k^r\right) < \theta\left(\mu_k^h\right)$, the contraction point is determined by calculate $\mu_k^p = \mu_k^c + \gamma\left(\mu_k^r - \mu_k^c\right)$,

(ii) if $\theta\left(\mu_k^r\right) \geq \theta\left(\mu_k^h\right)$, the contraction point is determined by calculate $\mu_k^p = \mu_k^c + \gamma\left(\mu_k^h - \mu_k^c\right)$.

In these case, if $\theta\left(\mu_k^p\right) < \theta\left(\mu_k^r\right)$, the contraction accepted, If contraction is accepted, let μ_k^p replaced μ_k^h, return to Step 4;

Step 6 when all previous Step s fail, we use adaptive random search to generate new points, then return to Step 1.

4. Numerical Experiment

Consider the problem (1)-(3) in which $X \in R^3, H = I \in R^{3\times 3}, T = -H$, and

$$C = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, W = \begin{pmatrix} 1 & -2 & 3 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{pmatrix}, q = \begin{pmatrix} 6 \\ 3 \\ 4 \end{pmatrix}$$

$$c = \begin{pmatrix} -4 \\ -3 \\ -2 \end{pmatrix}, A = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}, b = \begin{pmatrix} 12 \\ 8 \end{pmatrix}$$

Assumed stochastic vector $\omega_i = \left(\omega_i^1, \omega_i^2, \omega_i^3\right)^{\mathrm{T}}$ $(i = 1, 2, \cdots, N)$ is three-dimensional vector, ω_i^1, ω_i^2 and ω_i^3 are independent of each other. We use MATLAB to randomly generate one hundred values for each other. Let $h\left(\omega_i\right) = \omega_i$, we experiment with the effectiveness of the algorithm under the following conditions.

Case 1: let $N = 10$, the parameter in the linear part information of the probability distribution π is set to

$$D_8 = (1, \cdots, 1)^{\mathrm{T}}, B = \left(E_8, \mathrm{O}_{8\times 1}, D_8\right),$$
$$d = (1/6, 1/8, 1/5, 1/4, 1/7, 1/6, 1/7, 2/7)^{\mathrm{T}}$$

This means the incomplete information probability distribution is satisfied

$$p_1+p_{10}\le 1/6, p_2+p_{10}\le 1/8,\cdots,p_1+p_2+\cdots+p_{10}=1$$

The parameters of modified NM method are given, reflection factor $\alpha=1$, expansion factor $\beta=2$, contraction factor $\gamma=0.5$, convergence criterion $\varepsilon=1\times10^{-8}$.

Use MATLAB R2008a to achieve the above problems, Table 1 gives the decision variable x and the function value θ, the calculation result retains four decimal places.

The actual running results show that the algorithm terminates after 61 times, the optimal solution is $x=(0.8634,0.4194,0.2721)^{\mathrm{T}}$, the optimal function value is $\theta=4.4630$. The running time is 28.515197 seconds.

Case 2: To verify the effectiveness of the algorithm, we expand the value of N, let $N=100$, the parameter in the linear part information of the probability distribution π as follows, $D_8=(1,\cdots,1)^{\mathrm{T}}$, $B=(E_8,O_{8\times91},D_8)$, $d=(1/6,1/8,1/5,1/4,1/7,1/6,1/7,2/7)^{\mathrm{T}}$, that is keep the row of B unchanged, extend the number of random variables to 100, this means

$p_1+p_{100}\le 1/6, p_2+p_{100}\le 1/8,\cdots,p_1+p_2+\cdots+p_{100}=1$, use MATLAB R2008a to achieve the above problems again, the result is Table 2.

From Table 2, we can see the program stops at 89 times, the optimal solution is $x=(0.8659,0.3530,0.2682)^{\mathrm{T}}$, optimal function value is $\theta=4.8989$. The running time is 382.307942 seconds. Comparing the results, we find that when the value of N is increased by 10 times, the running time increased by 13 times, this is normal, it need more times to calculate the recourse function. However, the number of iterations of the algorithm is only increased by 1/4, which shows that the algorithm in this paper is effective for solving stochastic programming problems.

Case 3: In order to investigate the sensitivity of the linear partial information constraint condition of the probability distribution to the algorithm, we increase

Table 1. Iterative results.

Iteration	x1	x2	x3	θ
1	1.0000	0.0000	0.0000	4.7366
2	1.0000	0.0000	0.0000	4.7366
3	1.0000	0.0000	0.0000	4.7366
4	1.0000	0.0000	0.0000	4.7366
5	0.5488	0.2128	0.0784	4.6275
6	0.5488	0.2128	0.0784	4.6275
7	1.0160	0.6095	0.5309	4.5584
…	…	…	…	…
59	0.8634	0.4192	0.2722	4.4630
60	0.8634	0.4192	0.2722	4.4630
61	0.8640	0.4194	0.2721	4.4630

the constraints to observe the change of the optimal solution. Take $N = 10$, and add the following two constraints to the probability distribution, $p_2 + p_3 \le 1/9$, $p_4 + p_5 \le 1/10$, following is the running result,

From Table 3 we can see, the program stops at 64 times, the optimal solution is $x = (0.8929, 0.4113, 0.3524)^{\mathrm{T}}$, optimal function value is $\theta = 3.7382$, the running time is 29.490318 seconds. this shows, on the one hand, as the information of the probability distribution changes from incomplete to complete information, the objective function values of the models (1)-(3) constructed tend to have better results. On the other hand, there is no significant increase in the number of iterations of the algorithm during the optimization process.

5. Conclusion

For the case that the probability distribution has incomplete information, this

Table 2. Iterative results.

iteration	x1	x2	x3	θ
1	1.0000	0.0000	0.0000	5.1179
2	1.0000	0.0000	0.0000	5.1179
3	1.0000	0.0000	0.0000	5.1179
4	1.0000	0.0000	0.0000	5.1179
5	0.5488	0.2128	0.0784	5.0412
6	1.0037	0.6341	0.4862	5.0223
7	1.0037	0.6341	0.4862	5.0223
…	…	…	…	…
87	0.8659	0.3530	0.2681	4.8989
88	0.8659	0.3530	0.2681	4.8989
89	0.8659	0.3530	0.2682	4.8989

Table 3. Iterative results.

iteration	x1	x2	x3	θ
1	1.0000	0.0000	0.0000	4.0385
2	1.0000	0.0000	0.0000	4.0385
3	1.0000	0.0000	0.0000	4.0385
4	1.0000	0.0000	0.0000	4.0385
5	0.5488	0.2128	0.0784	3.9667
6	0.5488	0.2128	0.0784	3.9667
7	0.5937	0.1421	0.2134	3.9321
…	…	…	…	…
62	0.8929	0.4113	0.3524	3.7382
63	0.8929	0.4113	0.3524	3.7382
64	0.8929	0.4113	0.3524	3.7382

paper establishes a stochastic quadratic programming model with incomplete probability distribution based on LPI theory, and designs an improved Nelder-Mead algorithm. The numerical examples are used to calculate the results. The results show that the established models and algorithms are reasonable and effective.

References

[1] Dantzig, G.B. and Ferguson, A.R. (1956) The Allocation of Air Craft Routes-An Example of Linear Programming under Uncertainty. *Management Science*, **3**, 23-34.

[2] Birge, J.R. and Louveanx, F. (2003) Introduction to Stochastic Programming. Springer-Verlag, New York.

[3] Slyke, R. and Wets, R. (1969) L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics*, **17**, 638-663. https://doi.org/10.1137/0117061

[4] Abaffy, J. and Allevi, E. (2004) A Modified L-shaped Method. *Journal of Optimization Theory and Applications*, **11**, 255-270. https://doi.org/10.1007/s10957-004-5148-y

[5] Facchinei, F. and Pang, J.S. (2003) Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer-Verlag, New York.

[6] Qi, L. (1993) Convergence Analysis of Some Algorithms for Solving Non-Smooth Equations. *Mathematics and Operations Research*, **18**, 227-244. https://doi.org/10.1287/moor.18.1.227

[7] Qi, L. and Sun, J. (1993) A Non-Smooth Version of Newton's Method. *Mathematical Programming*, **58**, 353-367. https://doi.org/10.1007/BF01581275

[8] Chen, X., Qi, L, and Womersley, R.S. (1995) Newton's Method for Quadratic Stochastic Programs with Recourse. *Journal of Computational and Applied Mathematics*, **60**, 29-46. https://doi.org/10.1016/0377-0427(94)00082-C

[9] Abdelaziz, F.B. and Masri, H. (2005) Stochastic Programming with Fuzzy Linear Partial Information on Probability Distribution. *European Journal of Operational Research*, **162**, 619-629. https://doi.org/10.1016/j.ejor.2003.10.049

[10] Abdelaziz, F.B. and Masri, H. (2009) Multistage Stochastic Programming with Fuzzy Probability Distribution. *Fuzzy Sets and Systems*, **160**, 3239-3249. https://doi.org/10.1016/j.fss.2008.10.010

[11] Zhang, Y. and Ma, X. (2016) A Class Recourse Stochastic Programs Algorithm with MaxEM in Evaluation. *Operations Research Transactions*, **20**, 52-60.

[12] Kofler, E. (2001) Linear Partial Information with Applications. *Fuzzy Sets and Systems*, **118**, 167-177. https://doi.org/10.1016/S0165-0114(99)00088-3

[13] Boyd, S. and Vandenberghe, L. (2004) Convex Optimization. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511804441

[14] Nelder, J.A. and Mead, R. (1965) A Simplex Method for Function Minimization. *The Computer Journal*, **7**, 308-313. https://doi.org/10.1093/comjnl/7.4.308

[15] Chang, K.H. (2012) Stochastic Nelder-Mead Simplex Method—A New Globally Convergent Direct Search Method for Simulation Optimization. *European Journal of Operational Research*, **220**, 684-694. https://doi.org/10.1016/j.ejor.2012.02.028