Computational Approaches for Solving Systems of Nonlinear Equations

Panos M. Pardalos
ppardalos@toxeus.org

Chief Research Officer - TOXEUS Systems, LLC
https://toxeus.org/

Distinguished Emeritus Professor - University of Florida
https://www.ise.ufl.edu/pardalos/

Agenda

- Introduction to systems of nonlinear equations (SNEs)
- Root finding methods for solving SNEs
- Reformulating SNEs as optimization problems
- Optimization methods for solving SNEs
- Conclusion

What is the Problem?

- Finding one or more solutions to a system of nonlinear equations (SNE) is a challenging and ubiquitous task[1,2]
- The problem of solving even a system of polynomial equations has been proven to be NP-hard[3]
- Related to Hilbert's 10th problem

[1] Ilias S. Kotsireas et al. Survey of Methods for Solving Systems of Nonlinear Equations, Part I: Root-finding Approaches. 2022. DOI: 10.48550/ARXIV.2208.08530. URL: https://arxiv.org/abs/2208.08530.

[2] Ilias S. Kotsireas et al. Survey of Methods for Solving Systems of Nonlinear Equations, Part II: Optimization Based Approaches. 2022. DOI: 10.48550/ARXIV.2208.08532. URL: https://arxiv.org/abs/2208.08532.

[3] C. Jansson. "An NP-Hardness Result for Nonlinear Systems". In: Reliable Computing 4.4 (1998), pp. 345–350.

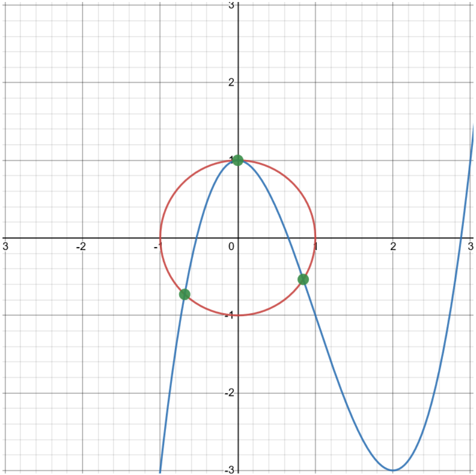
System of Nonlinear Equations (SNE)

Consider a SNE of the form:

$$F_m(x) = \Theta_m \equiv (0, 0, \ldots, 0)^\top \quad \Longleftrightarrow \quad \begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n) = 0, \\ \qquad \vdots \\ f_m(x_1, x_2, \ldots, x_n) = 0, \end{cases}$$

where $F_m = (f_1, f_2, \ldots, f_m) : \mathcal{D}_n \subset \mathbb{R}^n \to \mathbb{R}^m$, where $f_1, f_2, \ldots, f_m$ are real-valued continuous or continuously differentiable functions on the domain $\mathcal{D}_n$, and where at least one of $f_1, f_2, \ldots, f_m$ is nonlinear.

Example SNE



1
$$x^2 + y^2 - 1 = 0$$

2
$$x^3 - 3x^2 + 1 - y = 0$$

Another Example SNE

$$F_2(x) = \Theta_2 \equiv (0,0)^\top \quad \Longleftrightarrow \quad \begin{cases} f_1(x_1,x_2) = x_1 - x_1\sin(x_1+5x_2) - x_2\cos(5x_1-x_2) = 0 \\ f_2(x_1,x_2) = x_2 - x_2\sin(5x_1-3x_2) + x_1\cos(3x_1+5x_2) = 0 \end{cases}$$

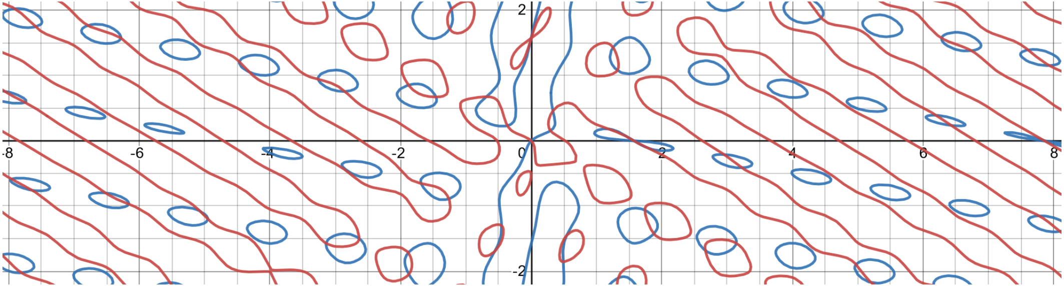


Figure: An example SNE with two equations of two unknowns: Blue - $f_1(x_1,x_2)$; Red - $f_2(x_1,x_2)$

SNE in Industry: Robot Kinematics

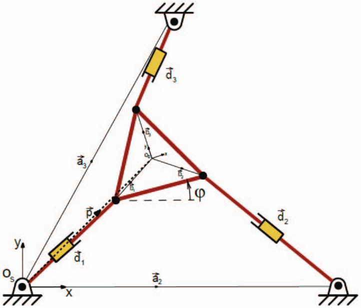


Fig. 1. 3-RPR planar parallel mechanism

4

$$t = \tan\frac{\phi}{2} \quad \sin\phi = \frac{2t}{1+t^2} \quad \cos\phi = \frac{1-t^2}{1+t^2}$$

[4] A. Salimi Lafmejani, A. Kalhor, and M. Masouleh. "A New Development of Homotopy Continuation Method, Applied in Solving Nonlinear Kinematic System of Equations of Parallel Mechanisms". In: in Proc. of the 3rd RSI International Conference on Robotics and Mechatronics (Oct. 2015), pp. 737–742.

SNE in Industry: The Reactor Problem

This problem deals with a model of two continuous nonadiabatic stirred tank reactors with a recycle ratio parameter R:

$$f_1(x_1, x_2) = (1 - R)\left(\frac{D}{10(1 + \beta_1)} - x_1\right) \exp\left(\frac{10x_1}{1 + \frac{10x_1}{\gamma}}\right) - x_1$$

$$f_2(x_1, x_2) = x_1 - (1 + \beta_2)x_2 + (1 - R)\left(\frac{D}{10} - \beta_1 x_1 - (1 + \beta_2)x_2\right) \exp\left(\frac{10x_2}{1 + \frac{10x_2}{\gamma}}\right).$$

When the parameters $\gamma, D, \beta_1,$ and β_2, R are set to 1, 22, 2, 2, and 0.965 respectively, and $x_1, x_2 \in [0, 1]$, this system has 5 solutions.

Industries Where SNEs Arise (Part 1)

- Chemistry[5]
- Chemical engineering[6]
- Automotive steering[7]
- Power flow[8]
- Large-scale integrated circuit designs[9]
- Climate modeling[10]

---

[5] C. A. Floudas. "Global optimization in design and control of chemical process systems". In: Journal of Process Control 10.2 (2000), pp. 125–134.

[6] H. Jiménez-Islas et al. "Nonlinear Homotopic Continuation Methods: A Chemical Engineering Perspective Review". In: Industrial & Engineering Chemistry Research 52.42 (2013), pp. 14729–14742.

[7] N. Henderson, W. Sacco, and G. Platt. "Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization". In: Chemical Engineering Research and Design 88.5 (2010), pp. 551–561.

[8] H. Chiang et al. "Homotopy-Enhanced Power Flow Methods for General Distribution Networks With Distributed Generators". In: IEEE Transactions on Power Systems 29.1 (2014), pp. 93–100.

[9] H. Chiang and T. Wang. "Novel Homotopy Theory for Nonlinear Networks and Systems and Its Applications to Electrical Grids". In: IEEE Transactions on Control of Network Systems 5.3 (2018), pp. 1051–1060.

[10] C. Yang, J. Cao, and X. Cai. "A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere". In: SIAM Journal on Scientific Computing 32.1 (2010), pp. 418–438.

Industries Where SNEs Arise (Part 2)

- Materials engineering[11]
- Robotics[12]
- Nuclear engineering[13]
- Image restoration[14]
- Protein interaction networks[15]

---

[11] M. Schneider, D. Wicht, and T. Böhlke. "On polarization-based schemes for the FFT-based computational homogenization of inelastic materials". In: Computational Mechanics 64.4 (2019), pp. 1073–1095.

[12] Y. Zhang. "A set of nonlinear equations and inequalities arising in robotics and its online solution via a primal neural network". In: Neurocomputing 70.1 (2006). Neural Networks, pp. 513–524.

[13] S. Mavrodiev and M. Deliyergiyev. "Modification of the nuclear landscape in the inverse problem framework using the generalized Bethe-Weizsäcker mass formula". In: International Journal of Modern Physics E 27.2 (2018).

[14] S. Aji et al. "A modified conjugate descent projection method for monotone nonlinear equations and image restoration". In: IEEE Access 8 (2020), pp. 158656–158665.

[15] Chiang and Wang, "Novel Homotopy Theory for Nonlinear Networks and Systems and Its Applications to Electrical Grids".

Industries Where SNEs Arise (Part 3)

- Neurophysiology[16]
- Economics[17]
- Finance[18]
- Applied mathematics[19]
- Physics[20]

---

[16] J. Verschelde, P. Verlinden, and R. Cools. "Homotopies Exploiting Newton Polytopes for Solving Sparse Polynomial Systems". In: SIAM Journal on Numerical Analysis 31.3 (1994), pp. 915–930.

[17] C. Grosan and A. Abraham. "A New Approach for Solving Nonlinear Equations Systems". In: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 38.3 (2008), pp. 698–714.

[18] A. Golbabai, D. Ahmadian, and M. Milev. "Radial basis functions with application to finance: American put option under jump diffusion". In: Mathematical and Computer Modelling 55.3 (2012), pp. 1354–1362.

[19] G. Zhang and L. Bai. "Existence of solutions for a nonlinear algebraic system". In: Discrete dynamics in nature and society 2009 (2009).

[20] K. Kowalski and K. Jankowski. "Towards Complete Solutions to Systems of Nonlinear Equations of Many-Electron Theories". In: Phys. Rev. Lett. 81 (6 1998), pp. 1195–1198.

Industries Where SNEs Arise (Part 4)

- Finding string vacua[21]
- Machine learning[22]
- Geodesy[23]

[21] D. Mehta. "Numerical Polynomial Homotopy Continuation Method and String Vacua". In: Advances in High Energy Physics 2011 (2011), p. 263937.

[22] Y. Song et al. "Nonlinear equation solving: A faster alternative to feedforward computation". In: arXiv preprint arXiv:2002.03629 (2020).

[23] B. Paláncz et al. "Linear homotopy solution of nonlinear systems of equations in geodesy". In: Journal of Geodesy 84.1 (2009), p. 79.

Determining the Number of Solutions to a SNE

- The total number of solutions to a *square SNE* (a SNE that has the same number of equations and decision variables, $n = m$) can be obtained by computing the topological degree

- The topological degree of F_n at $\Theta_n = (0, 0, \dots, 0)^\top$ relative to $\mathcal{D}_n$ can be defined as:

$$\deg[F_n, \mathcal{D}_n, \Theta_n] = \sum_{x \in F_n^{-1}(\Theta_n)} \operatorname{sgn} \det J_{F_n}(x),$$

  where $\det J_{F_n}(x)$ denotes the determinant of the Jacobian matrix and sgn defines the three-valued sign function. The above definition can be generalized when F_n is only continuous[24].

[24] J. Ortega and W. Rheinboldt. Iterative Solution of Nonlinear Equations in Several Variables. Society for Industrial and Applied Mathematics, 2000.

Newton's and Broyden's Methods

- Traditional methods for solving SNEs include the classical Newton's method and Broyden's method

Newton's method

$$x^{k+1} = x^k - J_{F_n}(x^k)^{-1} F_n(x^k),$$

k=0, 1, 2, ...

Broyden's method

(a) **Solve:** the system of linear equations $B_k s^k = -F_n(x^k)$ for s^k.

(b) **Set:** $x^{k+1} = x^k + s^k$.

(c) **Set:** $y^k = F_n(x^{k+1}) - F_n(x^k)$.

(d) **Set:**

$$B_{k+1} = B_k + \frac{1}{(s^k)^\top s^k}(y^k - B_k s^k)(s^k)^\top.$$

Generalizations of Methods for Solving Systems of Linear Equations

- For example, a generalization of the Gauss-Seidel method can be used
- Solve the $i-$th equation for x_i, and set $x_i^{k+1} = x_i$

$$f_i(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \dots, x_n^k) = 0, \quad k = 0, 1, \dots, \quad i = 1, 2, \dots, n.$$

Chebyshev-Halley Method

- Although Newton's method has a quadratic rate of convergence, Halley's method has a cubic rate of convergence

$$x^{k+1} = x^k - \left\{ I + \frac{1}{2} L_{F_n}(x^k) \left[I - \alpha L_{F_n}(x^k) \right]^{-1} \right\} F_n'(x^k)^{-1} F_n(x^k),$$

$$L_{F_n}(x) = F_n'(x)^{-1} F_n''(x) F_n'(x)^{-1} F_n(x)$$

Tensor Methods

- Tensor methods may be used to accelerate convergence when higher order derivatives are available.
- Tensor methods have demonstrated effectiveness at solving large, sparse, and ill-formed SNEs with singular Jacobian matrices.
- For example, the model
$$\|M^k(d)\|_2 = \left\| F_n(x^k) + J_{F_n}(x^k)d + \frac{1}{2}T^k dd \right\|_2 \leqslant \eta^k \|F_n(x^k)\|_2$$
seeks to determine a step d^k at each iteration. In this model, $T^k = F_n''(x^k)$ is the tensor of second derivatives of $F_n(x^k)$, and $\eta^k \in [0,1)$[25].
- The rate of convergence of the above method has been shown to be at least:
  (a) Q-super-linear when $\eta^k \to 0$.
  (b) Q-quadratic when $\eta^k = O(\|F_n(x^k)\|_2)$.
  (c) Q-cubic when $\eta^k = O(\|F_n(x^k)\|_2^2)$.
  (d) Q-order $\min\{\hat{p}, 3\}$ when $\eta^k = O(\|F_n(x^k)\|_2^{\hat{p}-1})$, $1 < \hat{p}$.

[25] T. Steihaug and S. Suleiman. "Rate of convergence of higher order methods". In: Applied Numerical Mathematics 67 (2013), pp. 230–242.

Quasi-Newton Methods

- Newton's method for finding roots requires the Jacobian
- If the Jacobian is not available, an approximation of the Jacobian is utilized
- One famous Quasi-Newton method is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

Levenberg-Marquardt Method

The Levenberg-Marquardt algorithm[26,27] is an iterative procedure that at each iteration calculates an updated solution using the rule

$$x^{k+1} = x^k + \lambda d$$

where the search direction d is found by solving equations of the form

$$\left(J_{F_m}^\top J_{F_m} + \mu\, I_n\right) d = -J_{F_m}^\top F_m,$$

where $J_{F_m} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of F_m, $\mu \in \mathbb{R}$, I_n is the identity matrix, and $F_m : \mathbb{R}^n \to \mathbb{R}^m$.

[26] K. Levenberg. "A method for the solution of certain non-linear problems in least squares". In: Quarterly of Applied Mathematics 2.2 (1944), pp. 164–168.

[27] D. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: SIAM Journal on Applied Mathematics 11.2 (1963), pp. 431–441.

Deflation Techniques for Finding Additional Solutions

- Including the root finding methods discussed in this section, there are a plethora of methods that can be used to search for a single solution to a SNE
- Deflation techniques may be used to help find other solutions[28]
- The SNE is perturbed for each solution that is found to push future search iterations away from already found solutions

[28] K. Brown and W. Gearhart. "Deflation techniques for the calculation of further solutions of a nonlinear system". In: Numerische Mathematik 16 (1971).

Deflation Techniques for Finding Additional Solutions

- Examples of deflated functions include

$$\hat{f}_i(x) = \frac{1}{\prod_{j=1}^{p} \|x - r_j\|} f_i(x), \qquad i = 1, 2, \ldots, n,$$

and

$$\tilde{f}_i(x) = \frac{1}{\prod_{j=1}^{p} \langle \nabla f_i(r_j), (x - r_j) \rangle} f_i(x), \qquad i = 1, 2, \ldots, n,$$

where r_1, $r_2, \ldots, r_p$ are the p already computed roots.

Symbolic Computation Methods

- Symbolic Computation methods can be utilized to find exact solutions to polynomial SNEs
- Two such methods within the realm of Symbolic Computation include the theories of Resultants and Gröbner Bases[29]
- Symbolic Computation methods can be applied through the use of Computer Algebra Systems (CAS) (see[30] for more information)
- Major CAS include Maple, Magma, Mathematica, and MATLAB among others
- Smaller, more specialized CAS include Singular, Fermat, and Cocoa

[29] K. Geddes, S. Czapor, and G. Labahn. Algorithms for computer algebra. Kluwer Academic Publishers, Boston, MA, 1992.

[30] M. Wester. Computer Algebra Systems: A Practical Guide. John Wiley & Sons, Chichester, United Kingdom, 1999.

Symbolic Computation Methods: Resultants

- Resultants can be used to:
  - Solve polynomial SNEs.
  - Determine whether or not solutions exist to polynomial SNEs.
  - Reduce a given polynomial SNE to one with fewer variables and/or fewer equations.
- The resultant of two univariate polynomials of degrees n, m respectively is defined as the determinant of the $(n+m) \times (n+m)$ Sylvester matrix associated with the two polynomials.
- The Macaulay resultant can be utilized to determine if a polynomial SNE has a solution. The Macaulay resultant can be computed as the quotient of two determinants, and it is zero if and only if the system has a solution.

Symbolic Computation Methods: Gröbner Bases

- Gröbner Bases were invented by Bruno Buchberger as a method which generalizes Euclid's algorithm and Gauss elimination for polynomial SNEs.
- For systems with a finite number of solutions, the Gröbner Basis of the system can be computed with respect to a lexicographical ordering of the variables.
- The Gröbner Basis of the system has the property that it contains a univariate polynomial $p(x)$ of a certain degree. Subsequently, $p(x)$ is solved and via a back-substitution process, we can recover all the solutions to the SNE (resemblance with Gauss elimination).
- Over the past 50+ years, the theory and applications of Gröbner Bases have developed into an entire autonomous research area, as documented in: `https://www3.risc.jku.at/research/theorema/Groebner-Bases-Bibliography/`
- For some polynomial SNEs, Buchberger's algorithm can generate large intermediate sets of polynomials until reduction of the polynomial ideal to the Gröbner Basis is achieved.

Homotopy / Continuation Methods

- Homotopy / Continuation methods can be utilized to search for solutions to polynomial SNEs
- Homotopy (or deformation) of a system $F_n(x)$ is a function H_n such that

$$H_n(x, 1) = G_n(x)$$

$$H_n(x, 0) = F_n(x)$$

  where the roots of $G_n(x)$ are known
- We can numerically trace a curve starting from $(x, 1)$ and ending at solution point $(x, 0)$ by decreasing λ by small increments $\Delta\lambda$

$$H_n(x, \lambda) = \lambda G_n(x) + (1 - \lambda)F_n(x)$$

Homotopy / Continuation Methods (see[31])

1. Choose $x_1 \in \mathbb{R}^n$ such that $H_n(x_1, 1) = 0$; choose integer m
   - In the case of polynomial SNEs it is trivial to find solutions of the starting system $G_n(x) = \Theta_n$
2. $x := x_1; \lambda := (m-1)/m; \Delta\lambda = 1/m$
3. for $i = 1, ..., m$:
   - solve $H_n(y, \lambda) = 0$ for y using x as a starting value using an iterative method (for example, Newton's method)
   - set $x := y; \lambda := \lambda - \Delta\lambda$
4. output x as a solution

[31] E. Allgower and K. Georg. Introduction to Numerical Continuation Methods. SIAM, 2003.

TOXEUS

Interval Methods

- *Interval arithmetic* was introduced in article[32].
- Interval arithmetic accounts for rounding errors due to limited machine precision, errors in measurements, and other uncertainty arising in practical problems.
- In interval arithmetic a value is represented by a lower and upper bound which provide reliable results during computations.
- Interval arithmetic was standardized by the IEEE in 2015, and there are multiple packages which implement interval arithmetic.

[32] R. Moore. Interval analysis. Prentice-Hall, 1966.

Interval Methods

- A real interval X is defined as a set of real numbers between lower and upper bounds
$$X = [a, b] = \{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$$
- Interval arithmetic is a set of operations such as addition, subtraction, multiplication, and division defined on the intervals. For example:
  - $[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$
  - $[x_1, x_2] \cdot [y_1, y_2] = [\min\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}, \max\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}]$
- Interval functions, domain and range of which are the intervals, have been defined
- Global optimization problems can be solved using the *interval branch-and-bound method* which iteratively splits the search space and removes parts that do not contain a global solution[33]

[33] H. Ratschek and J. Rokne. "Interval Global Optimization". In: Encyclopedia of Optimization, Second Edition. Ed. by C. Floudas and P. Pardalos. Springer, 2009, pp. 1739–1757.

Reformulating a SNE as an Optimization Problem

Original SNE

$$F_m(x) = \Theta_m \equiv (0, 0, \ldots, 0)^\top \iff \begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n) = 0, \\ \vdots \\ f_m(x_1, x_2, \ldots, x_n) = 0, \end{cases}$$

$$F_m = (f_1, f_2, \ldots, f_m) : \mathcal{D}_n \subset \mathbb{R}^n \to \mathbb{R}^m$$

Reformulation as an Optimization Problem

$$\arg\min_{x \in \mathcal{D}} \varphi^0(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p$$

$$\alpha > 0, p > 0$$

TOXEUS

Other Reformulations

- SNEs have also been transformed into multi-objective optimization problems and single-objective constrained optimization problems:

  1. Reformulation as a multi-objective optimization problem[34,35]:

  $$\min \left[|f_1(x_1, x_2, \ldots, x_n)|, \ldots, |f_m(x_1, x_2, \ldots, x_n)| \right]$$

  2. Reformulation as a single-objective constrained optimization problem[36]:

  $$\min \sum_{i=1}^{m} f_i(x), \ \ni f_i(x) \geq 0 \ \ \forall \ \ i = 1, \ldots, m.$$

[34] Grosan and Abraham, "A New Approach for Solving Nonlinear Equations Systems".

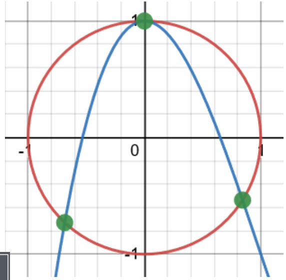
[35] W. Song et al. "Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization". In: IEEE Transactions on Evolutionary Computation 19.3 (2015), pp. 414–431.

[36] A. Kuri. "Solution of Simultaneous Non-Linear Equations using Genetic Algorithms". In: WSEAS Transactions on Systems 2 (2003).

Example of a SNE Reformulated as an Optimization Problem

Original SNE

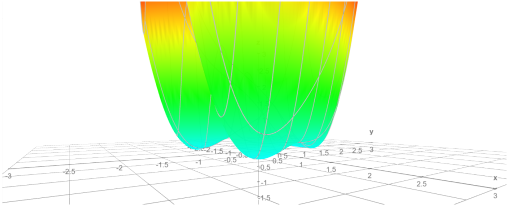
$$x^2 + y^2 - 1 = 0$$
$$x^3 - 3x^2 + 1 - y = 0$$



Reformulation as an Optimization Problem

$$\arg\min_{x \in \mathcal{D}} \varphi^0(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p \text{ (Let } \alpha = 1, p = 2)$$

$$\arg\min_{(x,y) \in \mathcal{D} \subset \mathbb{R}^2} \varphi^0(x,y) = (x^2 + y^2 - 1)^2 + (x^3 - 3x^2 + 1 - y)^2$$



General Characteristics of the Reformulated Global Optimization Problem

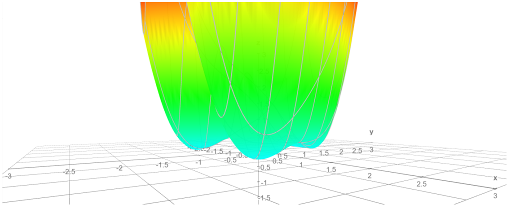
General Characteristics of the Reformulated Optimization Problem

- Non-linear
- Non-convex
- May be non-differentiable

Reformulation as an Optimization Problem

$$\arg\min_{x \in \mathcal{D}} \varphi^0(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p \text{ (Let } \alpha = 1, p = 2)$$

$$\arg \min_{(x,y) \in \mathcal{D} \subset \mathbb{R}^2} \varphi^0(x,y) = (x^2 + y^2 - 1)^2 + (x^3 - 3x^2 + 1 - y)^2$$



Penalized Objective Function for Finding Multiple Solutions to a SNE

- To find multiple solutions to a SNE via global optimization, after finding one or more solutions, a reformulated objective function can be use to push future search iterations away from previously found solutions

Initial reformulation used to find a first solution

$$\arg\min_{x \in \mathcal{D}} \varphi^0(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p$$

where $\alpha > 0, p > 0$

Penalized reformulation to find additional solutions (assume k solutions have been found so far)

$$\begin{aligned}\arg\min_{x \in \mathcal{D}} \varphi(x) &= \varphi^0(x) + \varphi^k(x) \\ &= \alpha \sum_{i=1}^{m} |f_i(x)|^p + \sum_{j=0}^{k} a\, \xi\left(\frac{\|x - x^j\|_2}{\rho}\right)\end{aligned}$$

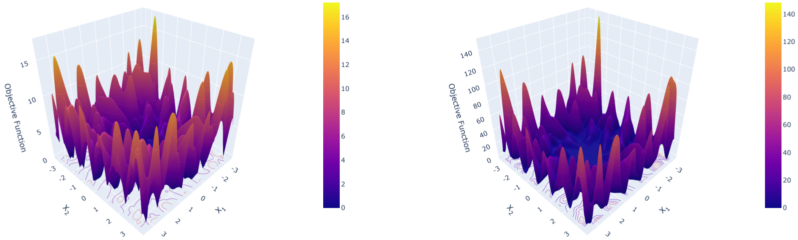
where $\alpha > 0, p > 0, a > 0, \rho > 0$, and $\xi(\delta)$ is a continuous real function that is positive for $|\delta| < 1$ and zero elsewhere

Example of a Non-Penalized Objective Function

- A non-penalized objective function surface (no solutions have been found so far)

$$F_2(x) = \Theta_2 \equiv (0,0)^\top \quad \Longleftrightarrow \quad \begin{cases} f_1(x_1,x_2) = x_1 - x_1\sin(x_1+5x_2) - x_2\cos(5x_1-x_2) = 0 \\ f_2(x_1,x_2) = x_2 - x_2\sin(5x_1-3x_2) + x_1\cos(3x_1+5x_2) = 0 \end{cases}$$

$$\arg\min_{x\in\mathcal{D}} \varphi(x) = \varphi^0(x) = \alpha\sum_{i=1}^{m} |f_i(x)|^p$$



(a) $\alpha = 1,\ p = 1$ (b) $\alpha = 1,\ p = 2$

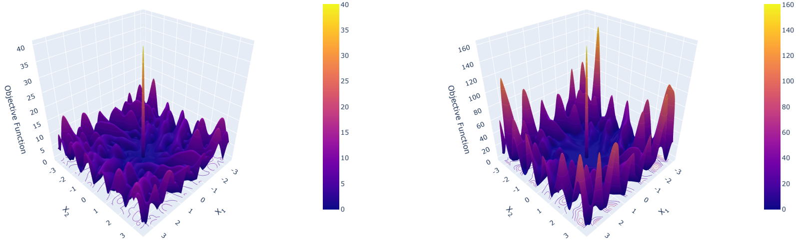
TOXEUS

Example of a Penalized Objective Function

- A penalized objective function surface with a single found solution ($k = 1$) at $(0, 0)$. Notice the sharply elevated region around the point $(0, 0)$.

$$\arg\min_{x \in \mathcal{D}} \varphi(x) = \varphi^0(x) + \varphi^k(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p + \sum_{j=0}^{k} a\,\xi\left(\frac{\|x - x^j\|_2}{\rho}\right)$$

$$\text{where } \xi(\delta) = (2 - \delta^2)^2 \text{ for } |\delta| < \sqrt{2} \text{ and } 0 \text{ otherwise}$$



(c) $\alpha = 1,\ p = 1,\ a = 1,\ \rho = 1$ (d) $\alpha = 1,\ p = 2,\ a = 4,\ \rho = 1$

Other Penalized Objective Functions (Part 1)

- Other penalized objective functions have also been utilized:

  1. $\varphi(x) = \varphi^0(x) + \varphi^k(x) = \alpha \sum_{i=1}^{m} |f_i(x)|^p + \beta \sum_{j=0}^{k} \exp\left(-\|x - x^j\|_2\right) \chi\left(\|x - x^j\|_2\right)$ where β is a large constant, ϱ is a small constant, $\chi(\delta) = 1$ when $\delta < \varrho$ and otherwise $\chi(\delta) = 0$[37,38]

  2. $\varphi(x) = \dfrac{\varphi^0(x)}{\prod_{j=1}^{k} \arctan \|x - x^j\|_2}$[39]

[37] M. Hirsch, P. Pardalos, and M. Resende. "Solving systems of nonlinear equations with continuous GRASP". In: Nonlinear Analysis: Real World Applications 10.4 (2009), pp. 2000–2006.

[38] R. Silva, M. Resende, and P. Pardalos. "Finding multiple roots of a box-constrained system of nonlinear equations with a biased random-key genetic algorithm". In: Journal of Global Optimization 60.2 (2014), pp. 289–306.

[39] Henderson, Sacco, and Platt, "Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization".

TOXEUS

Other Penalized Objective Functions (Part 2)

- Other penalized objective functions have also been utilized (Cont.):

  1. $\varphi(x) = \left(\varphi^0(x) + \varepsilon\right) \Pi_{j=1}^{k} \left|\coth(\alpha \|x - x^j\|_2)\right|$ where $\varepsilon > 0$ is a small user defined constant and $\alpha \geqslant 1$ is a user defined parameter which is utilized to adjust the radius of the penalty region[40]
  2. $\varphi(x) = \varphi^0(x) + \sum_{j=1}^{k} \xi(x, x^j, \alpha, \rho)$ where $\xi(x, x^j, \alpha, \rho) = \alpha\left(1 - \mathrm{erf}(\|x - x^j\|_2)\right)$ if $\|x - x^j\|_2 \leqslant \rho$ and 0 otherwise, where $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$, and where $\rho > 0$ and $\alpha > 0$ are user defined parameters which adjust the radius of the penalty region and the magnitude of the penalty respectively

[40] E. Pourjafari and H. Mojallali. "Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering". In: Swarm and Evolutionary Computation 4 (2012), pp. 33–43.

TOXEUS

Global Optimization

- The objective of global optimization is to find the global minima or maxima of a function or a set of functions over a given domain.

- The books[41,42] provide a nice introduction to global optimization, the book[43] focuses specifically on heuristics for global optimization, and[44,45] discuss strategies for developing portfolios of algorithms for global optimization

[41] R. Horst and P. Pardalos, eds. Handbook of Global Optimization. Nonconvex Optimization and Its Applications. Springer US, 1995.

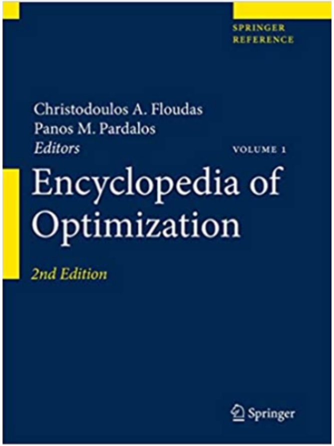
[42] P. Pardalos and H. Romeijn, eds. Handbook of Global Optimization, Volume 2. Springer US, 2002.

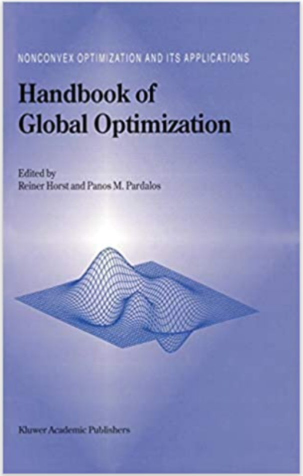
[43] R. Martí, P. Pardalos, and M. Resende, eds. Handbook of Heuristics. Springer International Publishing, 2018.

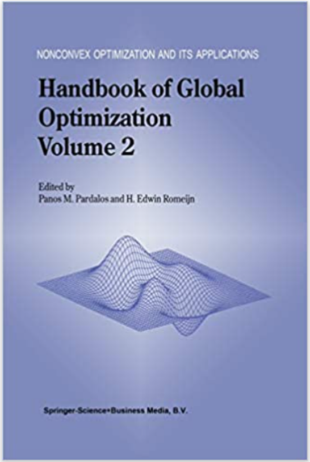
[44] D. Souravlias et al. "Parallel algorithm portfolios with performance forecasting". In: Optimization Methods and Software 34.6 (2019), pp. 1231–1250.

[45] D. Souravlias et al. Algorithm Portfolios: Advances, Applications, and Challenges. Springer, 2021.

Global Optimization







Classes of Global Optimization Methods

A taxonomy of global optimization methods can include the following (among others):

Exact Methods

- Branch-and-bound methods
- Inner and outer approximation
- Cutting-plane methods
- Interval methods
- Methods based on real algebraic geometry

Stochastic Methods

- Direct Monte-Carlo Sampling
- Stochastic Tunneling
- Parallel Tampering

Metaheuristics

- Single-solution-based methods
- Population-based methods

TOXEUS

Global Optimization Metaheuristics Used to Solve SNEs (Part 1)

A taxonomy of metaheuristics for global optimization can include (among others):

Single-solution-based Methods

- Continuous Variable Neighborhood Search (C-VNS)[46]
- Continuous Greedy Randomized Adaptive Search Procedure (C-GRASP)[47,48,49]
- Simulated Annealing (SA)[50,51]
- Tabu Search (TS)[52]

---

46 J. Pei et al. "Continuous variable neighborhood search (C-VNS) for solving systems of nonlinear equations". In: INFORMS Journal on Computing 31.2 (2019), pp. 235–250.

47 Hirsch, Pardalos, and Resende, "Solving systems of nonlinear equations with continuous GRASP".

48 M. Hirsch et al. "Global optimization by continuous GRASP". In: Optimization Letters 1.2 (2007), pp. 201–212.

49 M. Hirsch, P. Pardalos, and M. Resende. "Speeding up continuous GRASP". In: European Journal of Operational Research 205.3 (2010), pp. 507–521.

50 Henderson, Sacco, and Platt, "Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization".

51 H. e Oliveira and A. Petraglia. "Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing". In: Applied Soft Computing 13.11 (2013), pp. 4349–4357.

52 G. Ramadas and E. Fernandes. "Self-adaptive combination of global tabu search and local search for nonlinear equations". In: International Journal of Computer Mathematics 89.13-14 (2012), pp. 1847–1864.

Global Optimization Metaheuristics Used to Solve SNEs (Part 2)

A taxonomy of metaheuristics for global optimization can include (among others) (Cont.):

Population-based Methods

- Evolutionary Algorithms (EAs)[53,54]
- Genetic Algorithms (GAs)[55,56,57]
- Differential Evolution (DE)[58,59]

---

53 Grosan and Abraham, "A New Approach for Solving Nonlinear Equations Systems".

54 Song et al., "Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization"; H. Geng et al. "Research of Ranking Method in Evolution Strategy for Solving Nonlinear System of Equations". In: 2009 First International Conference on Information Science and Engineering. 2009, pp. 348–351.

55 Kuri, "Solution of Simultaneous Non-Linear Equations using Genetic Algorithms".

56 Silva, Resende, and Pardalos, "Finding multiple roots of a box-constrained system of nonlinear equations with a biased random-key genetic algorithm".

57 M. A. El-Shorbagy and Adel M. El-Refaey. "Hybridization of Grasshopper Optimization Algorithm With Genetic Algorithm for Solving System of Non-Linear Equations". In: IEEE Access 8 (2020), pp. 220944–220961; Weifeng Gao et al. "Solving Nonlinear Equation Systems by a Two-Phase Evolutionary Algorithm". In: IEEE Transactions on Systems, Man, and Cybernetics: Systems 51.9 (2021), pp. 5652–5663.

58 M. Tawhid and A. Ibrahim. "A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems". In: Evolving Systems 11.1 (2020), pp. 65–87.

59 W. He et al. "Fuzzy neighborhood-based differential evolution with orientation for nonlinear equation systems". In: Knowledge-Based Systems 182 (2019), p. 104796.

Global Optimization Metaheuristics Used to Solve SNEs (Part 3)

A taxonomy of metaheuristics for global optimization can include (among others) (Cont.):

Population-based Methods (Cont.)

- Particle Swarm Optimization (PSO)[60,61,62]
- Spiral Dynamics Optimization (SPO)[63]
- Other metaphor-based metaheuristics[64,65,66,67]

60 A. Ouyang, Y. Zhou, and Q. Luo. "Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations". In: 2009 IEEE International Conference on Granular Computing. IEEE. 2009, pp. 460–465.

61 A. Ibrahim and M. Tawhid. "A hybridization of cuckoo search and particle swarm optimization for solving nonlinear systems". In: Evolutionary Intelligence 12.4 (2019), pp. 541–561.

62 Y. Li, Y. Wei, and Y. Chu. "Research on solving systems of nonlinear equations based on improved PSO". In: Mathematical Problems in Engineering 2015 (2015).

63 K. Sidarto and A. Kania. "Finding All Solutions of Systems of Nonlinear Equations Using Spiral Dynamics Inspired Optimization with Clustering". In: Journal of Advanced Computational Intelligence and Intelligent Informatics 19 (Sept. 2015), pp. 697–707.

64 Pourjafari and Mojallali, "Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering".

65 X. Zhang, Q. Wan, and Y. Fan. "Applying modified cuckoo search algorithm for solving systems of nonlinear equations". In: Neural Computing and Applications 31.2 (2019), pp. 553–576.

66 Y. Zhou, J. Liu, and G. Zhao. "Leader glowworm swarm optimization algorithm for solving nonlinear equations systems". In: Electrical Review 88.1 (2012), pp. 101–106.

67 R. Jia and D. He. "Hybrid artificial bee colony algorithm for solving nonlinear system of equations". In: 2012 Eighth international conference on computational intelligence and security. IEEE. 2012, pp. 56–60.

Simulated Annealing (SA)

- Annealing is the process of heating a metal and then slowly cooling it down until it solidifies.
- At higher temperatures, the atoms in the metal have high energy, and more freedom to move around and arrange themselves.
- As the temperature decreases, the energy in the atoms reduces – they are not able to move around as much.
- If a metal is allowed to cool slowly, crystals with regular structure are obtained, at an atomic state of minimal energy.
- If the metal is cooled too quickly (rapid quenching), defects and irregularities appear in the crystal structure. The atoms do not reach their minimal energy state. This results in a very brittle solid.
- Simulated Annealing (SA) is the concept of applying the physical annealing process of metals to the software process of finding the global minima of an optimization problem[68].

[68] S. Kirkpatrick, D. Gelatt, and M. Vecchi. "Optimization by Simulated Annealing". In: Science 220.4598 (1983), pp. 671–680.

Simulated Annealing (SA)

Generalized SA Pseudo-Code

procedure SA(InitTemp, FinalTemp, ΔT, $f(\cdot)$)
    $x \leftarrow$ RandomFeasibleSoln ();
    $T \leftarrow$ InitTemp;
    while $T \geq$ FinalTemp **do**
        $\hat{x} \leftarrow$ PerturbSoln (x);
        $r \leftarrow$ RandomUniform $(0, 1)$;
        if $f(\hat{x}) < f(x)$ **or** $r < e^{\frac{-[f(\hat{x})-f(x)]}{T}}$ **then**
            $x \leftarrow \hat{x}$;
        else
            $T \leftarrow T - \Delta T$;
        end if
    end while
    return(x);
end SA;

Continuous Greedy Randomized Adaptive Search Procedure (C-GRASP)

High-Level C-GRASP Pseudo-Code

```
procedure GRASP(Problem Instance)
    InputInstance();
    while stopping criteria not met do
        ConstructGreedyRandomizedSolution(Solution);
        LocalSearch(Solution);
        if Solution is better than BestSolution then
            UpdateBestSolution(Solution,BestSolution);
        end if
    end while
    return(BestSolution);
end GRASP;
```

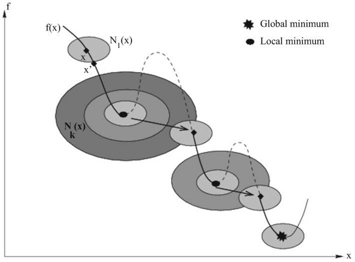
C-GRASP's Construction Phase

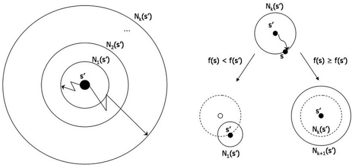
```
procedure ConstructGreedyRandomizedSolution(Problem Instance)
    Solution ← ∅;
    while Solution construction not done do
        MakeRCL(RCL);
        S ← SelectRandomElement(RCL);
        Solution ← Solution ∪ {S};
        AdaptGreedyFunction(S);
    end while
    return(Solution);
end ConstructGreedyRandomizedSolution;
```

C-GRASP's Local Search Phase

```
procedure LocalSearch(Solution, Neighborhood))
    Solution* ← Solution;
    while Solution* not locally optimal do
        Solution ← SelectRandomElement(Neighborhood(Solution* ));
        if Solution better than Solution* then
            Solution* ← Solution;
        end if
    end while
    return(Solution* );
end LocalSearch;
```

Continuous Variable Neighborhood Search (C-VNS)





C-VNS base algorithm

/* Initialization */
1. Select the set of neighborhood structures $\mathcal{N}_k$, $k = 1, \dots, k_{\max}$ with the corresponding random distributions.
2. Choose an arbitrary initial point $x \in S$
3. Set $x^* \leftarrow x$, $F^* \leftarrow F(x)$

/* *Main loop* */
4. **repeat** the following steps **until** the stopping condition is met
5. Set $k \leftarrow 1$
6. **repeat** the following steps **until** $k > k_{\max}$
7. *Shake:* Generate at random a point $y \in \mathcal{N}_k(x^*)$
8. Apply some *local search* method from y to obtain a local minimum y'
9. **if** $F(y') < F^*$ **then**
10. Set $x^* \leftarrow y'$, $F^* \leftarrow F(y')$ and **goto** 5
11. **endif**
12. Set $k \leftarrow k + 1$
13. **end**
14. **end**
15. **Stop**. Point x^* is an approximate solution of the problem.

Finding multiple solutions

/* *Initialization* */
1. Select p, ρ_0, $\rho_{\min}$, a_0, $q_\rho < 1$, $q_a < 1$, tol, K and function $\varphi(x)$
2. Set $k = 0$, $\rho \leftarrow \rho_0$, $a \leftarrow a_0$

/* *Main loop* */
3. **while** $k < K$ and $\rho \geq \rho_{\min}$
4. Perform global optimization for function (6) by C-VNS resulting with x^* as the best approximation of optimal point
5. **if** $F(x_*) < tol$ **then**
6. Set $k \leftarrow k + 1$
7. Set $sol(k) = x^*$
8. **else**
9. Set $\rho \leftarrow q_\rho \cdot \rho$
10. Set $a \leftarrow q_a \cdot a$
11. **endif**
12. **endwhile**
13. **Stop**. Points $sol(i)$, $i = 1, \dots, k$ are approximate solutions of (1).

Comparison of Two C-VNS Variants with C-GRASP for the Reactor Problem

		$p = 1,\ \xi(\delta) = 1 - \lvert\delta\rvert$			$p = 2,\ \xi(\delta) = (1 - \delta^2)^2$			$p = 2$, GRASP	
R	No. of solutions	Solutions	Time	Computer efforts	Solutions	Time	Computer efforts	Solutions	Time
0.935	1	1	< 1 ms	1,364	1	< 1 ms	649	1	0.600
0.940	1	1	< 1 ms	1,443	1	< 1 ms	609	1	0.770
0.945	3	3	0.001	5,674	3	0.001	2,810	3	0.190
0.950	5	5	0.018	81,332	5	0.011	48,143	4.99	1.110
0.955	5	5	0.003	13,038	5	0.003	11,141	5	1.690
0.960	7	7	0.005	19,566	7	0.005	21,512	6.96	2.410
0.965	5	5	0.003	15,540	5	0.001	5,793	4.95	1.810
0.970	5	5	0.001	4,309	5	0.001	4,087	4.99	1.340
0.975	5	5	0.003	15,561	5	0.002	6,773	4.98	1.830
0.980	5	5	0.002	9,134	5	0.001	3,900	4.98	1.900
0.985	5	5	0.002	8,980	5	0.002	6,161	4.99	2.230
0.990	1	1	< 1 ms	608	1	< 1 ms	428	1	0.010
0.995	1	1	< 1 ms	201	1	< 1 ms	69	1	0.010

Comparison of C-VNS and the Toxeus Cloud Solver for a Discrete Integral Equation Problem

- Martinez[70] introduced the following Discrete Integral Equation problem, which is a dense nonlinear system over an arbitrary number of variables, n.

$$f_i(x) = x_i + \frac{h}{2} \cdot \left[(1 - t_i) \cdot \sum_{j=1}^{i} t_j \cdot (x_j + t_j + 1)^3 + t_i \cdot \sum_{j=i+1}^{n} (1 - t_j) \cdot (x_j + t_j + 1)^3\right]$$

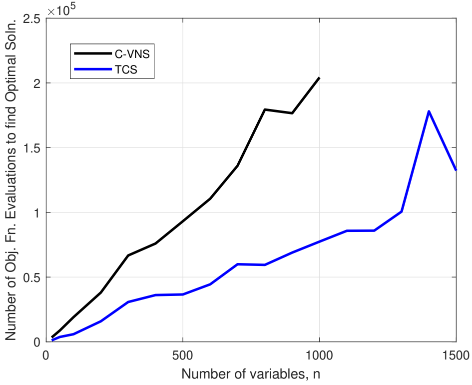
- In this system, $i = 1, \ldots, n$, $h = \frac{1}{n+1}$, and $t_i = i \cdot h$.
- The C-VNS algorithm solved this system for various values of n, up to $n = 1,000$.
- So far, the Toxeus Cloud Solver (TCS) was able to solve this system up to $n = 1,500$, in each instance needing significantly less function evaluations as compared with C-VNS.

[70] J. Martinez. "Solving systems of nonlinear equations by means of an accelerated successive orthogonal projection method". In: Journal of Computational and Applied Mathematics 16.2 (1986), pp. 169–179.

Comparison of C-VNS and the Toxeus Cloud Solver (TCS) for a Discrete Integral Equation Problem (Cont.)

n	C-VNS	TCS
20	3, 362	1, 141
50	8, 745	3, 747
100	19, 062	5, 873
200	38, 072	15, 863
300	66, 703	30, 757
400	75, 931	36, 110
500	93, 145	36, 502
600	110, 558	44, 408
700	135, 999	59, 914
800	179, 410	59, 425
900	176, 410	68, 928
1, 000	204, 244	77, 413
1, 100	—	85, 768
1, 200	—	85, 872
1, 300	—	100, 583
1, 400	—	178, 045
1, 500	—	132, 236

Table: Number of function evaluations to find the optimal solution.



Summary & Opportunities

- For the hardest classes of SNEs, many of the most promising techniques that we currently have are metaheuristics.
- Although metaheuristics are not guaranteed to converge to a solution in finite time, many can be applied to ill-formed and challenging classes of SNEs as they only require evaluations of the objective function.
- More research needs to be done to develop faster and more robust methods for solving SNEs.
- Future research efforts could explore ways to utilize machine learning and statistical analysis to more effectively and efficiently search for solutions to SNEs.

Questions?

TOXEUS Systems, LLC
https://toxeus.org/

Panos M. Pardalos
ppardalos@toxeus.org