SCIENCE CHINA
Information Sciences

• LETTER •

July 2022, Vol. 65 179102:1–179102:3
https://doi.org/10.1007/s11432-019-2882-0

Locating multiple roots of nonlinear equation systems via multi-strategy optimization algorithm with sequence quadratic program

Jing LIANG[1], Boyang QU[1,4], Baolei LI[1,2,3*], Kunjie YU[1] & Caitong YUE[1]

[1]*School of Electrical Engineering, Zhengzhou University, Zhengzhou* 450001*, China;*
[2]*School of Physics and Electronic Engineering, Nanyang Normal University, Nanyang* 473061*, China;*
[3]*Key Laboratory for the Structure and Evolution of Celestial Objects, Chinese Academy of Sciences, Kunming* 650011*, China;*
[4]*School of Electronic and Information Engineering, Zhongyuan University of Technology, Zhengzhou* 450007*, China*

Received 1 November 2019/Revised 26 February 2020/Accepted 20 April 2020/Published online 14 May 2021

Citation Liang J, Qu B Y, Li B L, et al. Locating multiple roots of nonlinear equation systems via multi-strategy optimization algorithm with sequence quadratic program. Sci China Inf Sci, 2022, 65(7): 179102, https://doi.org/10.1007/s11432-019-2882-0

Dear editor,

Locating all the roots of a nonlinear equation system (NES) is not only of great significance for solving real-world problems but also one of the core problems of mathematics [1].

A NES can be defined by

$$\begin{cases} f_1(\boldsymbol{x}) = 0, \\ \quad\vdots \\ f_m(\boldsymbol{x}) = 0, \end{cases} \tag{1}$$

where m is the number of equations, $\boldsymbol{x} = (x_1, \ldots, x_d)$ is a vector with d dimensions. The variable of the ith dimension x_i is in the interval $(x_i^{\mathrm{U}}, x_i^{\mathrm{L}})$, where x_i^{U} and x_i^{L} are respectively the upper and lower bounds.

Evolutionary algorithms (EAs) have been successfully used for solving NES problems by transforming a NES problem into an optimization problem [2]. Especially, EAs with multiple strategies have shown a good performance [3, 4]. However, most EAs treat NES problems as black-box, and the information in explicit function expression is not fully used. Consequently, EAs have to use many fitness evaluations (FEs) to find the exact roots of NES.

The sequence quadratic program (SQP) technique for constrained optimization problems can fast locate the exact roots of NES when it is started from approximate roots. Because the information in function expression of NES is used by SQP to guide the search efficiently. Moreover, Ref. [5] has proven the effectiveness of hybridizing SQP with EAs. There exist studies hybridizing SQP with EAs for global optimization problems. But, there are only a few methods that hybridized SQP with multimodal optimization EAs, which have been used for tackling NES problems.

Above all, the multi-strategy multimodal optimization algorithm with sequence quadratic program (SQP-MOA) was proposed for NES problems based on our previous work, i.e., multi-strategy optimization algorithm (MOA) [4]. In SQP-MOA, MOA is used to fast locate multiple approximate roots and then SQP is implemented to find the exact roots starting from these approximate solutions. To test the ability of the proposed methods in complex problems, we additionally proposed ten complex NES benchmark problems, where many local traps existed.

SQP-MOA is compared with other well-established methods for NES problems based on systematic experiments on fifty-two NES problems and four real-world engineering problems. The results demonstrate that SQP-MOA is very competitive in success rate, peak ratio and success performance. Moreover, we investigate the contributions of different components to improving SQP-MOA search capability and the influence of different parameter settings on the performance of SQP-MOA.

Transform method. To locate multiple approximate roots of NES by using MMOA, a NES problem can be transformed into a multimodal optimization problem (MOP) as

$$\sum_{i=1}^{m} (f_i(\boldsymbol{x}))^2. \tag{2}$$

To find the exact roots by using SQP, a NES problem can be transformed into a constrained optimization problem (COP) as

$$\min \sum_{i=1}^{m-1} (f_i(\boldsymbol{x}))^2 \quad \text{s.t. } f_m(\boldsymbol{x}) = 0. \tag{3}$$

Multi-strategy optimization algorithm with sequence quadratic program. MOA searches the solution space by iterative exploration and exploitation, which are respectively performed by global searchers and local searchers [4].

* Corresponding author (email: bl_li@qq.com)

The qth global searcher is expressed by

$$A^{q,0} = [a_1^{q,0}, \ldots, a_d^{q,0}], \quad q \in [1, \ldots, N_{\mathrm{g}}], \tag{4}$$

where $a_i^{q,0} = x_i^{\mathrm{L}} + \mathrm{rand}(0,1) \times (x_i^{\mathrm{U}} - x_i^{\mathrm{L}})$ is the ith dimension of $A^{q,0}$, d is the number of variables, the function rand(0, 1) generates uniformly distributed random number in the interval (0, 1), N_{g} is the number of global searchers.

The jth local searcher $A^{q,j}$, near $A^{q,0}$, is expressed by

$$A^{q,j} = [a_1^{q,j}, \ldots, a_d^{q,j}], \quad j \in [1, \ldots, N_{\mathrm{l}}], \tag{5}$$

where $a_i^{q,j} = a_i^{q,0} + r \times (2 \times \mathrm{rand}(0,1) - 1) \times (x_i^{\mathrm{U}} - x_i^{\mathrm{L}}), i = 1, 2, \ldots, d, j = 1, 2, \ldots, N_{\mathrm{l}}$. If $a_i^{q,j} > x_i^{\mathrm{U}}$ or $a_i^{q,j} < x_i^{\mathrm{L}}$, it is set as x_i^{U} or x_i^{L} respectively. N_{l} is the number of local searchers in each subgroup. r is a parameter, which determines the scope of the local search area.

To make MOA less sensitive to parameter r, the Bimodal-gauss distributed random number is used to generate a smaller r and a larger r randomly. The smaller r is helpful in refining the intermediate solution and the larger r is helpful in escaping from the local trap. The probability density distribution of r is described by

$$p(r) = \frac{0.5}{\delta_1 \sqrt{2\pi}} \mathrm{e}^{-(r-\mu_1)^2/2\delta_1^2} + \frac{0.5}{\delta_2 \sqrt{2\pi}} \mathrm{e}^{-(r-\mu_2)^2/2\delta_2^2}, \tag{6}$$

where the mean values and standard deviations of the two modes of Bimodal-gauss are (μ_1, δ_1) and (μ_2, δ_2). $\delta_1 = 0.5\mu_1$, $\mu_2 = \alpha \times \mu_1$, $\delta_2 = 0.5(\alpha - 2)\mu_1, (\alpha > 3)$ [6]. The distribution of r_i is determined by μ_1 and α.

One iteration in SQP-MOA included seven steps. (1) Local searchers are generated to exploit local areas, which are centered on selected global searchers. (2) New local searchers are evaluated. If a local searcher is better than its corresponding global searcher, it will replace the global searcher. (3) New global searchers are generated to explore new promising areas. (4) The new global searchers are evaluated. After competing with global searchers in the last iteration, the better ones are saved as centers of local exploitation areas in the next iteration. (5) If a global searcher remaining the same in β continuous iterations, it will be treated as an approximate solution. Then, SQP will be used to refine it. (6) The global searcher would be replaced by the solution obtained by the SQP if it is refined. If the global searcher cannot be refined by SQP in 3 iterations, it would be archived as a stagnant searcher. (7) If a global searcher did not change in ten continuous iterations or its fitness value meets the accuracy requirement, it would be archived as a stagnant searcher. The above iteration is repeated until the termination criterion is met. Experiments were carried out based on frequently used and new benchmark problems, and details can be found in Appendixes A and B.

Results on comparative experiments. Experiments on frequently-used NES (F01-F42) benchmark problems [7], ten complex NES benchmark problems (F43-F52) and four real-world engineering problems are carried out.

Results on F01-F42 show that SQP-MOA obtains the highest average SR value (100%) and the highest average PR value (100%); i.e., it succeeds in locating all the roots. It can be concluded that SQP-MOA is an effective method to locate multiple roots of NES problems, especially when the number of variables is high or the number of roots is high or close roots exist. A comparison between SQP-MOA and GA-SQP shows that it is reasonable to use MOA for fast finding multiple approximate solutions of NES.

Results on complex NES problems show that SQP-MOA significantly outperforms the other compared methods. It can be concluded that these new testing problems with many local traps are complex enough to challenge new methods, and SQP-MOA has a stronger ability to escape from local traps and converging to the roots of NES. But, SQP-MOA has a risk that multiple global searchers search near the same or located roots. Indeed, it is difficult to distinguish between close roots and the same root. This motivates us to develop more efficient global searcher selection methods in future work.

Results of real-world engineering design problems indicate that SQP-MOA is able to obtain more roots. The roots obtained by SQP-MOA are able to show the regularity of the distribution of roots clearly.

Results on contributions of different components. Results show that SQP-MOA outperforms SQP and MOA, It can be concluded that combining MOA and SQP can not only bring their advantages into full play but also offset their disadvantages. MOA contributes to improving the effectiveness of SQP-MOA by helping SQP-MOA jump out of local traps and focus its search on a more fruitful area. SQP contributes to improving the effectiveness of SQP-MOA by refining the solution effectively.

Results on the influence of different parameter settings. SQP-MOA contains five user-defined parameters, i.e., μ_1, N_{g}, N_{l}, α and β. The effectiveness and efficiency of SQP-MOA are insensitive to μ_1, and the value of μ_1 can be chosen from a vast range, for example, from 0.1 to 0.25.

The effectiveness of SQP-MOA is insensitive to N_{g}, and the optimal setting of N_{g} value is related to the number of roots.

The effectiveness of SQP-MOA is insensitive to N_{l}, and the value of N_{l} can be chosen from a large range, for example, from d to $2d$. Given the efficiency, the optimal N_{g} depends on the problem. This motivates us to develop adaptive parameter methods in future work.

The effectiveness of SQP-MOA is insensitive to β, and the value of β can be chosen from a large range, for example, from d to $3d$. Given the efficiency, β is suggested to set at d or $2d$.

Conclusion. This study aimed to propose a method that has a strong ability in both exploration and local exploitation when used for solving NES problems. The goal was achieved by combining MOA with SQP. The results showed that the proposed methods achieved a much higher success rate and peak ratio with fewer fitness evaluations. The Bimodal-Gauss distributed random radius also made MOA less sensitive to its main parameter. SQP-MOA is more effective and efficient than the compared methods when the number of variables and roots is large or close roots exist. But it has a risk that multiple global searchers search near the same or located roots.

Acknowledgements The work was supported in part by Outstanding Youth Science Foundation (Grant No. 61922072), National Natural Science Foundation of China (Grant Nos. 61876169, 61806179, U1304602), Program for Science and Technology Innovation Talents in Universities of Henan Province in China (Grant No. 16HASTIT033), Scientific and Technological Project of Henan Province (Grant No. 172102310334), and Open Project of Key Laboratory for the Structure and Evolution of Celestial Objects, Chinese Academy of Sciences (Grant No. OP201703).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 He W, Gong W Y, Wang L, et al. Fuzzy neighborhood-based differential evolution with orientation for nonlinear equation systems. Knowledge-Based Syst, 2019, 182: 104796

2 Deb K, Deb K. Multi-objective optimization. In: Search Methodologies. Boston: Springer, 2014. 403–449

3 Han Y, Li J Q, Gong D W, et al. Multi-objective migrating birds optimization algorithm for stochastic lot-streaming flow shop scheduling with blocking. IEEE Access, 2019, 7: 5946–5962

4 Li B L, Chen J H, Shi X L, et al. On the convergence of multivariant optimization algorithm. Appl Soft Comput, 2016, 48: 230–239

5 Wang Y, Li B, Yuan B. Hybrid of comprehensive learning particle swarm optimization and SQP algorithm for large scale economic load dispatch optimization of power system. Sci China Inf Sci, 2010, 53: 1566–1573

6 Li B L, Liang J, Yue C T, et al. Multivariant optimization algorithm with bimodal-gauss. In: Proceedings of Simulated Evolution and Learning. Cham: Springer, 2017. 920–928

7 Liao Z W, Gong W Y, Yan X S, et al. Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms. IEEE Trans Syst Man Cybern Syst, 2020, 50: 1590–1601