Spread metric: To analyze whether the points generated by the algorithm are well distributed in the approximate Pareto front of a given problem, two divergence metrics are used(Γ and $\Delta_{p,s}$).

$$\Gamma_{p,s} = \max_{j \in M} \max_{i \in \{0,1,\ldots,N\}} \delta_{i,j}$$

$$\Delta_{p,s} = \max_{j \in M} \left(\frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N-1} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right)$$

By definition, the smaller the value of the two distributions, the more uniform the Pareto algorithm solves the corresponding problem.

4.2 Numerical experiments

The MOSQP algorithm, MOSQP algorithm proposed by Fliege and Vaz in literature [1], and NSGAII algorithm are used for numerical comparison experiments in Python to solve the problem Problem1 ZDT1 ZDT2 MOP3[5-9]. respectively, and the purity index, divergence index and function calculation frequency index are calculated through the algorithm performance measurement formula in Section 4.1. Analysis the quantity and quality of the Pareto, and then compare the performance curve and analyze the advantages and disadvantages of the algorithm. To distinguish between them, we note that the MOSQP algorithm proposed by Fliege and Vaz is MOSQP (F). The numerical experimental parameters of the MOSQP algorithm are configured as follows:

$$k = 0.5\,,\quad b = 4\,,\quad \sigma = 0.2\,,\quad M = 2\,,\quad A = 0.5\,,\quad K = 5$$

Where the numerical experimental parameters of the MOSQP (F) algorithm are configured as:

$$\sigma = 0.2\,,\quad K = 5$$

K is the times of Spread, For the MOSQP algorithm and the MOSQP (F) algorithm for the same initial iteration point set diffusion 5 times. The gradient calculations involved in the algorithm were all solved using the jacobian function in the automated differential library autograd. NSGAII The number of populations is set to the maximum of the number of Pareto fronts acquired by the MOSQP algorithm and the MOSQP (F) algorithm, and the number of iterations is set to 300.

First, we give the following dual-objective constraint optimization problem

(Problem1):

$$\min \; F(x) = [f_1(x) = (1-x_1)^2 + 100(x_2 - x_1^2)^2 \,,$$
$$f_2(x) = (x_1 + x_2 - 1)^2 + 10(x_1 - x_2)^2]^T$$
$$s.t. \quad x_1^2 + x_2^2 \le 0.5$$

Using Monte Carlo simulations:

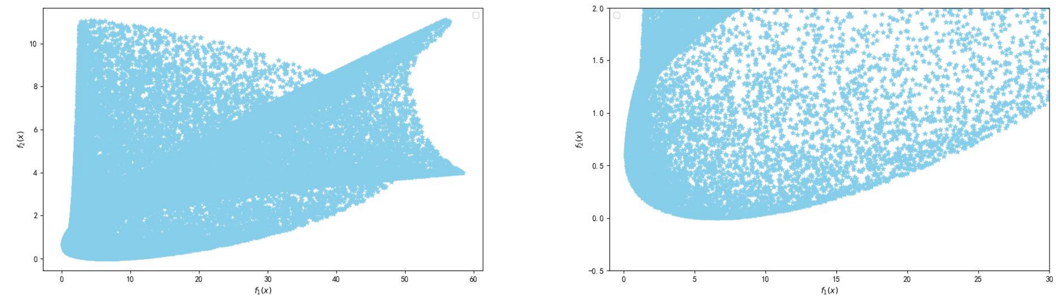
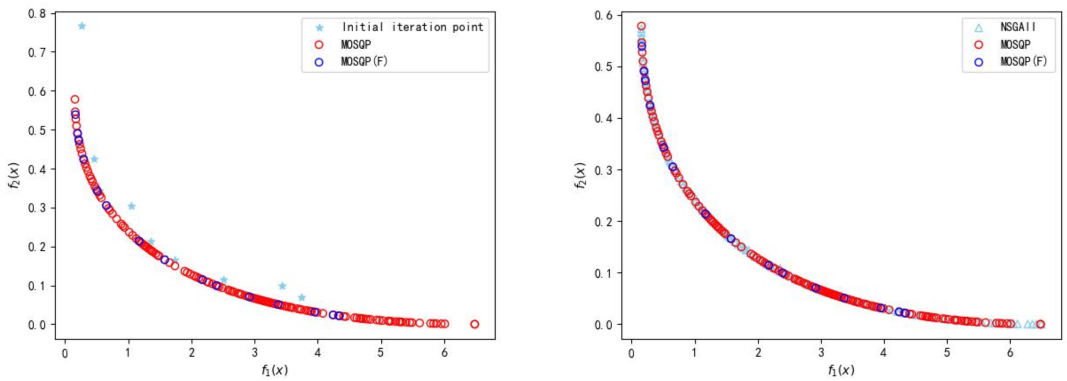


Figure 4,2.1 Functional images generated by the Monte Carlo method

The left side simulates the function image within the entire feasible domain, while the right side simulates the intercepted local features near the Pareto front.

MOSQP, MOSQP (F) and NSGAII algorithms are used to solve the above dual objective constraint optimization problem, and show the solved Pareto frontier:



(a) Algorithm iteration process (b) MOSQP (F) and NSGAII algorithm frontier comparison

Figure 4.2.2. MOSQP algorithm, NSGAII obtained problem (Problem1)

From figure4.2..2, we can see that the MOSQP algorithm proposed in this paper can effectively solve the nonlinear constraint multi-objective optimization problem, and improve the MOSQP algorithm of diffusion step Pareto front in quantity and quality are better than MOSQP (F), we calculated the MOSQP, MOSQP (F) and NSGAII algorithm (300) approximate Pareto front corresponding to the purity metric $\overline{t}_{p,s}$, Spread metric $\Gamma_{p,s}$ $\Delta_{p,s}$.

Table 4.2.1 Solving the corresponding values of each performance index of Problem1

	$\overline{t}_{p,s}$	$\Gamma_{p,s}$	$\Delta_{p,s}$
MOSQP	1.7808	0.4737	0.8965
MOSQP(F)	8.6667	0.5999	0.5438
NSGAII	3.0952	0.2111	0.5981

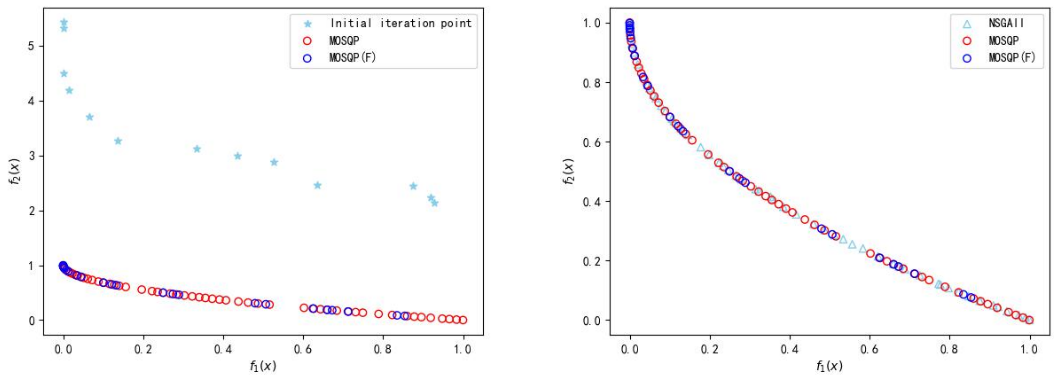
From the above indicators, it can be analyzed that based on the same initial iteration point set, the MOSQP algorithm proposed in this paper has more number of non-dominant solutions for the approximate Pareto front obtained by Problem1, with higher quality, which is better than MOSQP (F) and NSGAII algorithm, and the distribution is relatively good. Compared to MOSQP (F), the number of gradient calculation increases, but avoid the calculation of Hessian matrix, the PE index will be smaller.

Then we analyze the classical two-objective boundary constraint optimization problem ZDT 1 (n=30), with 30 decision variables in the problem, which increases the difficulty of the problem, which is defined as follows:

$$(\text{ZDT1})\quad \min\ F(x)=[f_1(x)=x_1, f_2(x)=1-\sqrt{\frac{f_1(x)}{g(x)}}]^T$$

$$g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$$

$$s.t.\quad 0\le x_i\le 1, i=1,\cdots,n$$



(a) Algorithm iteration process (b) MOSQP (F) and NSGAII algorithm frontier comparison

Figure 4.2.3 MOSQP algorithm, NSGAII obtained problem (ZDT 1)

From Figure 4.2.3, we can see that the MOSQP algorithm proposed here can effectively solve the high-dimensional boundary constraint multi-objective optimization problem ZDT 1, and improve the MOSQP algorithm of diffusion step Pareto front in