LOBACHEVSKY UNIVERSITY

Parallel Multi-objective Optimization Method for Finding Complete Set of Weakly Efficient Solutions

Vladislav Sovrasov

Lobachevsky University

19 October 2017
Ekaterinburg

Problem statement

$$\min\{f(y) : y \in D\}, D = \{y \in \mathbb{R}^n : a_i \leqslant y_i \leqslant b_i, 1 \leqslant i \leqslant n\},$$

where $f(y)$ is a vector-function.

Solution of the problem is a set of non-dominated points (Slater set):

$$S(D) = \{y \in D : \nexists z \in D, f_i(z) < f_i(y), 1 \leqslant i \leqslant m\}$$

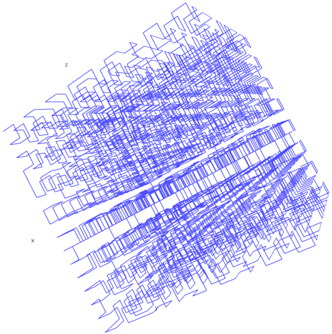
Assume objectives to satisfy Lipschitz condition in D:

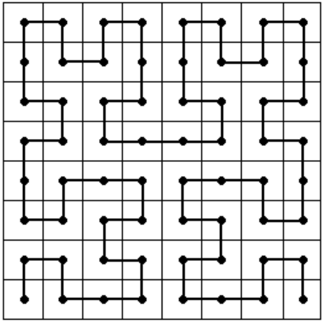
$$|f_i(y_1) - f_i(y_2)| \leqslant L_i \|y_1 - y_2\|, y_1, y_2 \in D, 0 < L_i < \infty, 1 \leqslant i \leqslant m$$

Dimension reduction

Peano-type curve $y(x)$ allows to reduce dimension of the original multi-objective problem:

$$\{y \in \mathbb{R}^N : -2^{-1} \leqslant y_i \leqslant 2^{-1}, 1 \leqslant i \leqslant N\} = \{y(x) : 0 \leqslant x \leqslant 1\}$$
$$\min\{f(y) : y \in D\} = \min\{f(y(x)) : x \in [0,1]\}$$





Scalarization tecnique

Multi-objective problem can be reduced to a scalar optimization problem using the following function:

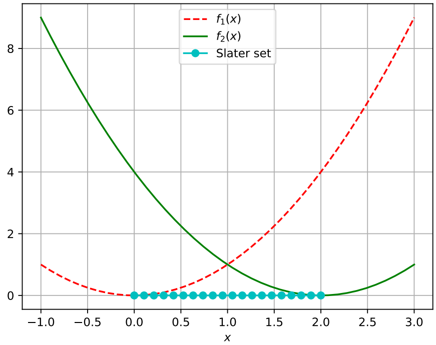
$$h(x,y) = \min\{f_i(x) - f_i(y) : 1 \leqslant i \leqslant m\},$$
$$\varphi(x) = \max\{h(x,y) : y \in [0,1]\}, x \in [0,1].$$

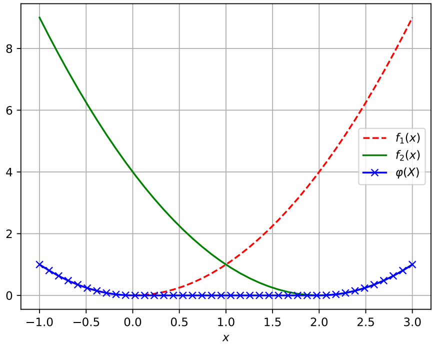
If x^* is weakly effective point, then $\varphi(x^*) \leq 0$, in the opposite case $\varphi(x^*) > 0$, so the equivalent scalar problem is:

$$\varphi^* = \min\{\varphi(x) : x \in [0,1]\}.$$

Scalarization tecnique

One-dimensional example:





Optimization method

Optimization method generates searh sequence $\{x_k\}$ and consists of the following steps:

Step 1. Sort the searh information (one-dimentional points) in increasing order.

Step 2. Compute an approximation of the function $\varphi(x)$.

Step 3. For each interval (x_{i-1}, x_i) compute quantity $R(i)$, called characteristic.

Step 4. Choose an interval (x_{t-1}, x_t) with the greatest characteristic and compute objective $f(y(x))$ in the point choosen using the descision rule d:

$$x^{k+1} = d(t) \in (x_{t-1}, x_t)$$

Step 4. If $x_t - x_{t-1} < \varepsilon$ stop the method.

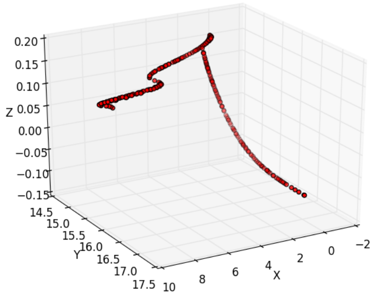
Improved parallel optimization method

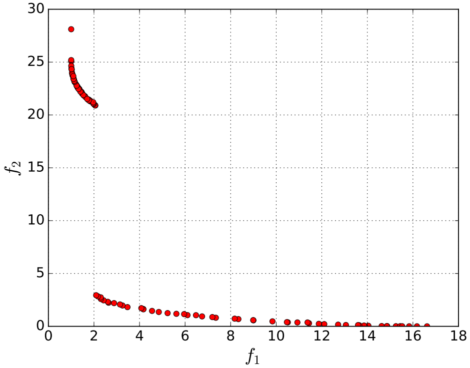
- *Local refinement.* Every q iterations ignore the characteristics $R(i)$ and perform calculation of the objective at the minimum point of the $\varphi(x)$ approximation.
- *Parallelization by characteristics.* At the Step 4 choose p best intervals and generate p new points using rule $d(i)$. Then compute the objective $f(y(x))$ at this point in parallel exploiting p computation units.

  In the best case (method doesn't generate redundand points compared to sequental one and wastes time only on computation of the objective) this scheme can give speedup in p times.

Results

Examples of numerical solutions.





Results

Number of iterations decreases when increasing number of threads p.

Problem	p				
	1	2	4	8	16
Markin-Strongin	1041(198)	516(198)	256(185)	131(197)	68(191)
Fonseca and Fleming 2d	1181(93)	636(99)	386(111)	176(95)	106(97)
Fonseca and Fleming 3d	5346(160)	3551(183)	1186(143)	606(153)	351(142)
Viennet problem	4896(276)	2156(273)	1226(270)	631(287)	286(274)
Poloni's function	3351(102)	1706(90)	856(88)	426(96)	201(99)

Results (speedup in iterations)

Problem	p			
	2	4	8	16
Markin-Strongin	2.02	4.07	7.95	15.31
Fonseca and Fleming 2d	1.86	3.06	6.71	11.14
Fonseca and Fleming 3d	1.51	4.51	8.82	15.23
Viennet problem	2.27	3.99	7.76	17.12
Poloni's problem	1.96	3.91	7.87	16.67

Results (speedup in time)

Problem	p				
	1(time, s)	2	4	8	16
Markin-Strongin	104.47	1.97	3.65	6.79	9.90
Fonseca and Fleming 2d	118.95	1.85	2.81	5.79	6.40
Fonseca and Fleming 3d	554.45	1.51	4.14	8.05	10.69
Viennet problem	1488.6	2.22	3.64	6.98	13.49
Poloni's problem	336.74	1.82	3.64	6.98	10.60

Q&A

Vladislav Sovrasov
sovrasov.vlad@gmail.com
https://github.com/sovrasov