An Scalable Algorithm for Solving Non-stationary Linear Programming Problems

Irina M. Sokolinskaya
South Ural State University (national research university)
School of Electrical Engineering and Computer Science

The large-scale linear programming (LP) problems

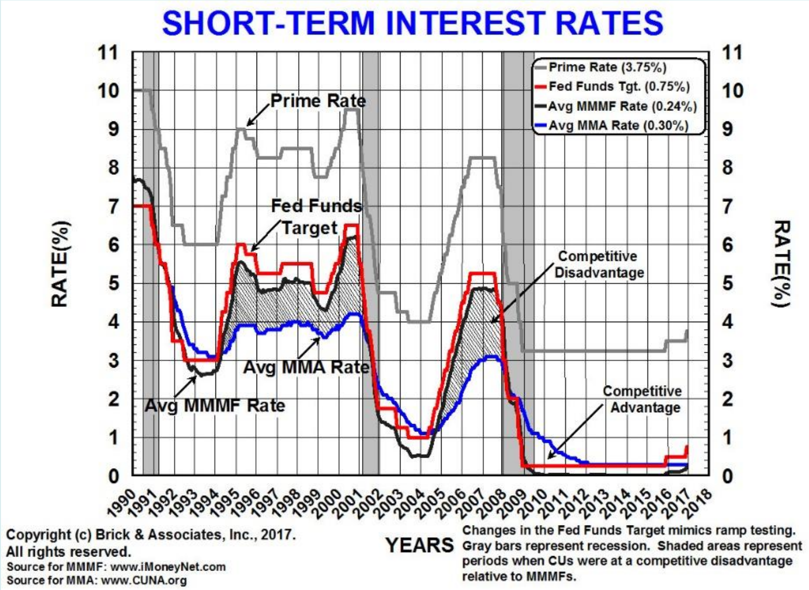
$$\max\left\{\left\langle c,x\right\rangle \mid Ax\le b, x\ge 0\right\}$$

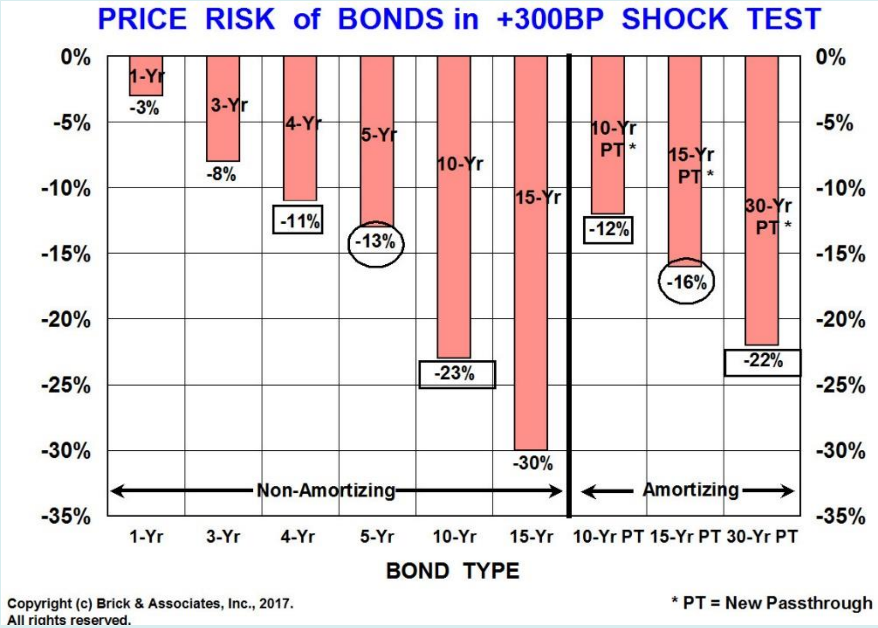
- Schedule crews for 3 400 daily flights in 40 countries
- Buy ads in 10-15 local publications across 40 000 zip codes
- Pick one of 742 trillion choices in creating the US National Football League schedule
- Select 5 offers out of 1 000 for each of 25 000 000 customers of online store
- Place 1 000s of stock keeping units on dozens of shelves in 2 000 stores
- Decide among 200 000 000 maintenance routing options

Chung W. Applying large-scale linear programming in business analytics // 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2015. C. 1860-1864.

Asset-liability management

- Dynamic LP task
- 1.7 billion constraints
- 5.1 billion variables

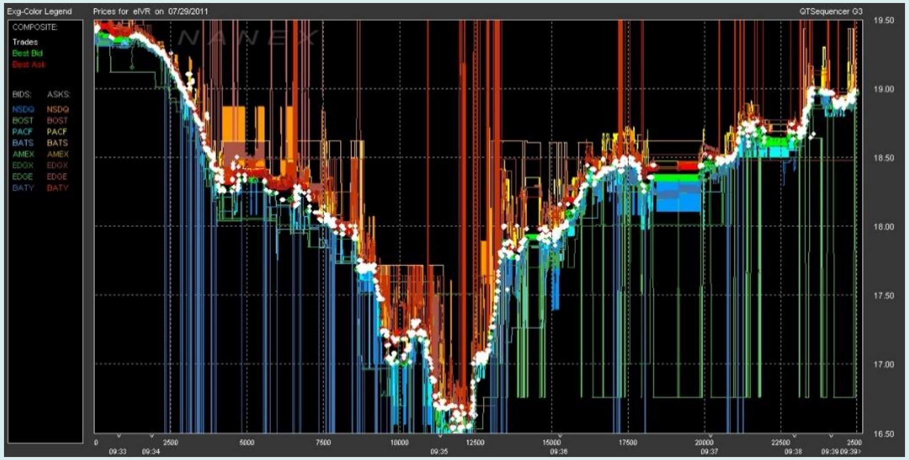


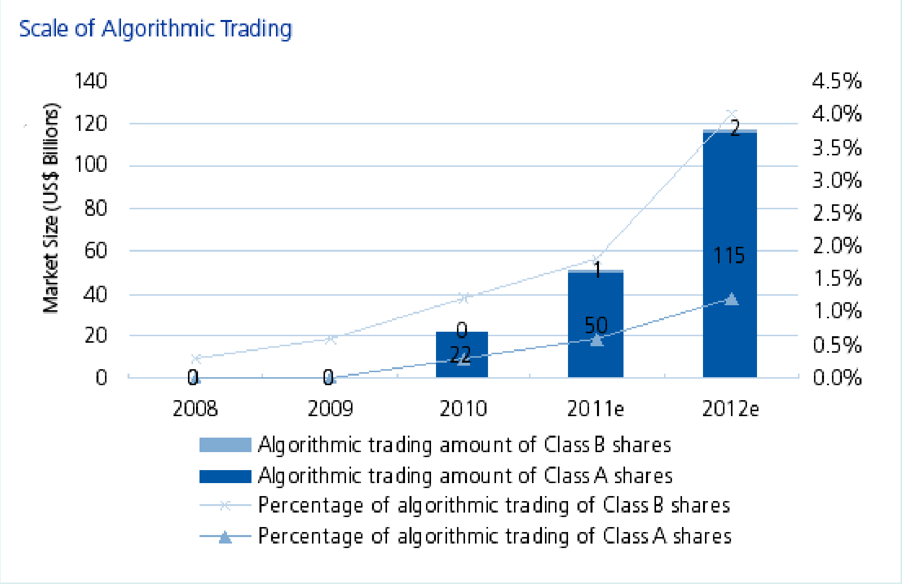


Sodhi M.S. LP modeling for asset-liability management: A survey of choices and simplifications // Operations Research. 2005. V. 53. No. 2. P. 181-196.

Algorithmic trading

- The number of variables: 10^5-10^6
- The number of inequalities: 10^6-10^7
- The period of input data change: 10^{-2}-10^{-3} sec.





Non-Stationary Linear Programming Problem

$$\max\left\{\left\langle c_t, x\right\rangle \mid A_t x \le b_t, x \ge 0\right\}$$

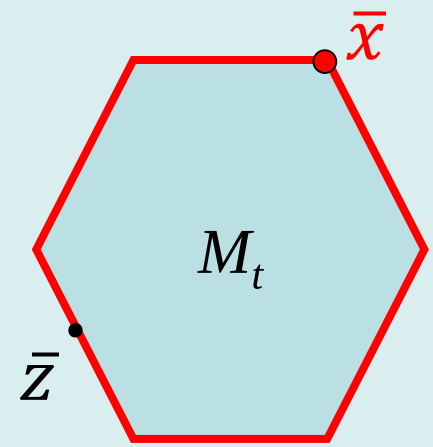
- $x \in \mathbb{R}_n$
- A_t – the matrix $m \times n$
- c_t, b_t – the vectors in the vector space $\mathbb{R}_n$
- $t \in \mathbb{R}_{\ge 0}$ – the time

The idea of the algorithm NSLP (Non Stationary Linear Programming)

Two phases of the algorithm :

- *Quest* – calculates a solution $\bar{z} \in M_t$

- *Targeting* – moves point $\bar{z}$ in such a way that the solution of the LP problem $\bar{x}$ permanently was in its ε- vicinity

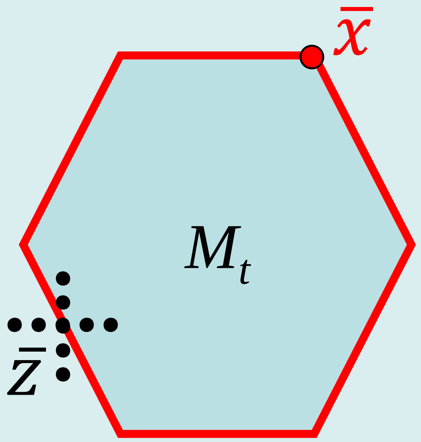


$$A_t x \le b_t \iff x \in M_t$$

The idea of the algorithm NSLP (Non Stationary Linear Programming)

Two phases of the algorithm :

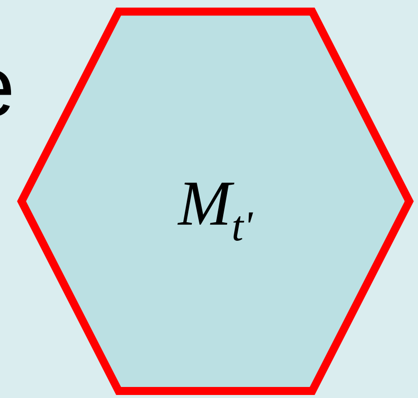
- *Quest* – calculates a solution $\bar{z} \in M_t$
- *Targeting* – moves point $\bar{z}$ in such a way that the solution of the LP problem $\bar{x}$ permanently was in its ε- vicinity



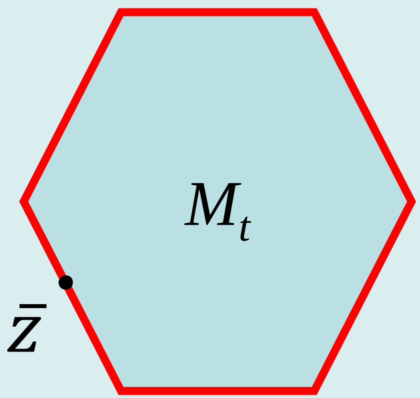
$$A_t x \le b_t \iff x \in M_t$$

The *Quest* phase
(search for $\bar{z} \in M_t$)

We can not simply solve the system of equations $A_t x = b_t$, since while we solve it, the polytope M_t will change the shape and position in space.



$$A_{t'} x \le b_{t'} \Leftrightarrow x \in M_{t'}$$



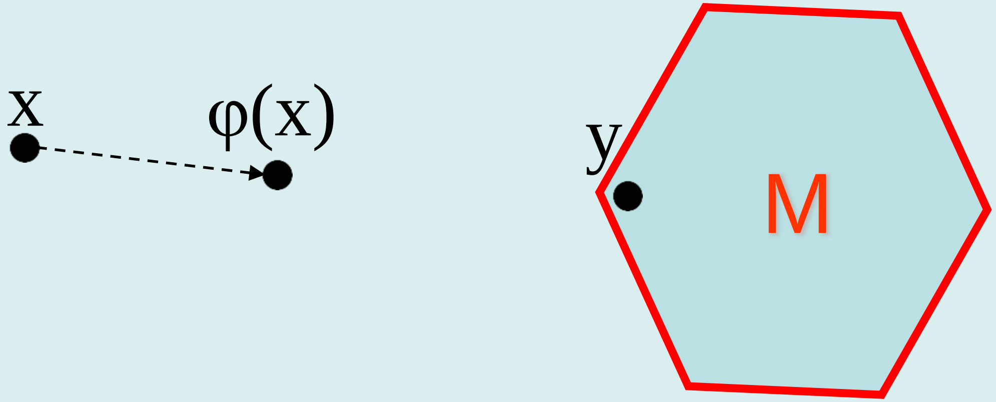
$$A_t x \le b_t \Leftrightarrow x \in M_t$$

The fejerian maps

M – convex bounded set

A single-valued map $\varphi: \mathbb{R}^n \to \mathbb{R}^n$ is called *M*-fejerian if

$$\varphi(y) = y, \forall y \in M;$$
$$\|\varphi(x) - y\| < \|x - y\|, \forall x \notin M, \forall y \in M$$



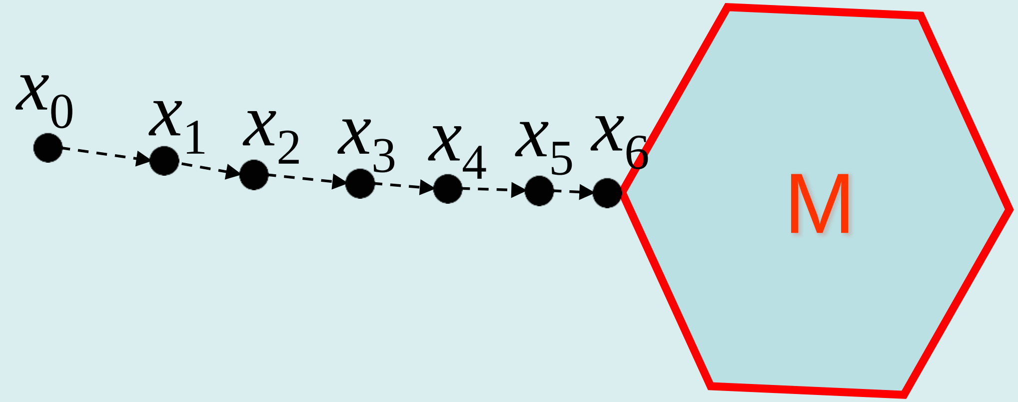
Lipót Fejér
(1880 – 1959)
Hungarian mathematician

Fejer process

$$\varphi^{s}(x) = \underbrace{\varphi \ldots \varphi(x)}_{s}$$

$$x_0 \in \mathbb{R}^n$$

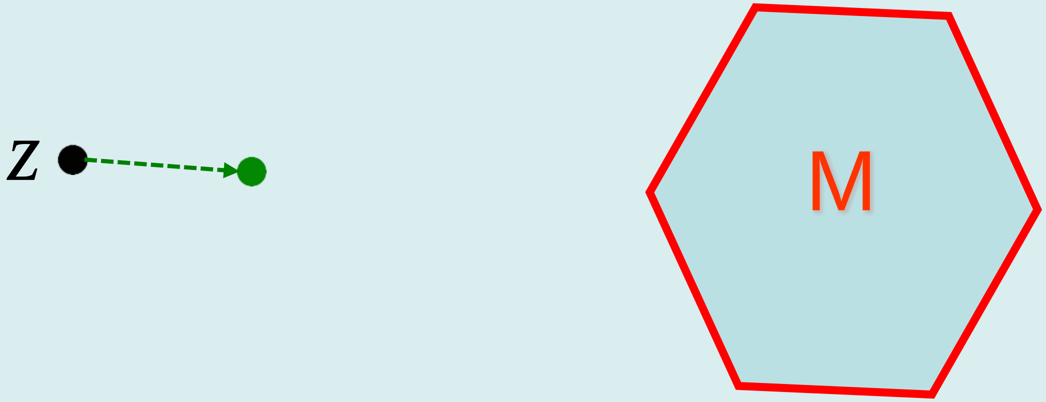
$$\{\varphi^{s}(x_0)\}_{s=0}^{+\infty}$$



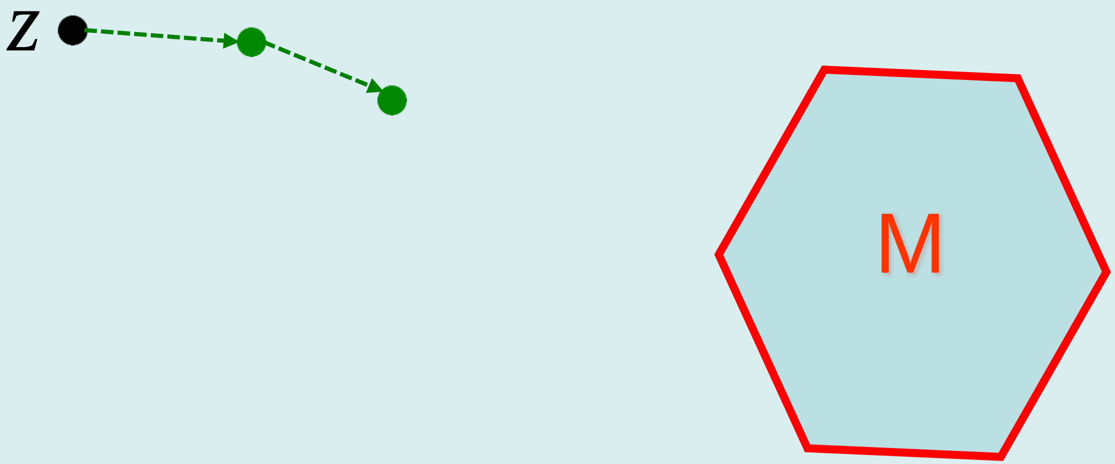
$$x_i = \varphi^{i}(x_0)$$

Continuous single-valued M-fejerian map converges to a point belonging to the polytope M (M - convex bounded set)

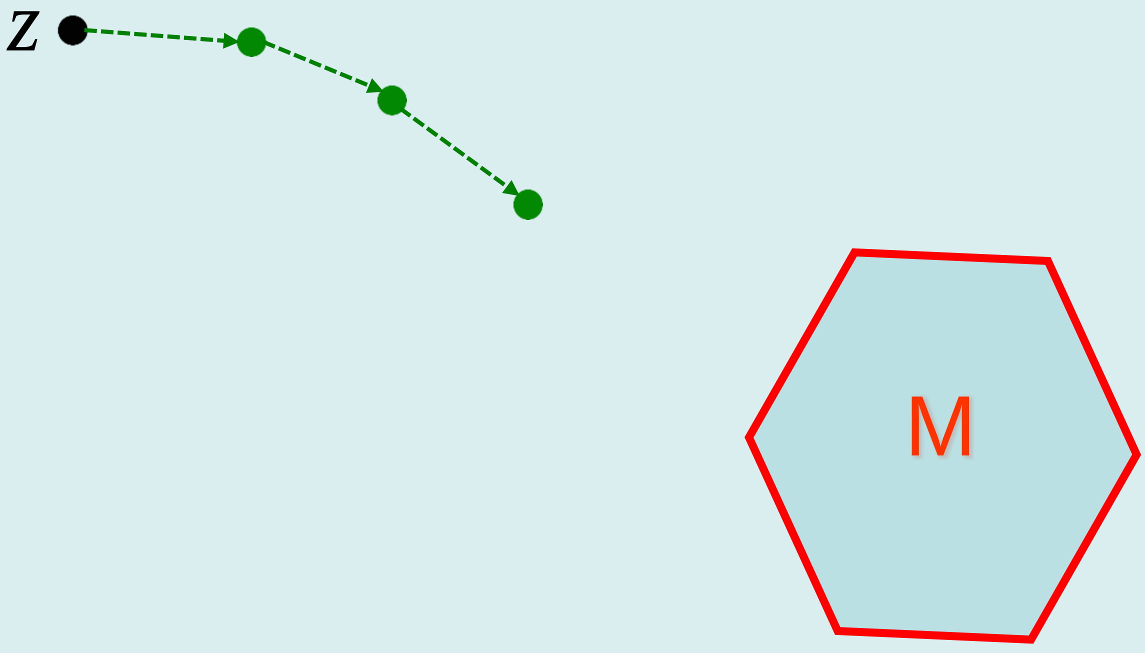
«Self-guided» Fejer process



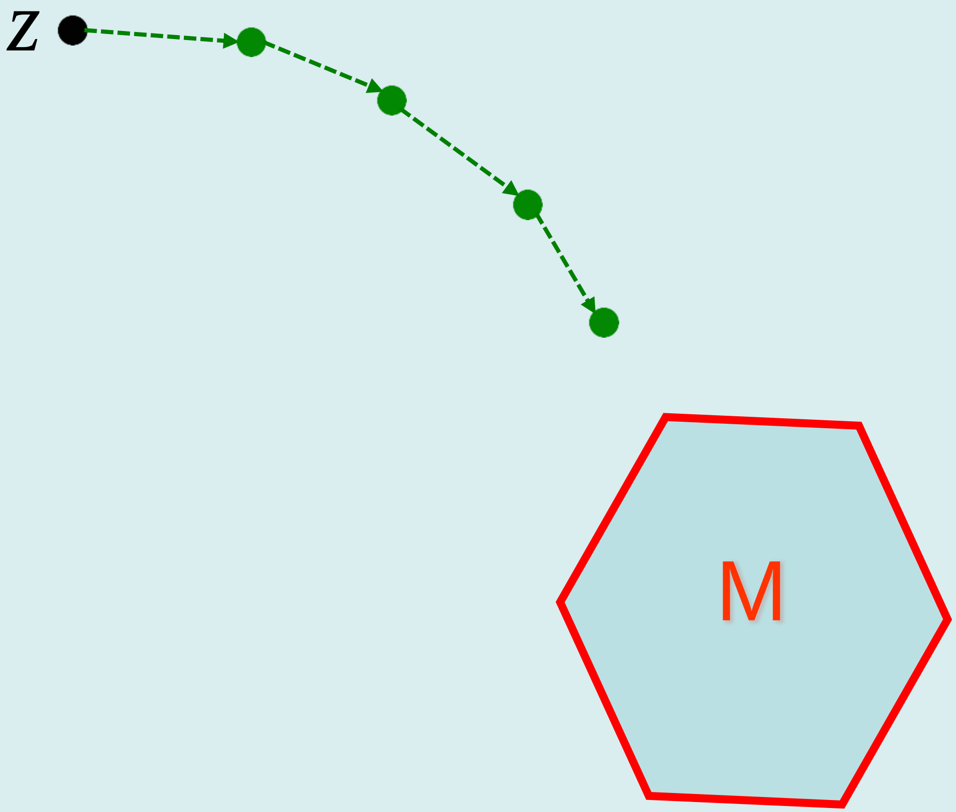
«Self-guided» Fejer process



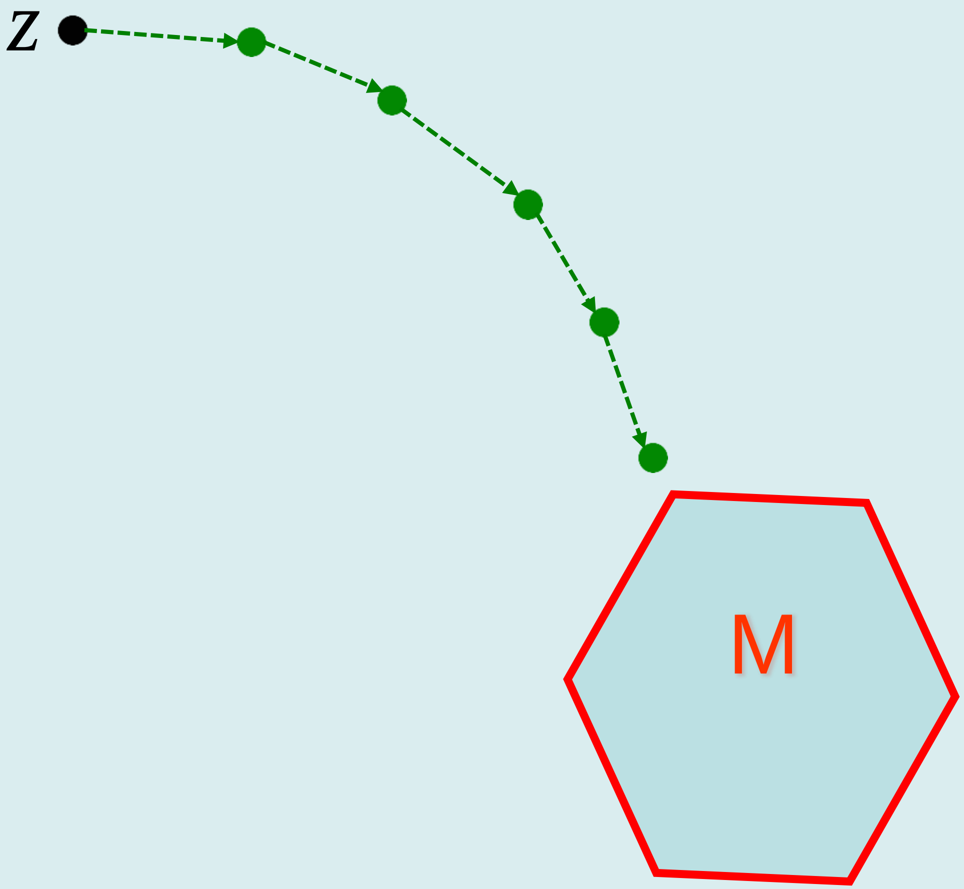
«Self-guided» Fejer process



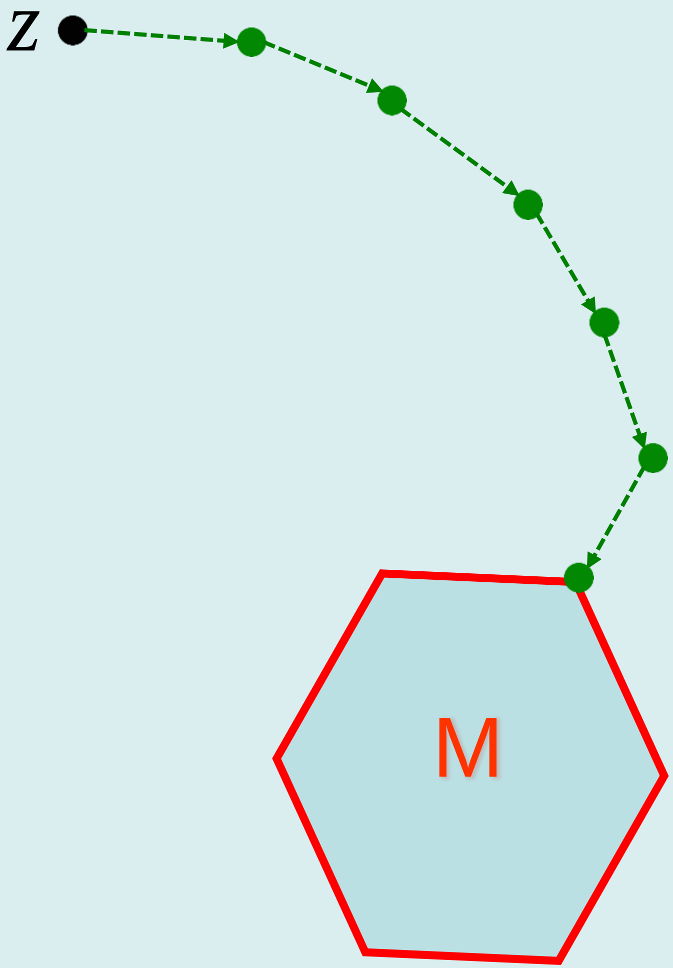
«Self-guided» Fejer process



«Self-guided» Fejer process



«Self-guided» Fejer process



Fejer map for the Quest phase

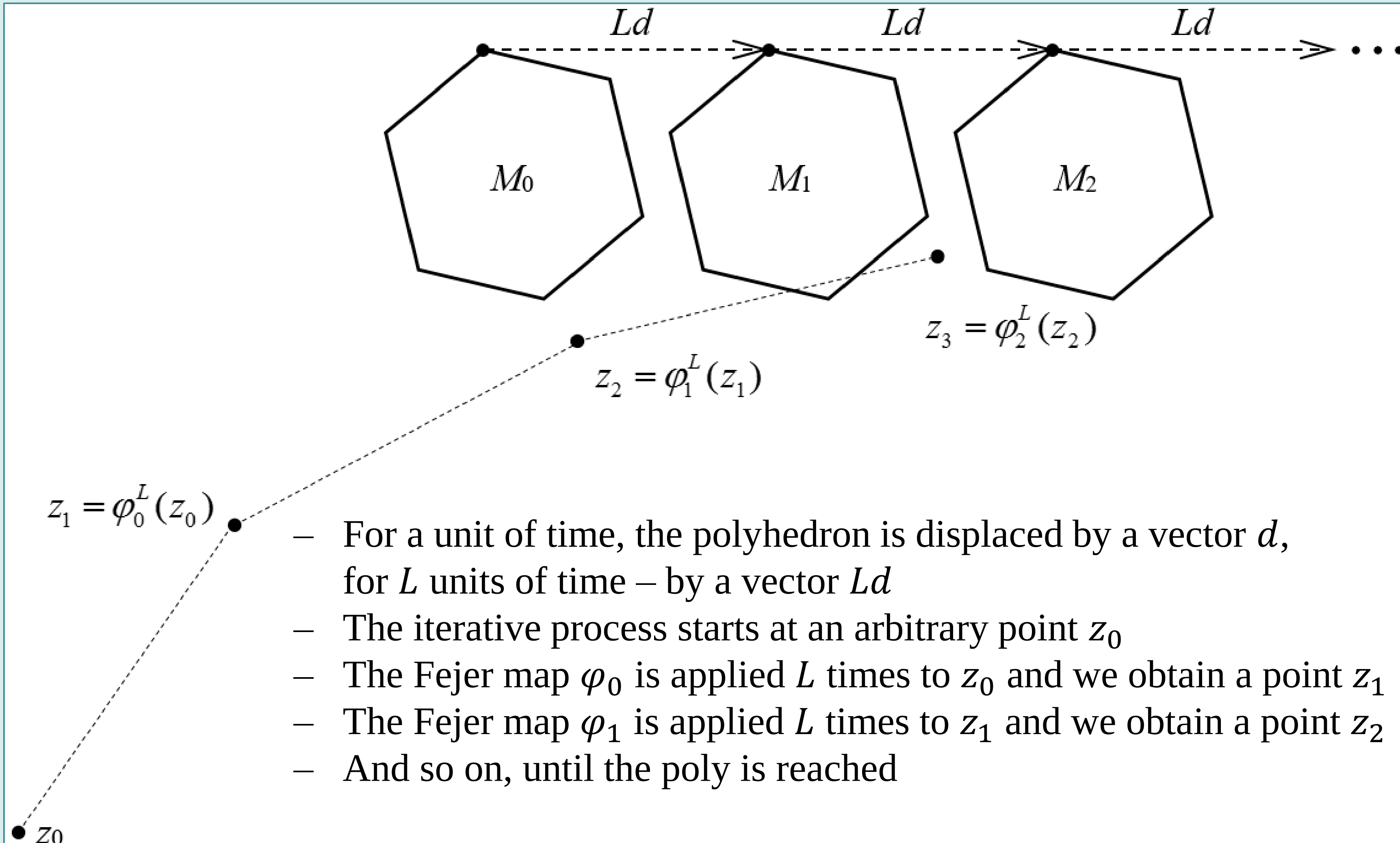
$$\varphi_t(x) = x - \frac{1}{m} \sum_{i=1}^{m} \frac{\max\{\langle a_{ti}, x \rangle - b_{ti}, 0\}}{\|a_{ti}\|^2} \cdot a_{ti}$$

- a_{ti} – i-th row the matrix A_t
- $b_{t1}, \dots, b_{tm}$ – elements of the column b_t
- m – number of rows of the matrix A_t
- t – the time

Organization of the Quest phase

- Every L iterations, the input data update is performed
- Let $t_0 = 0, t_1 = L, t_2 = 2L, \dots, t_k = kL, \dots$
- Let the polytope M_t take shapes and locations $M_0, M_1, \dots, M_k, \dots$ at this points of time
- Let $\varphi_0, \varphi_1, \dots, \varphi_k, \dots$ be Fejer maps at this points of time
- In the Quest phase, the iterative process calculates the following sequence of points: $\left\{z_1 = \varphi_0^L(z_0), z_2 = \varphi_1^L(z_1), \dots, z_k = \varphi_{k-1}^L(z_{k-1}), \dots\right\}$

Example «Translation»



- For a unit of time, the polyhedron is displaced by a vector d, for L units of time – by a vector Ld
- The iterative process starts at an arbitrary point z_0
- The Fejer map φ_0 is applied L times to z_0 and we obtain a point z_1
- The Fejer map φ_1 is applied L times to z_1 and we obtain a point z_2
- And so on, until the poly is reached

Mathematical model for Translation

$$\max\{\langle c, x\rangle | A(x - td) \leq b, x \geq 0\}$$

$$\varphi_k(x) = x - \frac{\lambda}{m}\sum_{i=1}^{m}\frac{\max\{\langle a_i, x - kLd\rangle - b_i, 0\}}{\|a_i\|^2} \cdot a_i$$

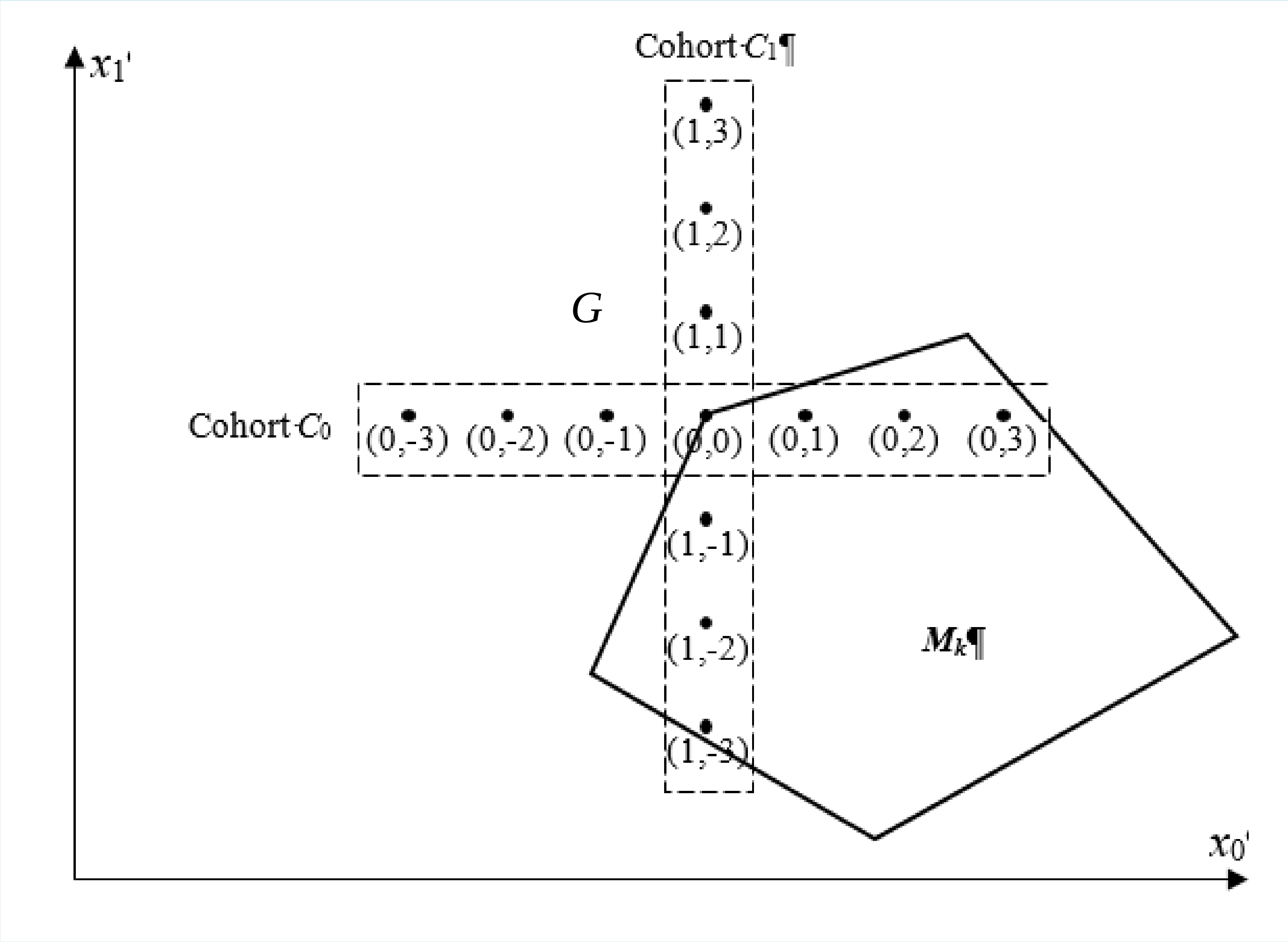
$$z_k = \varphi_{k-1}^L(z_{k-1})$$

Convergence theorem.

If $\forall x \in \mathbb{R}^n \backslash M\left(\|Ld\| < \mathrm{dist}(x, M) - \mathrm{dist}(\varphi^L(x), M)\right)$, then $\lim\limits_{k\to\infty} \mathrm{dist}(z_k, M_k) = 0$.

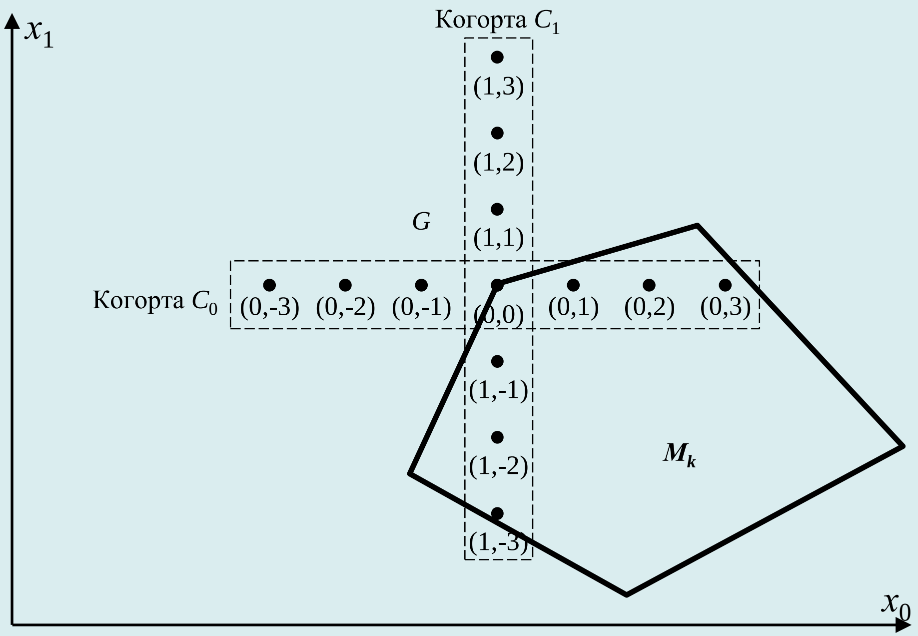
$\mathrm{dist}(z, M) = \inf\{\|z - x\|:\ x \in M\}$

The *Targeting* phase



Targeting phase includes the following steps

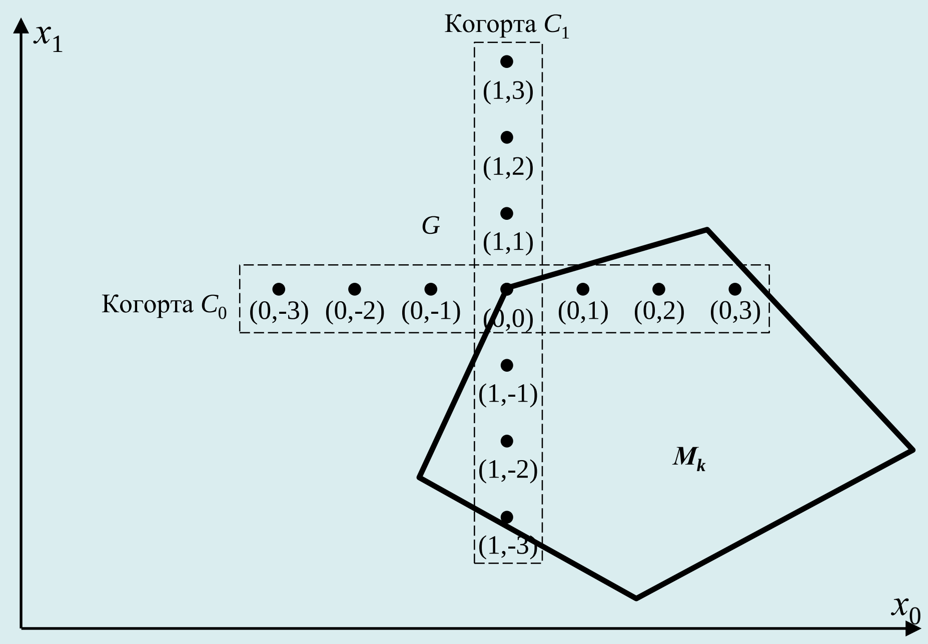
1. Build the *n*-dimensional axisymmetric cross G which has the center at point $g_0 = z_k$, where z_k is obtained as a result of the Quest phase
2. Calculate $G' = G \cap M_k$
3. Calculate $C'_i = C_i \cap G'$ for $i = 0, \ldots, n-1$
4. Calculate $Q = \bigcup_{\chi=0}^{n-1} \left\{ \mathrm{argmax}\{\langle c_k, g \rangle | g \in C'_\chi, C'_\chi \neq \emptyset\} \right\}$
5. If $g_0 \in M_k$ и $\langle c_k, g_0 \rangle \geq \max_{q \in Q} \langle c_k, q \rangle$ then $k := k + 1$ and go to the step 2
6. $g_0 := \frac{\sum_{q \in Q} q}{|Q|}$
7. $k := k + 1$ and go to the step 2



Parallelization of the step 2

$$g \in M_k \Leftrightarrow A_k g \le b_k$$

g – points of the cross G



Further research

- Parallel implementation of the NSLP algorithm in C++ language using MPI library
- Computational experiments on a cluster computing system using synthetic and real LP problems

Thank you for attention!