Using Intel Xeon Phi for Solving Non-Stationary Linear Programming Problems

Leonid Sokolinsky, Irina Sokolinskaya

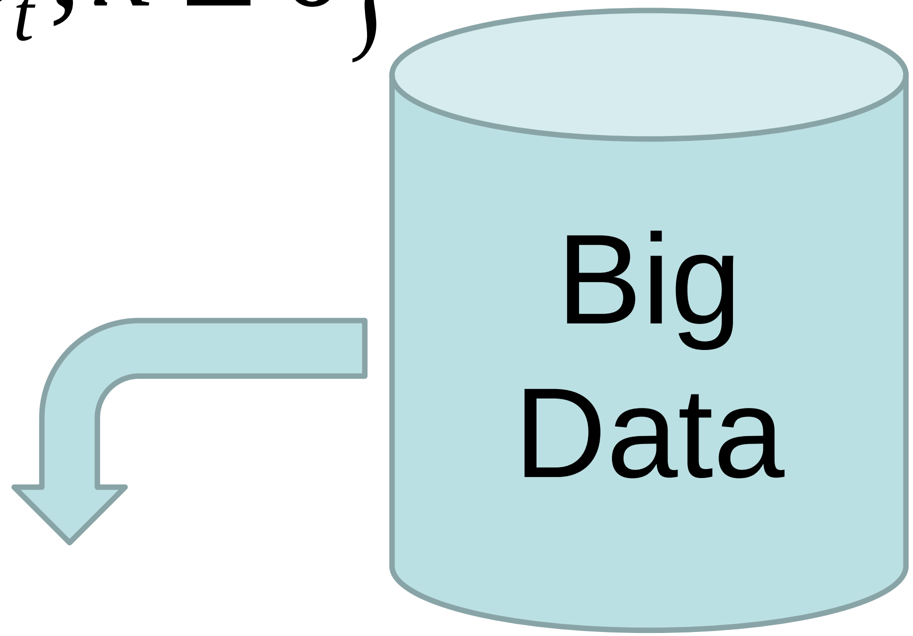
South Ural State University (national research university)

The reported study has been partially supported by the RFBR according to research project No. 17-07-00352-a, by the Government of the Russian Federation according to Act 211 (contract No. 02.A03.21.0011) and by the Ministry of Education and Science of the Russian Federation (government order 1.9624.2017/7.8).

Non-stationary large-scale 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 – matrix $m \times n$
- b_t – m-dimensional vectors
- c_t – n-dimensional vector
- $t \in \mathbb{R}_{\geq 0}$ – time



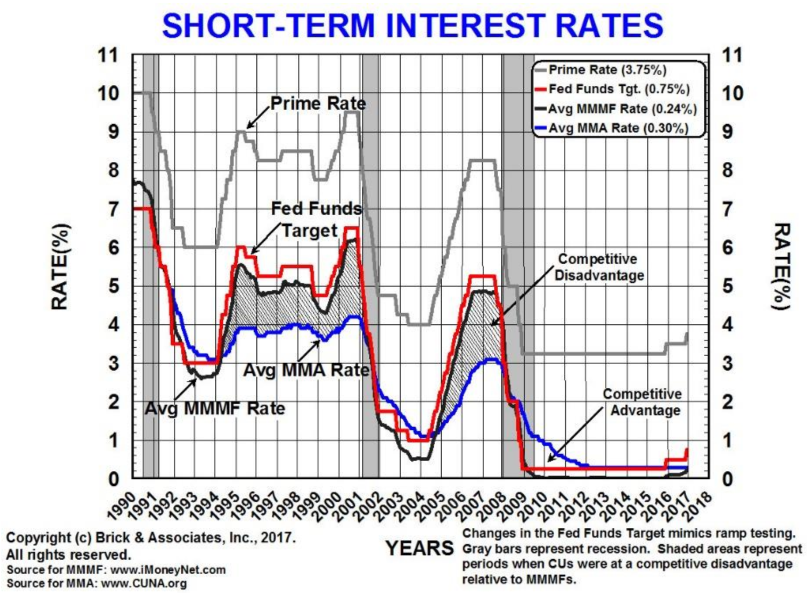
$n, m > 10^6$

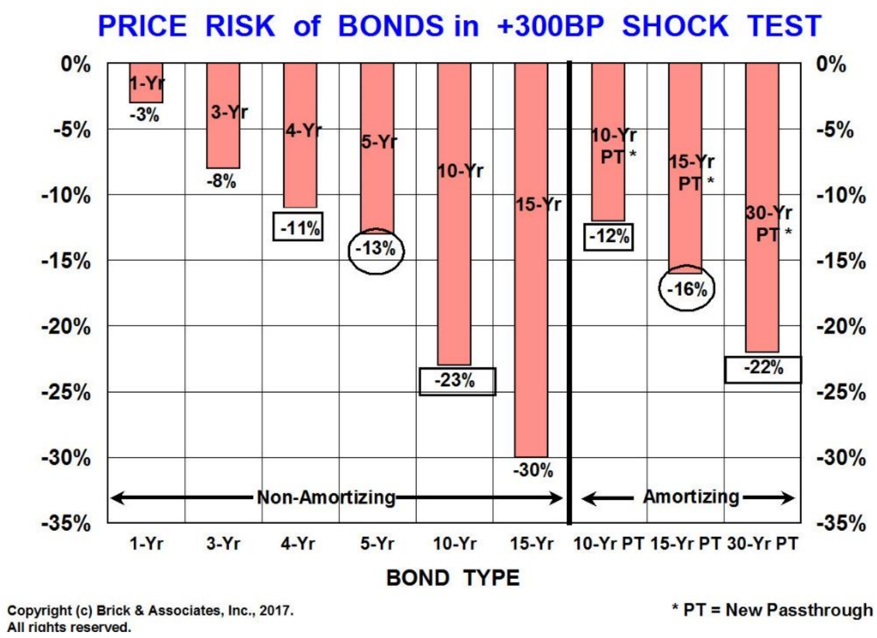
$Period\ of\ input\ data\ change < 10^{-2}$ sec.

Asset-Liability Management

Dynamic linear programming problem

- 1.7 billion inequalities
- 5.1 billion variables

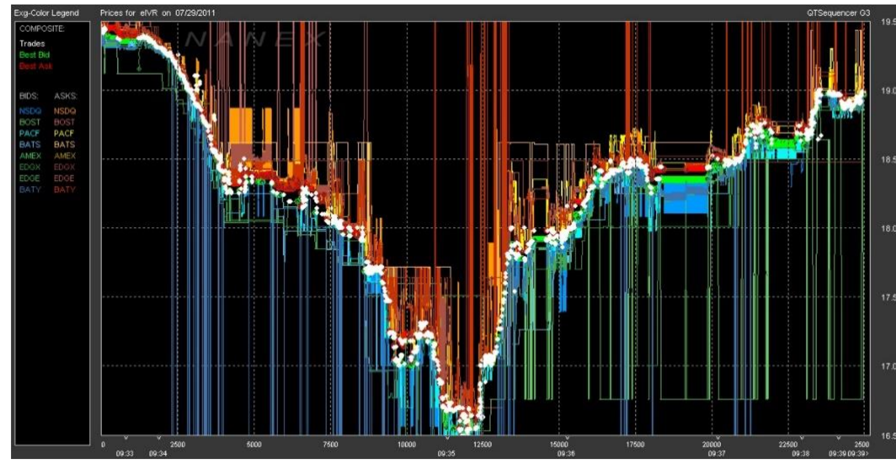


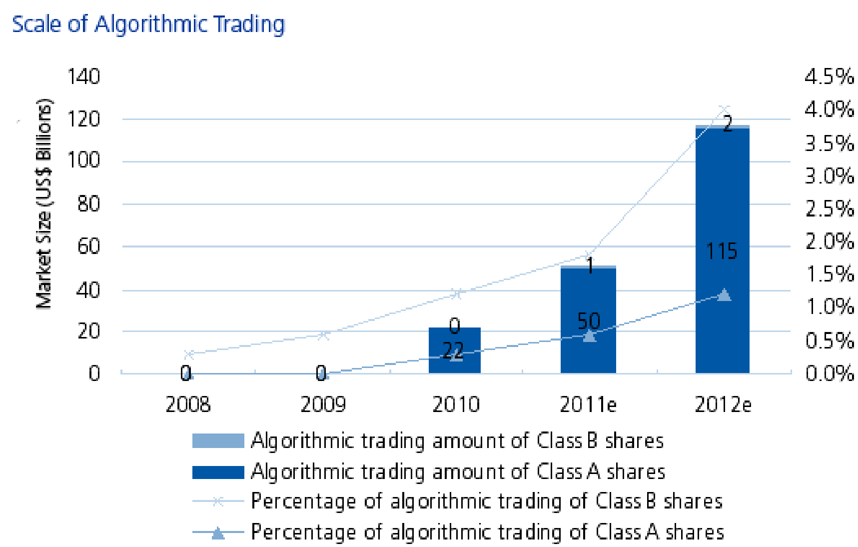


Sodhi M.S. LP modeling for asset-liability management: A survey of choices and simplifications // Operations Research. 2005. V. 53. No. 2. P. 181-196.

High Frequency Trading

- Dimension: 10^5-10^6
- Number of inequalities: 10^6-10^7
- Period of input data change: 10^{-2} - 10^{-3} sec.

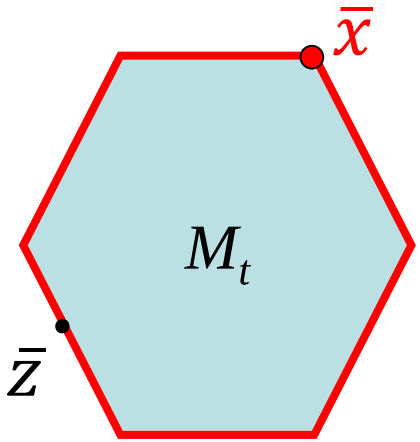




NSLP Algorithm (Non Stationary Linear Programming)

Algorithm phases:

- *Quest* – find point $\bar{z} \in M_t$
- *Targeting* – moving point $\bar{z}$ in such a way that the solution $\bar{x}$ of LP problem remains permanently in an ε-vicinity of $\bar{z}$

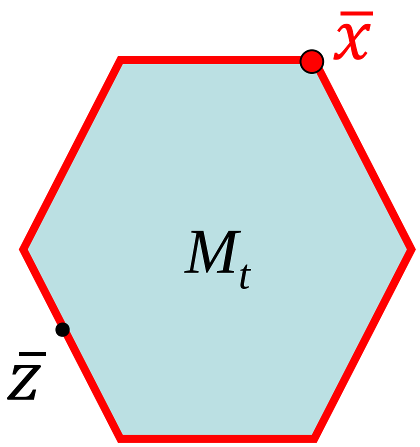


$$A_t x \le b_t \iff x \in M_t$$

NSLP Algorithm
(Non Stationary Linear Programming)

Algorithm phases:

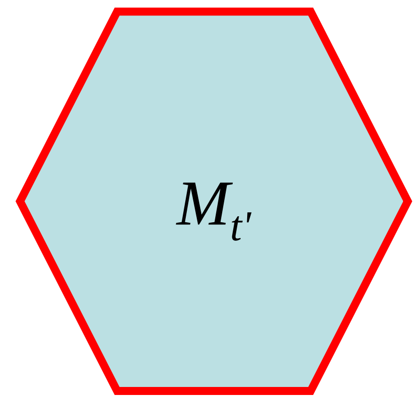
- *Quest* – find point $\bar{z} \in M_t$
- *Targeting* – moving point $\bar{z}$ in such a way that the solution $\bar{x}$ of LP problem remains permanently in an *ε*-vicinity of $\bar{z}$



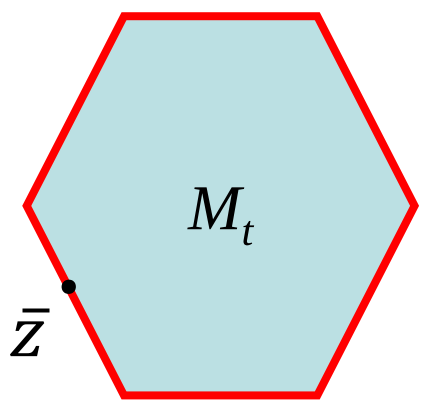
$$A_t x \le b_t \;\Leftrightarrow\; x \in M_t$$

Quest Phase (Finding $\bar{z} \in M_t$)

Conventional methods for solving a system of linear equalities can not give a solution to the problem $A_t x = b_t$ because of its non-stationarity.



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



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

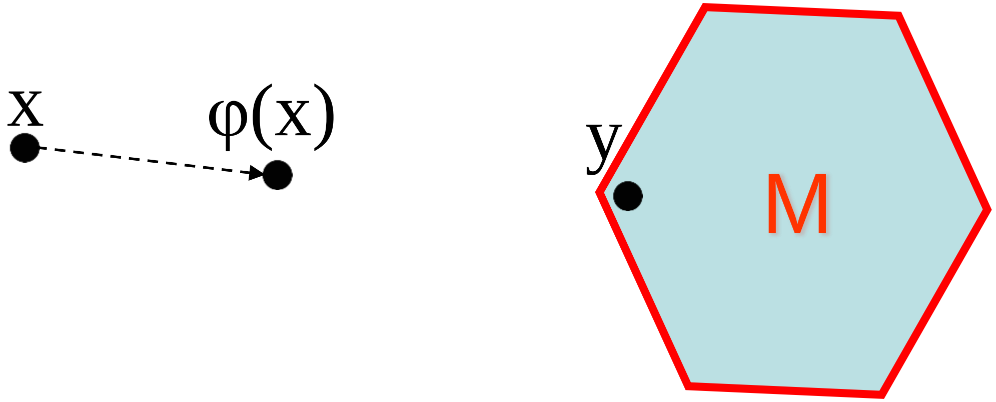
Fejerian Map

M – convex bounded set

$\varphi \in \{R^n \to R^n\}$ – M-fejerian map if

$$\varphi(y) = y, \ \forall y \in M ;$$

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



Lipót Fejér
1880 – 1959

Fejerian Map for *Quest* Phase

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

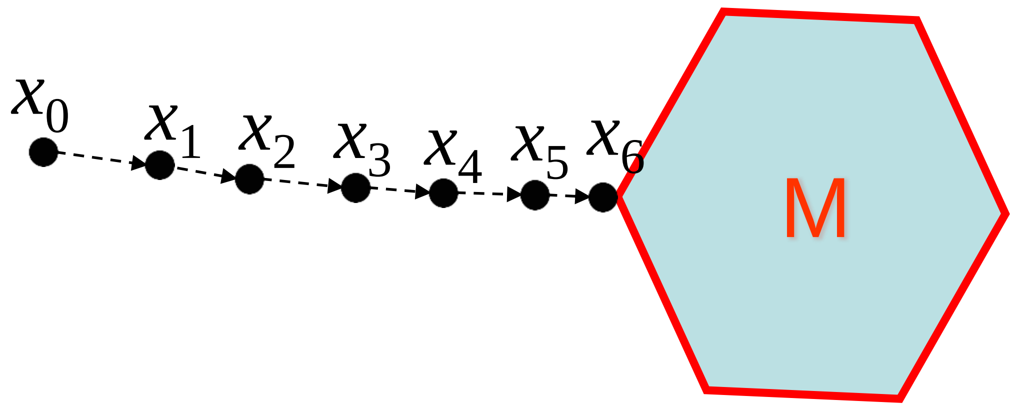
- a_{ti} – i-th line of matrix A_t
- $b_{t1}, \dots, b_{tm}$ – elements of column b_t
- m – number of lines in A_t
- t – time

Fejerian Process

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

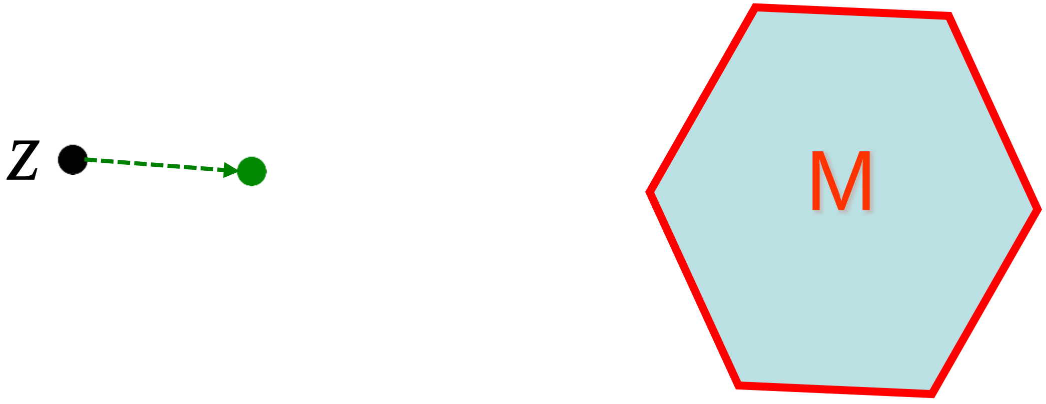
$$x_0 \in \mathbb{R}^n$$

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

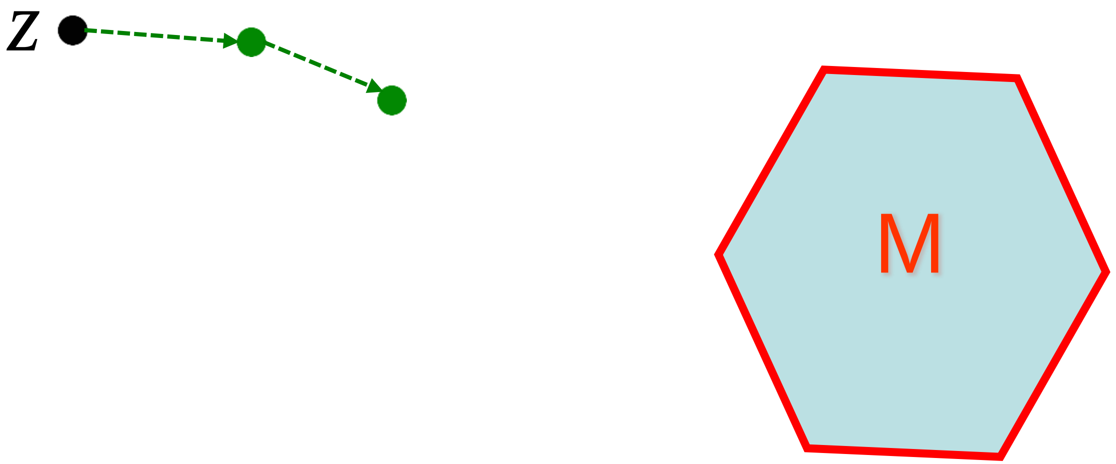


$$x_i = \varphi^{i}(x_0)$$

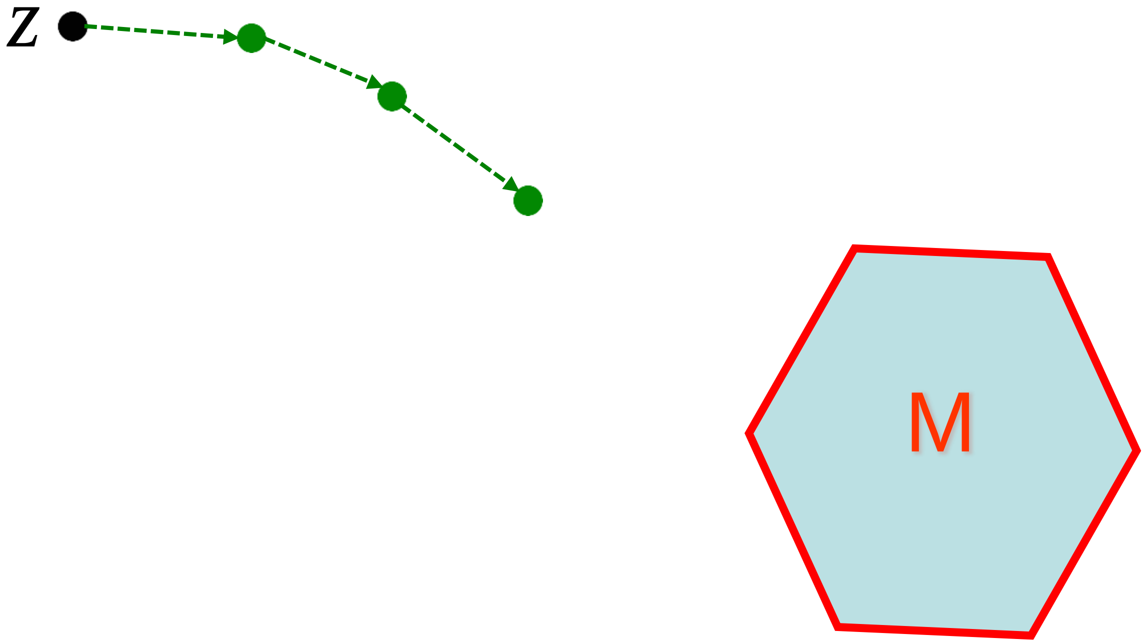
«Self-guidance» of Fejerian Process



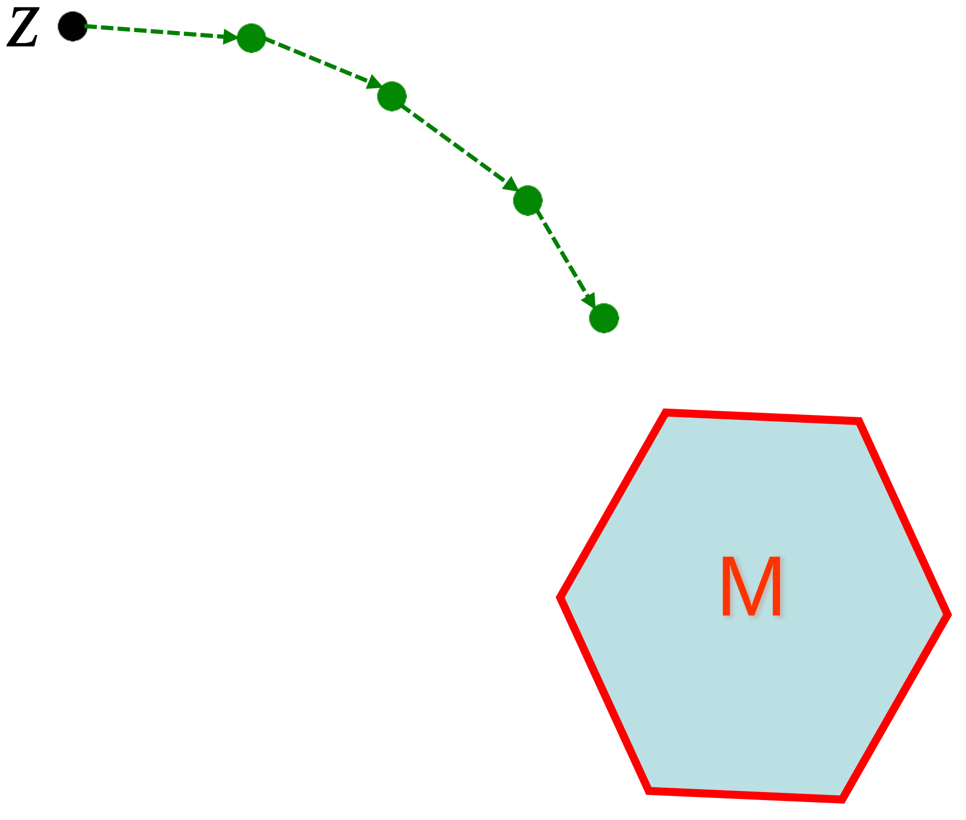
«Self-guidance» of Fejerian Process



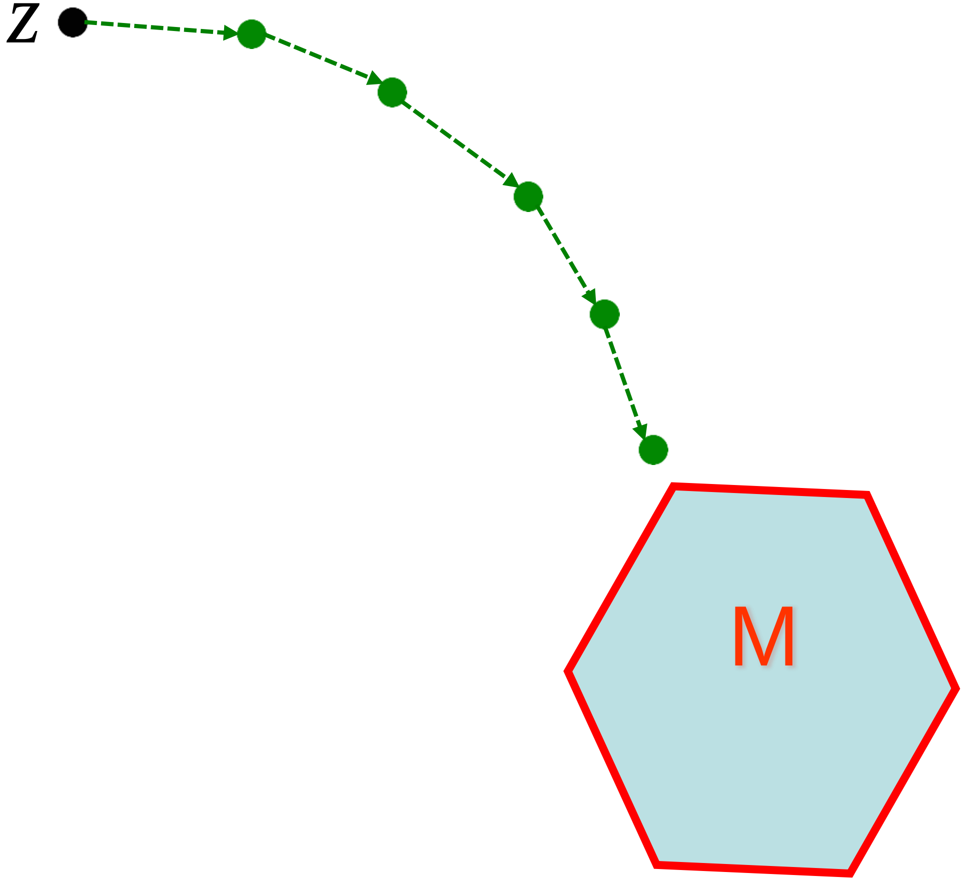
«Self-guidance» of Fejerian Process



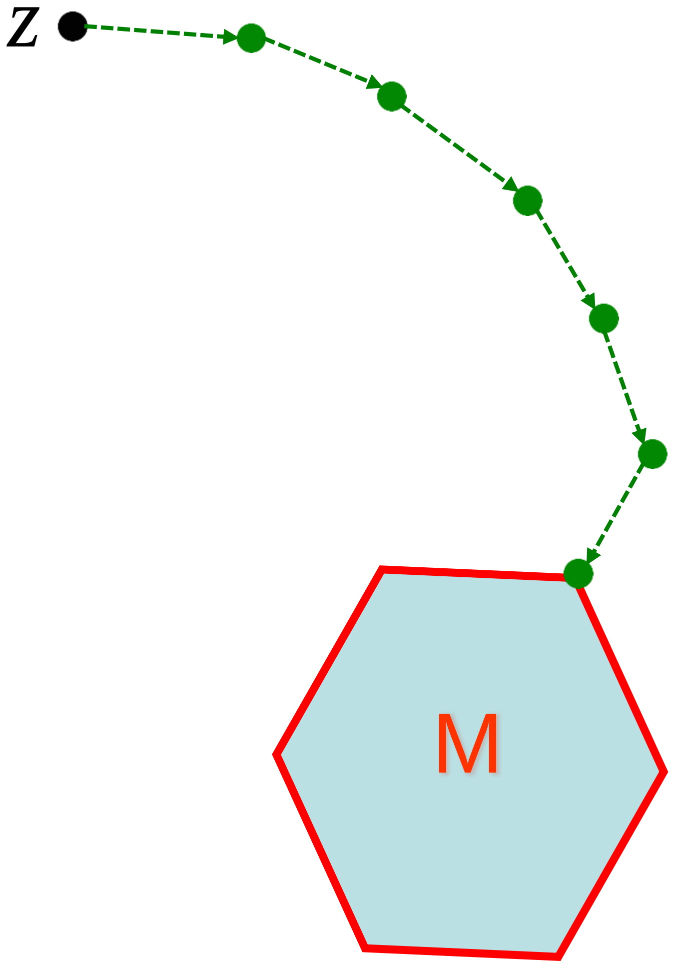
«Self-guidance» of Fejerian Process



«Self-guidance» of Fejerian Process



«Self-guidance» of Fejerian Process



Scalable Synthetic LP Problem

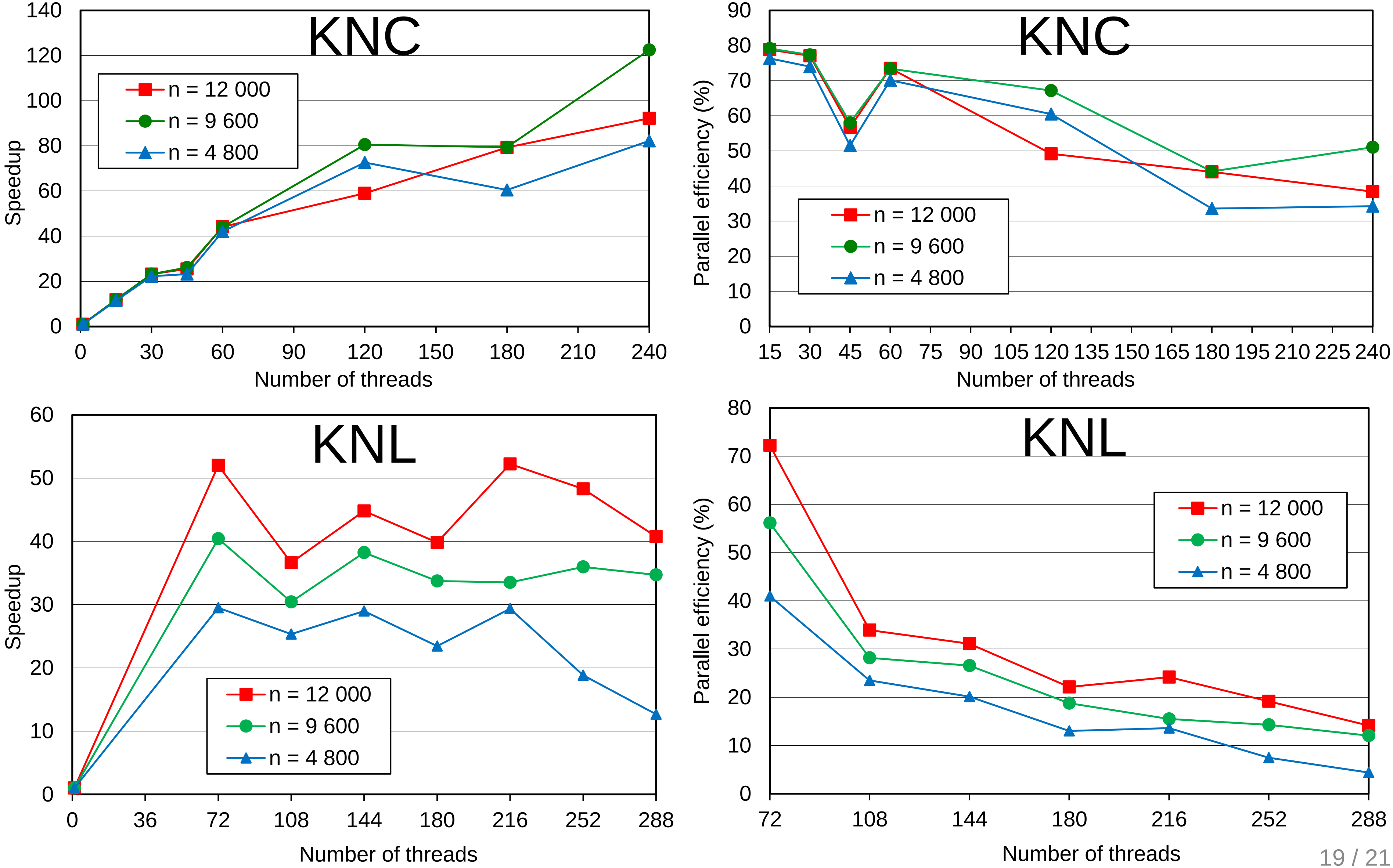
$$
\begin{cases}
x_0 & & & & & \leq & 200 \\
& & x_1 & & & \leq & 200 \\
& & & \ddots & & \cdots & \cdots \\
& & & & x_{n-1} & \leq & 200 \\
x_0 & + & x_1 & \cdots & + \; x_{n-1} & \leq & 200(n-1)+100 \\
x_0 & + & x_1 & \cdots & + \; x_{n-1} & \geq & 100 \\
x_0 & & & & & \geq & 0 \\
& & x_1 & & & \geq & 0 \\
& & & \ddots & & \cdots & \cdots \\
& & & & x_{n-1} & \geq & 0
\end{cases}
$$

$$
Q_{\max}(x) = 2x_0 + 2x_1 + \ldots + 2x_{n-2} + x_{n-1}
$$

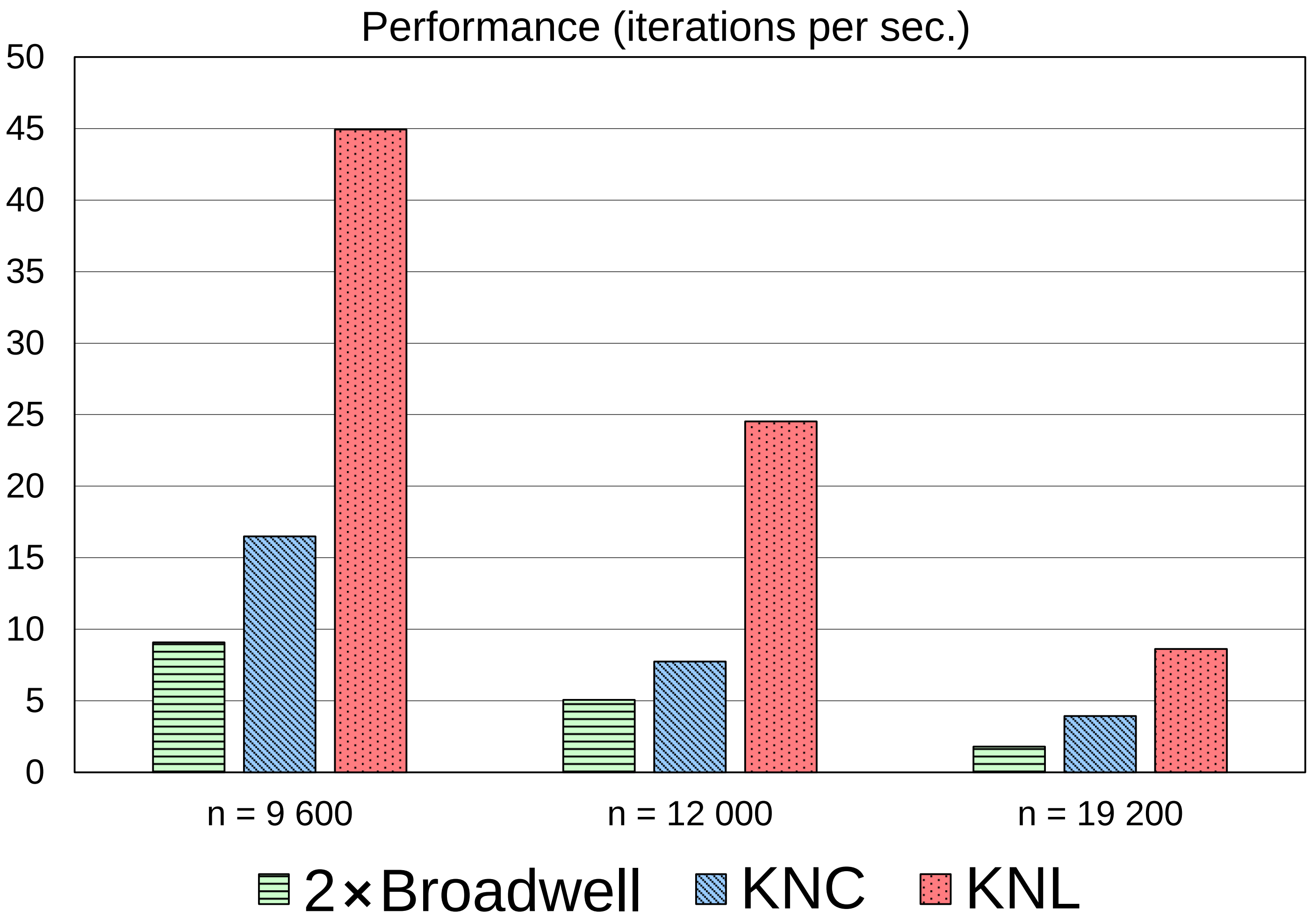
Benchmarked Processors

Code Name	Specifications
KNC	Intel Xeon Phi SE10X (61 cores, 1.1 GHz; 4 threads per core)
KNL	Intel Xeon Phi 7290 (72 cores, 1.5 GHz; 4 threads per core)
2×Broadwell	2×Intel Xeon E5 2697v3 32 cores, 2.6 GHz; 2 threads per core)

Speedup & Parallel Efficiency



Benchmark


Performance (iterations per sec.)
50
45
40
35
30
25
20
15
10
5
0
n = 9 600
n = 12 000
n = 19 200
2×Broadwell
KNC
KNL

Questions?