Международная научная конференция «Параллельные вычислительные технологии 2023»
(Санкт-Петербург, 28-30 марта 2023 г.)

Решение задач линейного программирования путем синтеза суперкомпьютерных и нейросетевых технологий на основе визуального представления n-мерных многогранников

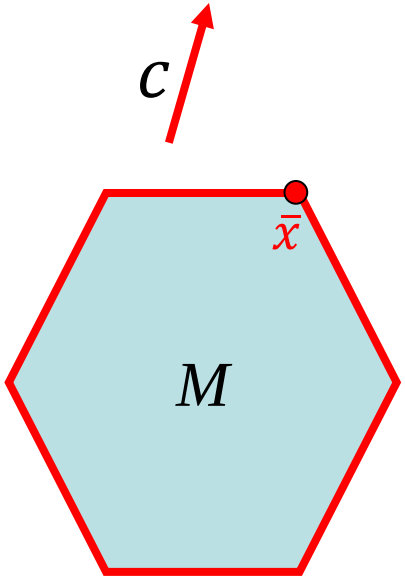
асп. Н.А. Ольховский
д.ф.-м.н. Л.Б. Соколинский

Южно-Уральский государственный университет
(национальный исследовательский университет)

Исследование выполнено при финансовой поддержке РНФ в рамках научного проекта № 23-21-00356

Задача линейного программирования

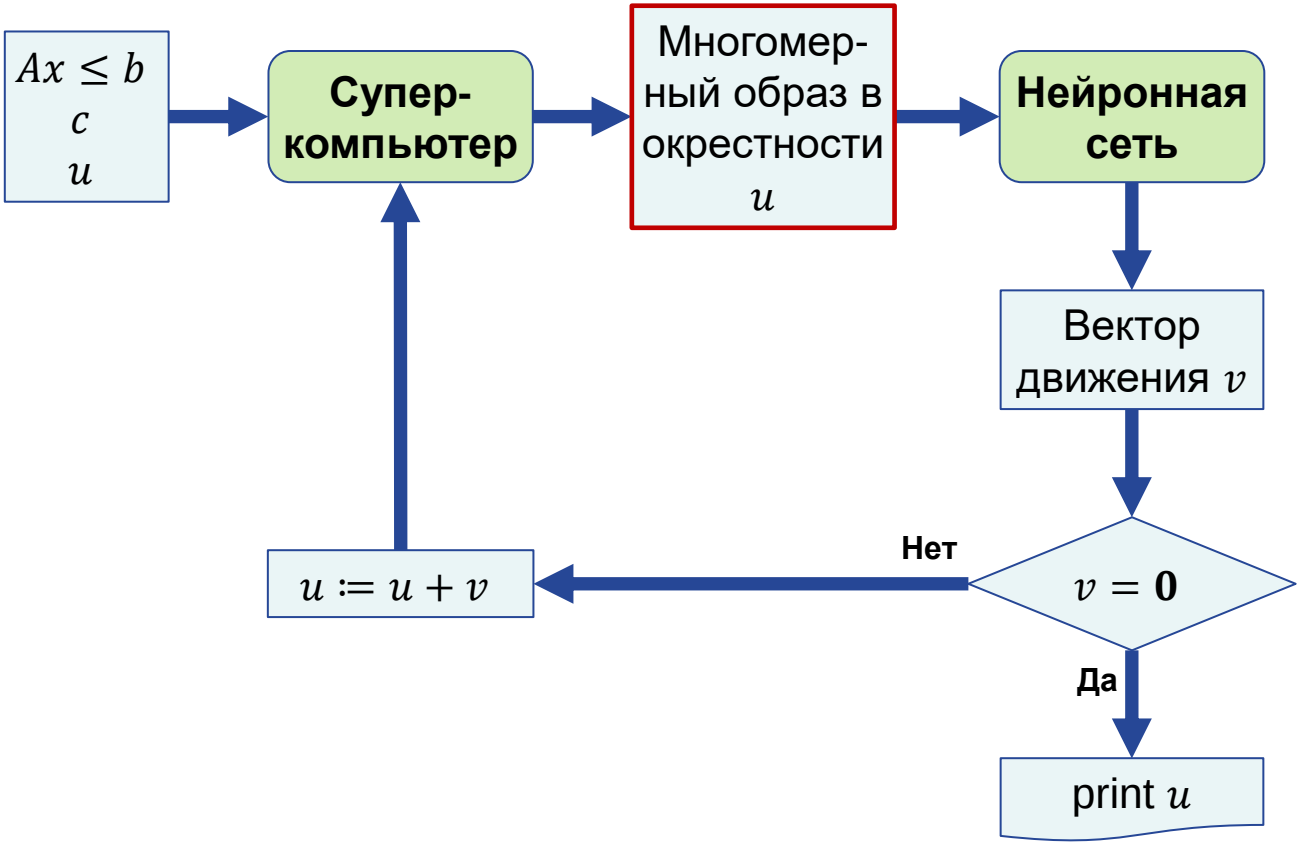
$$\bar{x} = \arg\max\{\langle c, x\rangle | Ax \leq b\}$$

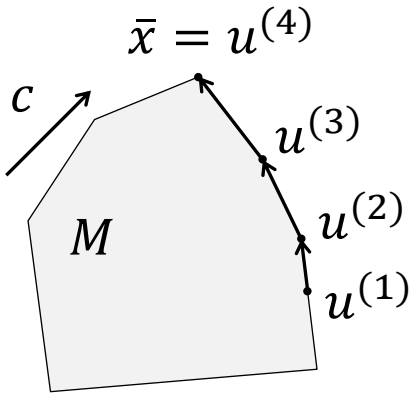


$M = \{x | Ax \leq b\}$

- $x \in \mathbb{R}^n$
- A – матрица $m \times n$
- b – вектор размерности m
- c – вектор размерности n
- $\langle c, x\rangle$ – скалярное произведение

Идея предлагаемого подхода

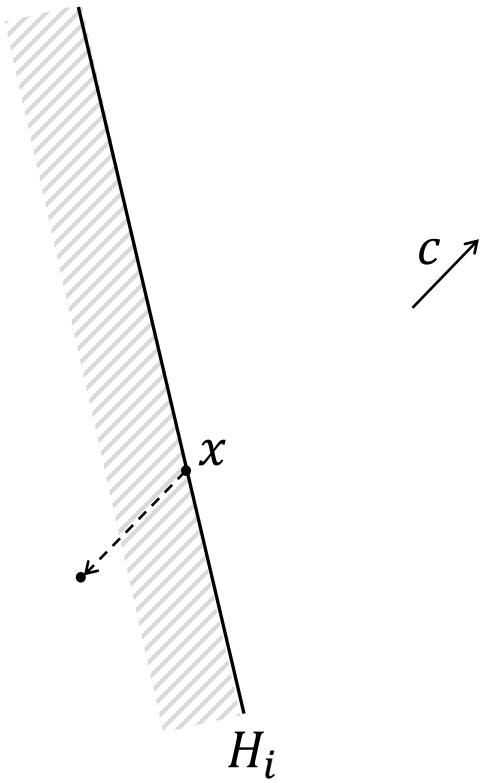




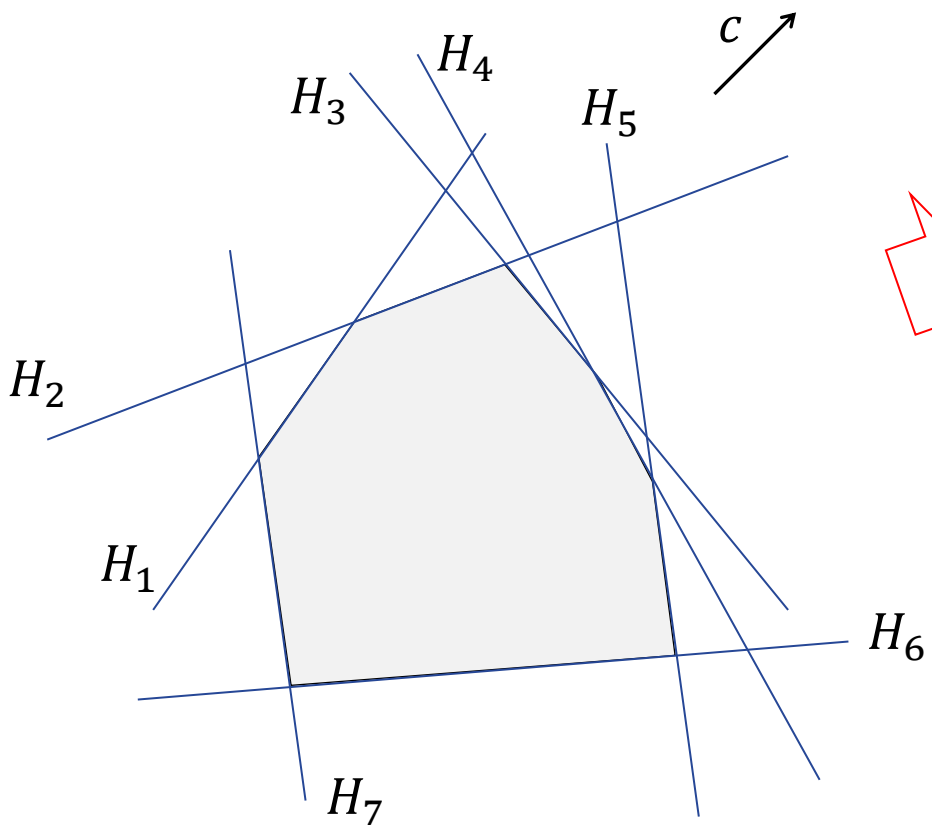
Рецессивное полупространство

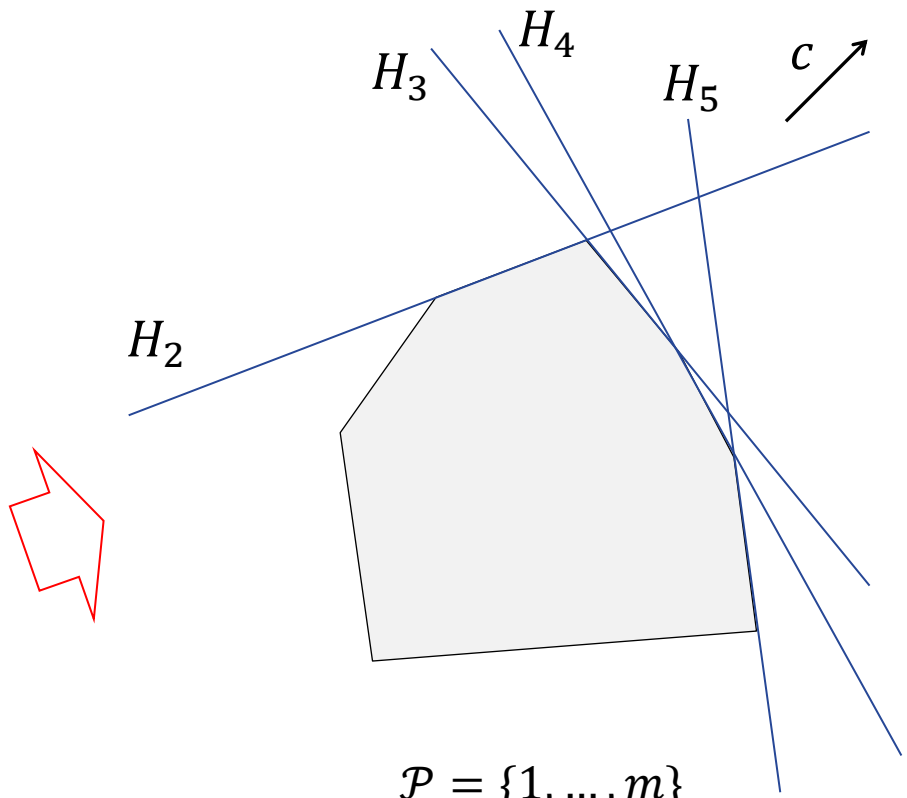
$$H_i = \{x \in \mathbb{R}^n | \langle a_i, x \rangle \leq b_i\}$$

$$\langle a_i, c \rangle > 0$$



Переход к рецессивному многограннику





$\mathcal{P} = \{1, \dots, m\}$

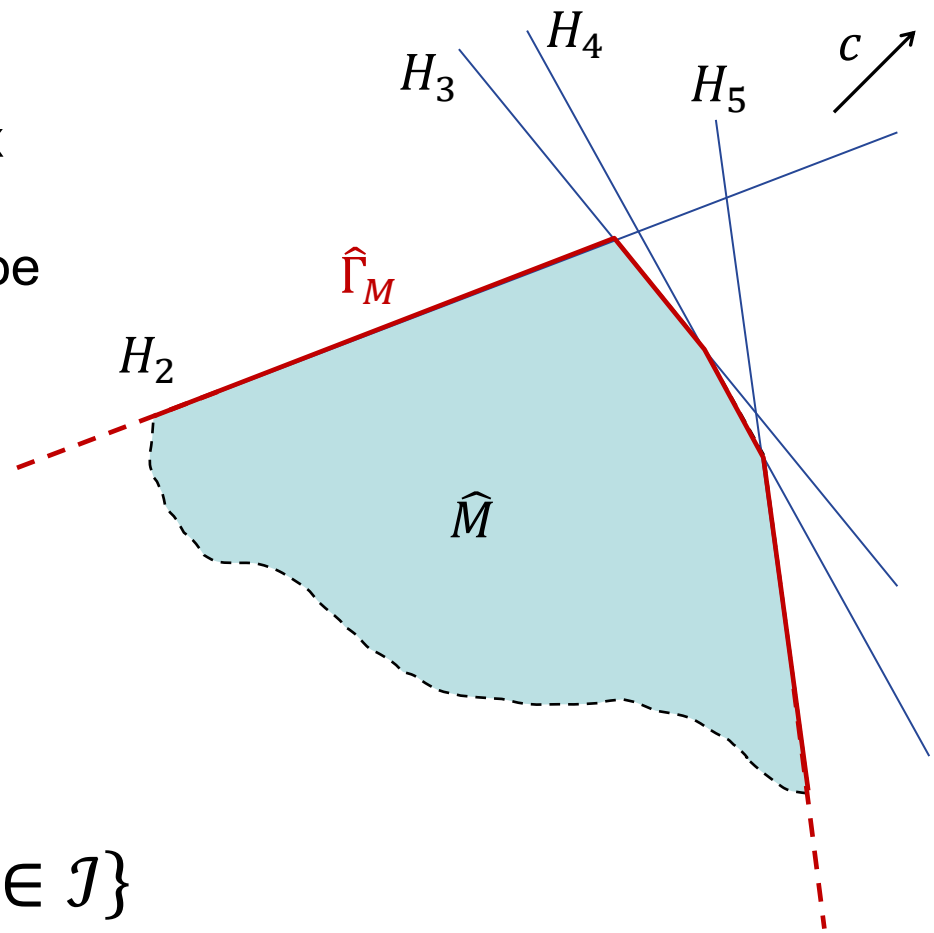
Номера полупространств, рецессивных по отношению к c:

$\mathcal{I} = \{i \in \mathcal{P} | \langle a_i, c \rangle > 0\}$

$\mathcal{I} \subset \mathcal{P}$

Рецессивный многогранник

- Многогранник, образуемый пересечением с-рецессивных полупространств
- Представляет собой замкнутое неограниченное множество



$$\widehat{M} = \bigcap_{i \in \mathcal{I}} \widehat{H}_i$$

$$\widehat{M} = \{x \in \mathbb{R}^n | \langle a_i, x \rangle \leq b_i, i \in \mathcal{I}\}$$

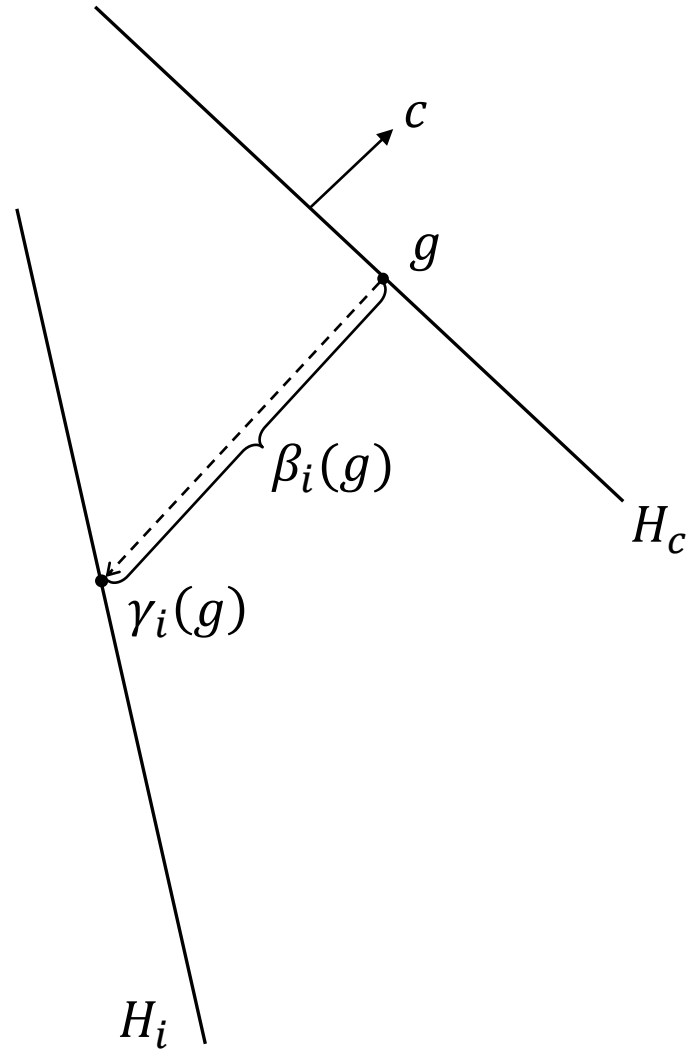
$$\hat{\Gamma}_M = \left\{x \in \mathbb{R}^n \middle| \forall \varepsilon > 0 : S_\varepsilon(x) \cap \widehat{M} \neq \emptyset \ \wedge S_\varepsilon(x) \cap \widehat{M}^C \neq \emptyset\right\}$$

Смещение

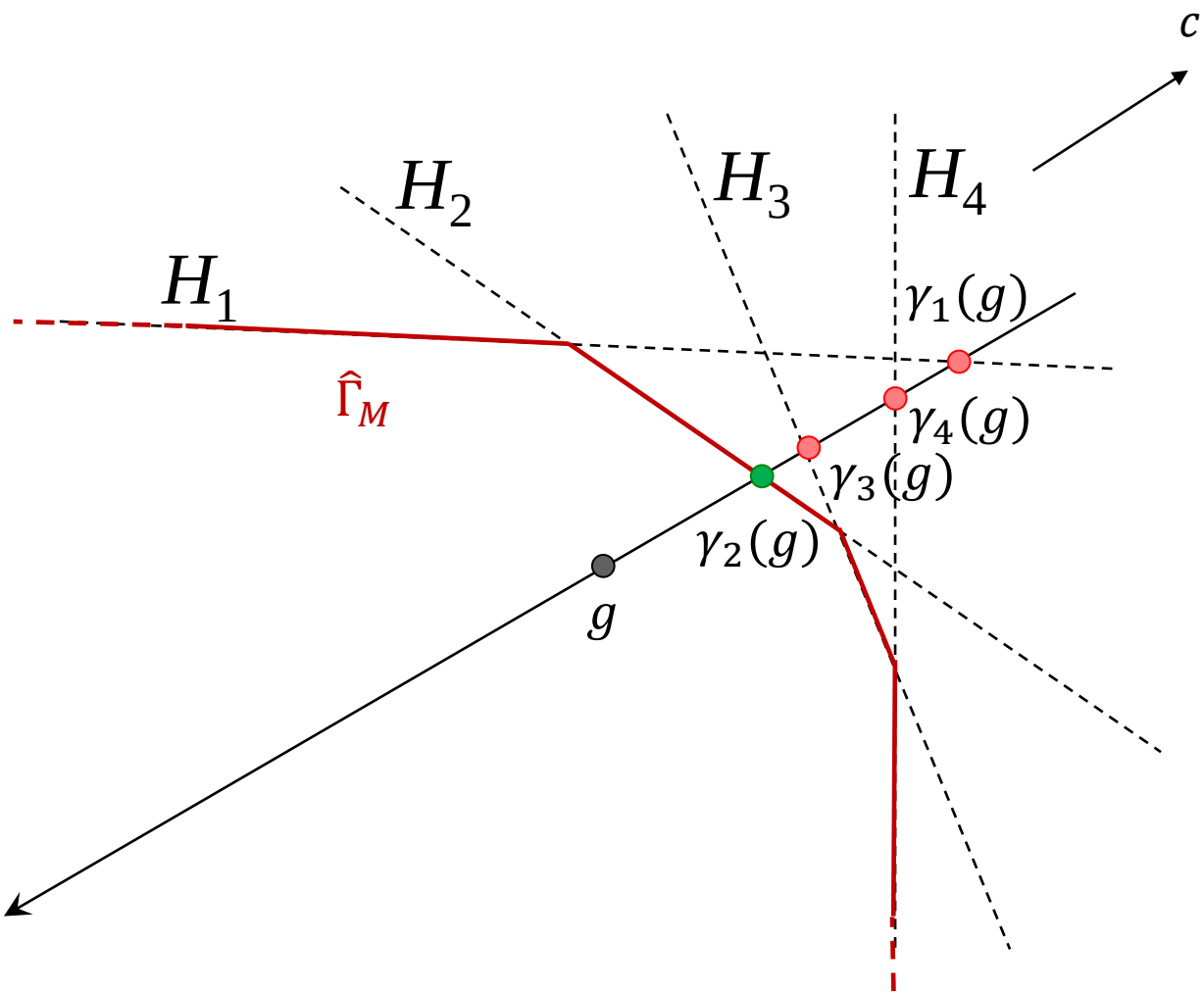
Целевая проекция: $\gamma_i(g) = g - \beta_i(g)\frac{c}{\|c\|}$

Ортогональная проекция: $\pi_{\mathrm{c}}(\gamma_{\mathrm{i}}(g)) = g$

$$\boxed{\beta_i(g) = \frac{\langle a_i, g\rangle - b_i}{\langle a_i, c\rangle}\|c\|}$$



Целевая проекция на границу рецессивного многогранника



$$\hat{\beta}(g) = \max_{i \in \mathcal{J}} \beta_i(g)$$

$$\hat{\gamma}(g) = g - \hat{\beta}(g) \frac{c}{\|c\|}$$

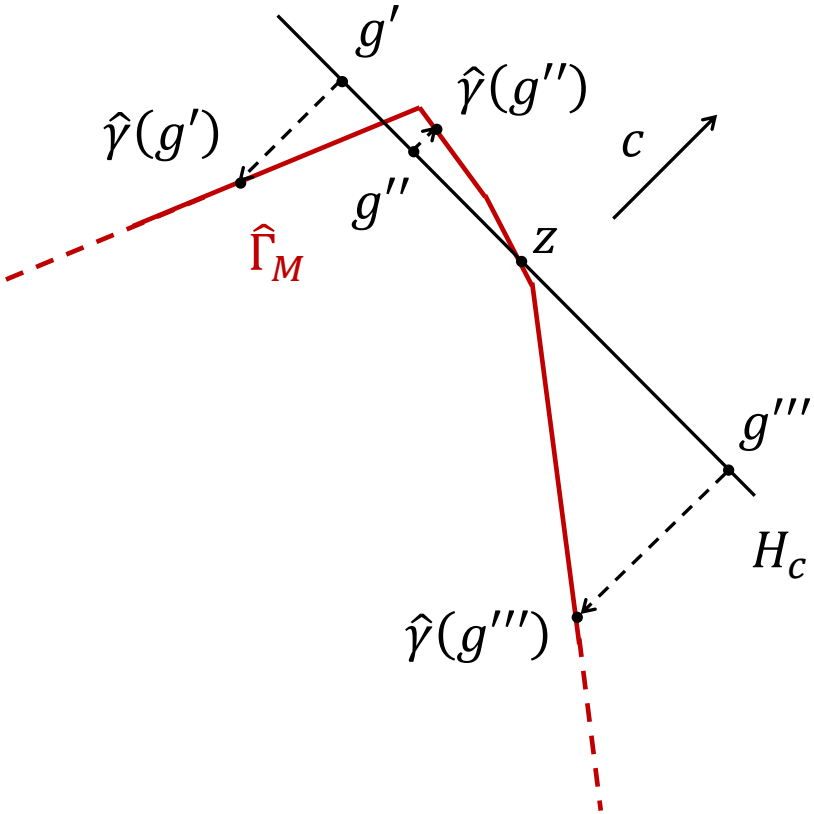
$$\hat{\gamma}(g) = \gamma_2(g)$$

$$\hat{\beta}(g) = \beta_2(g)$$

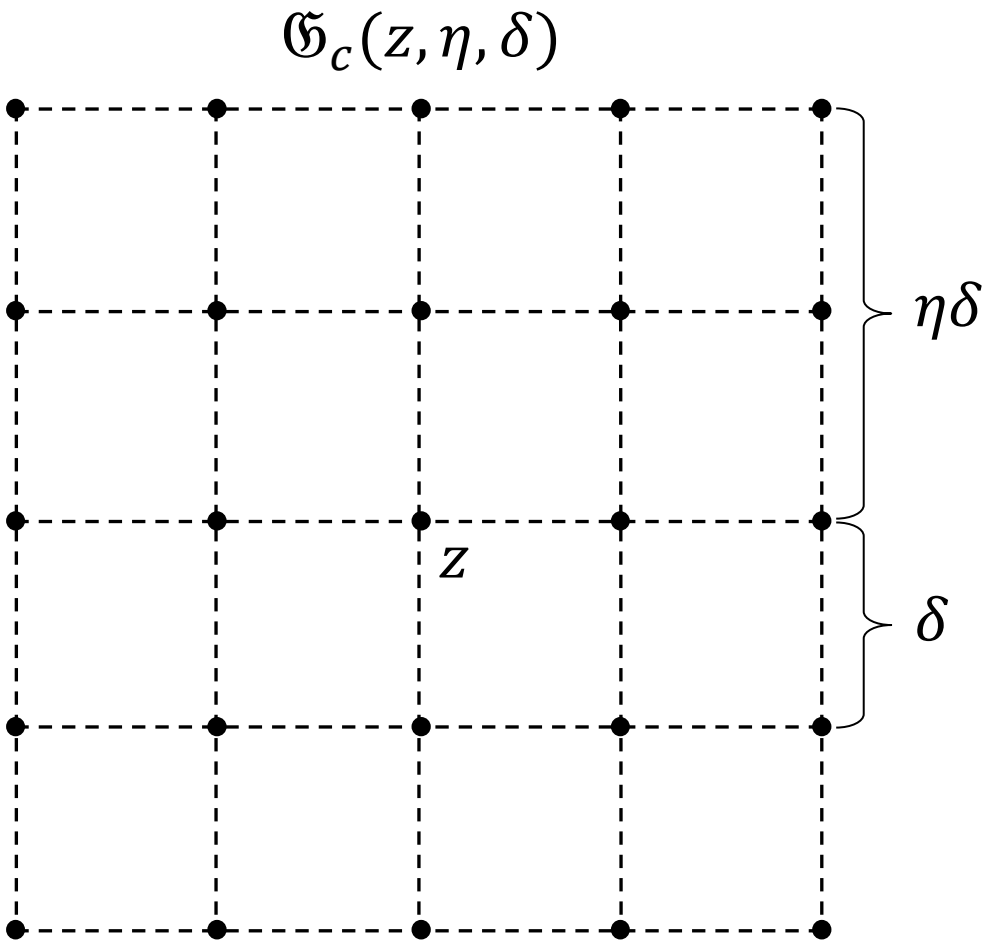
Построение рецептивного поля

Рецептивная гиперплоскость: $H_c = \{x \in \mathbb{R}^n \mid \langle c, x - z \rangle = 0\}$

Целевая проекция: $\hat{\gamma} \ : \ H_c \to \hat{\Gamma}_M$

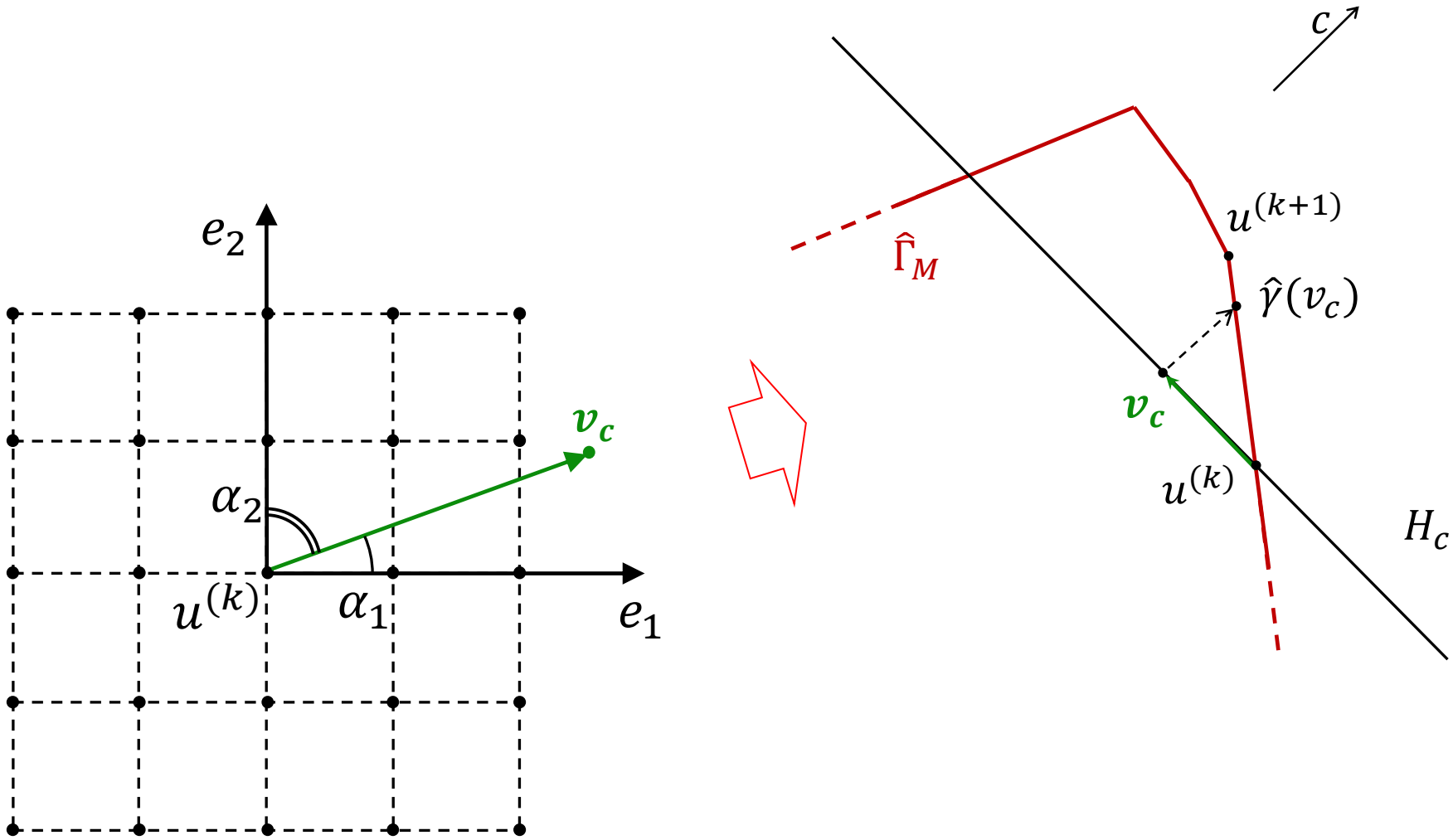


Рецептивное поле (flat)

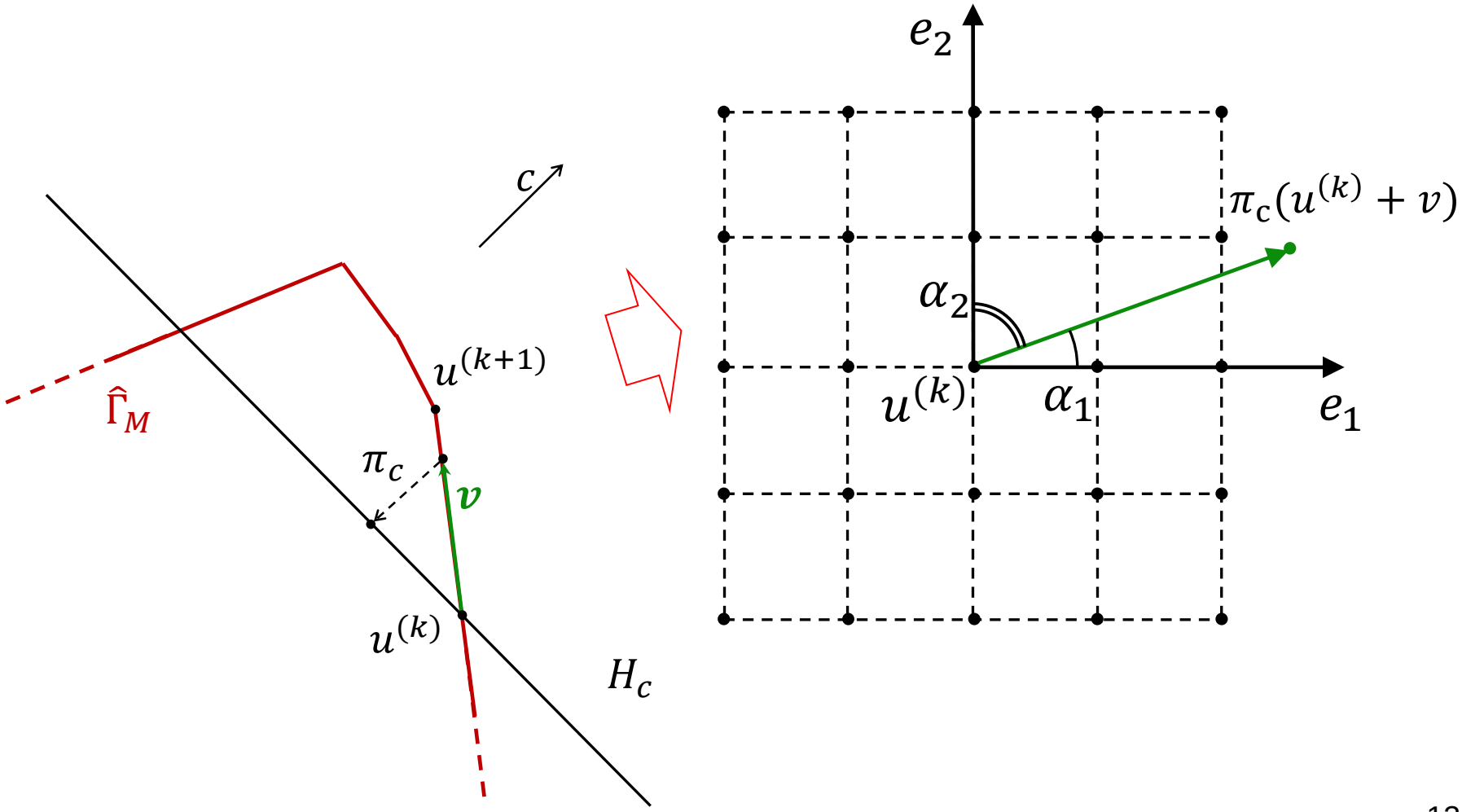


η – ранг
δ – шаг

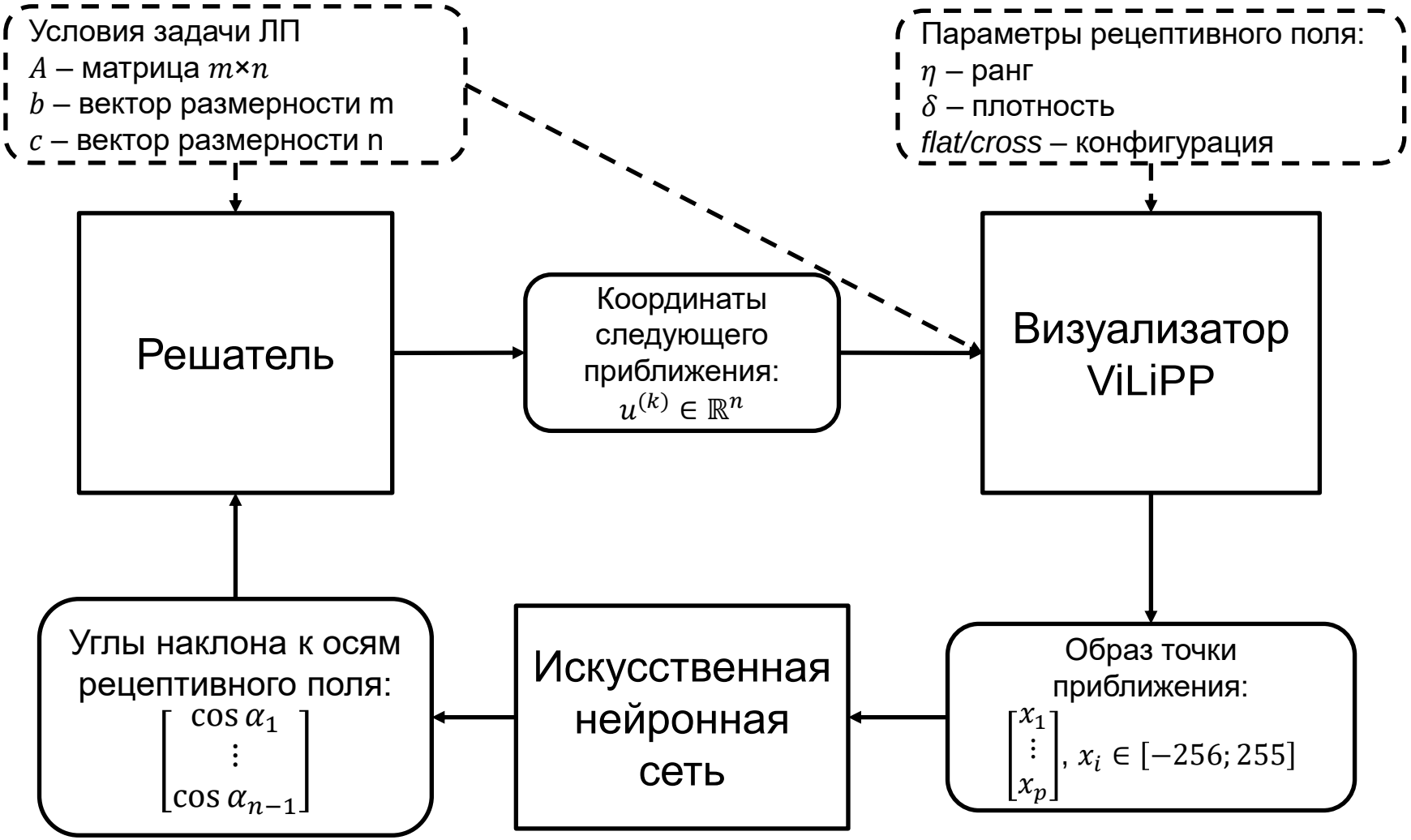
Вектор движения



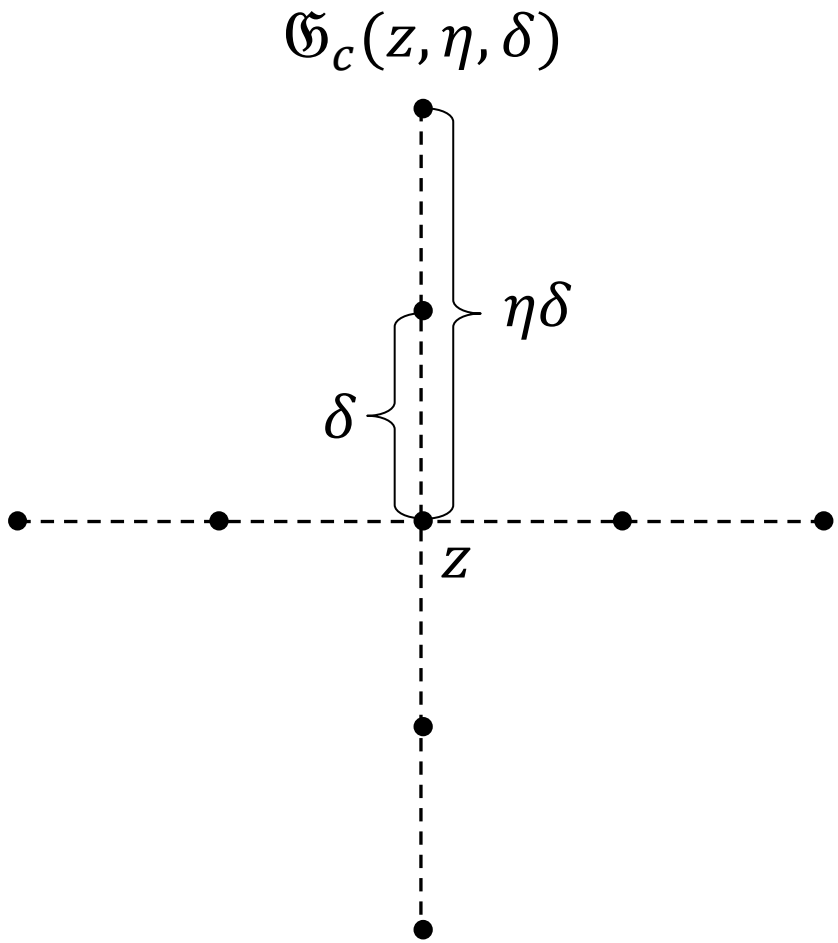
Создание обучающего прецедента



Архитектура программного комплекса



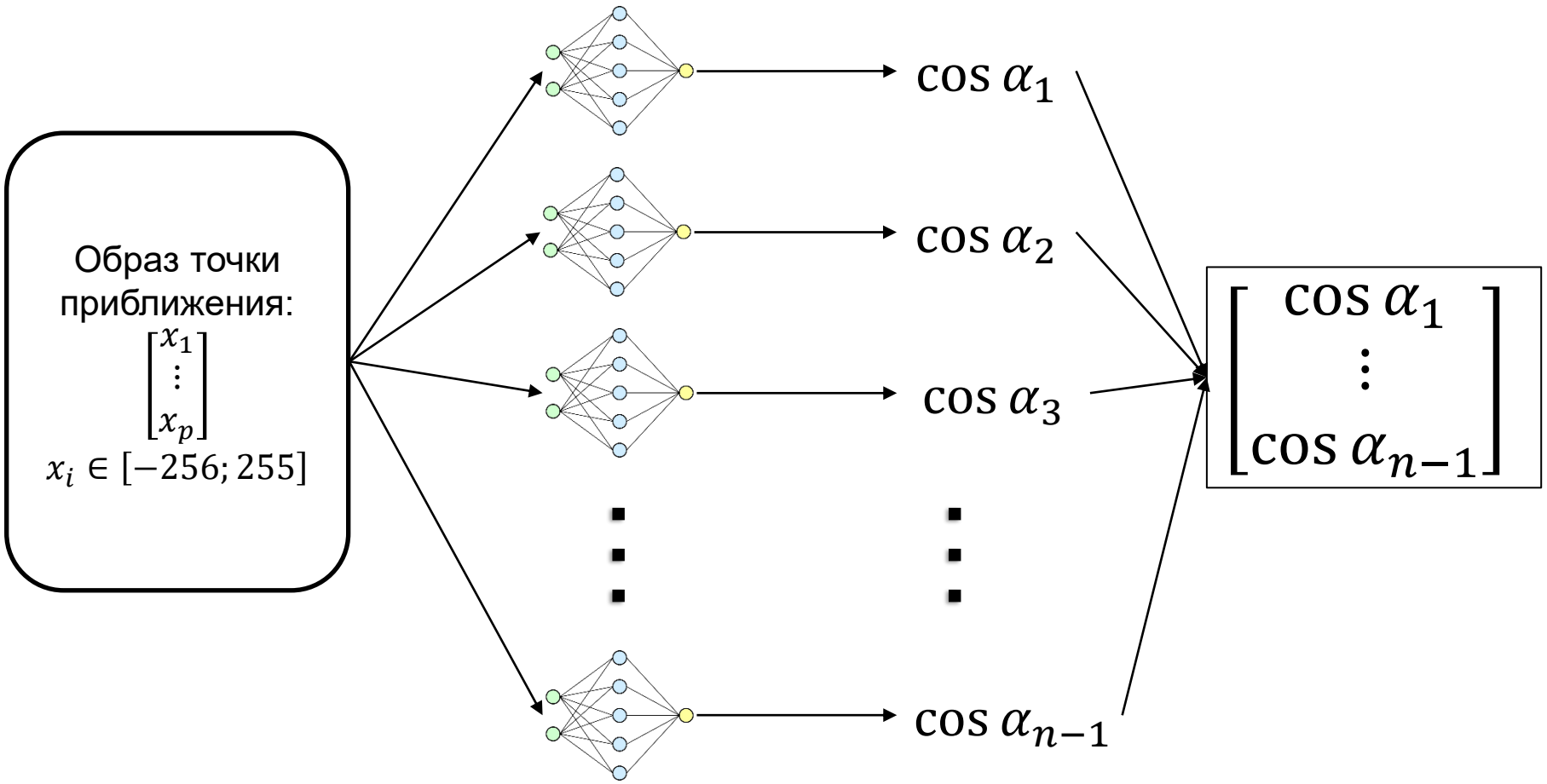
Рецептивное поле (cross)



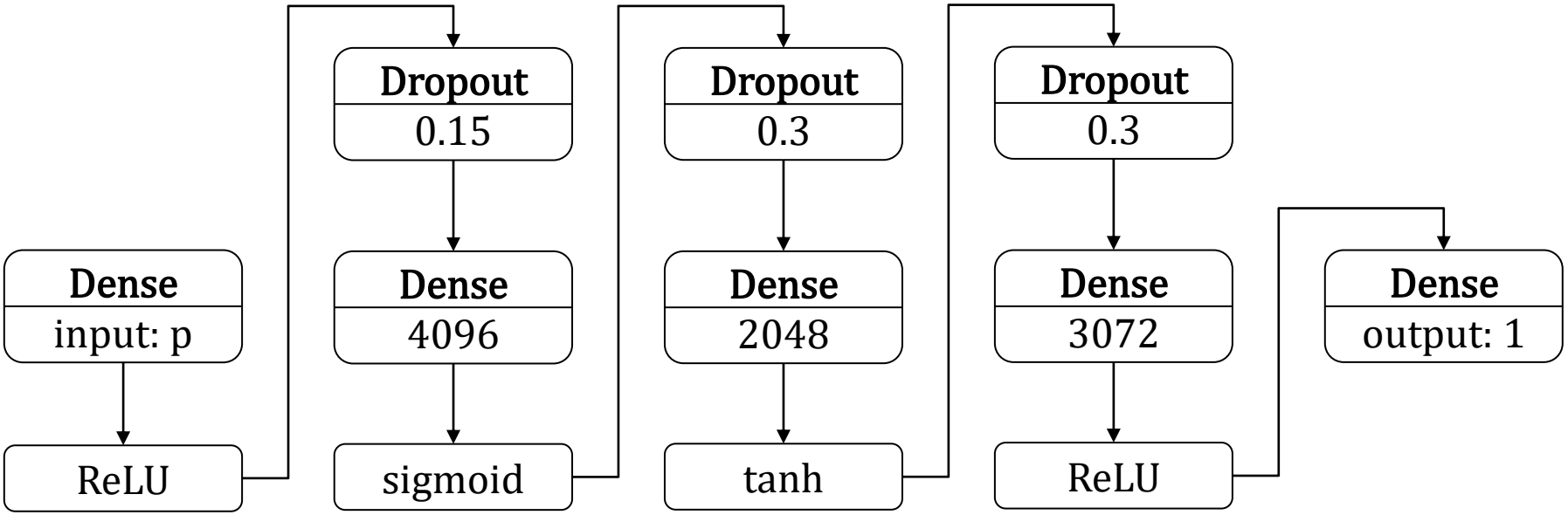
η – ранг

δ – плотность

Архитектура нейронной сети



Архитектура нейронной сети



Для **cross:** $p = 2\eta(n-1)+1$

Для **flat:** $p = (2\eta+1)^{n-1}$

Эксперименты

Параметры

Число случайных задач ЛП

$50\,000$ (для $\mathbb{R}^3$); $20\,000$ (для $\mathbb{R}^4$);

Число случайных ограничений

7 (для $\mathbb{R}^3$); 8 (для $\mathbb{R}^4$);

Конфигурации рецептивного поля

Ранг (η)	Плотность (δ)	Количество точек рецептивного поля			
		K^2_{flat}	K^2_{cross}	K^3_{flat}	K^3_{cross}
5	1.00	121	21	1331	31
4	1.25	81	17	729	25
3	1.66667	49	13	343	19
2	2.5	25	9	125	13
1	5	9	5	27	7

Метрики

Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{n}\sum\left|y_{true} - y_{pred}\right|$$

Cosine Similarity (CS)

$$CS = \frac{\sum(y_{true} \cdot y_{pred})}{\sqrt{\sum y_{true}^2} \cdot \sqrt{\sum y_{pred}^2}} \cdot 100$$

y_{true} – *истинное значение*
y_{pred} – *ответ ИНС*
n – *размер выборки (batch size)*

Достигнутая точность

Для $\mathbb{R}^3$:

flat

$\mathrm{CS} = 97{,}4\%$

$\mathrm{MAE} = 0{,}0311$

cross

$\mathrm{CS} = 96{,}2\%$

$\mathrm{MAE} = 0{,}0369$

Для $\mathbb{R}^4$:

flat

$\mathrm{CS} = 95{,}2\%$

$\mathrm{MAE} = 0{,}0437$

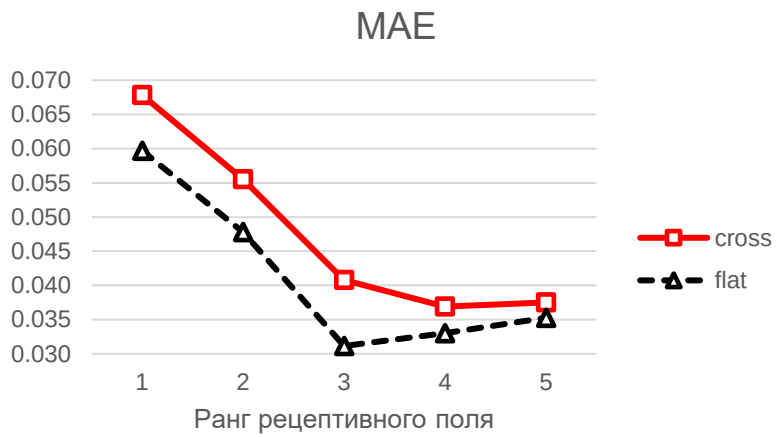
cross

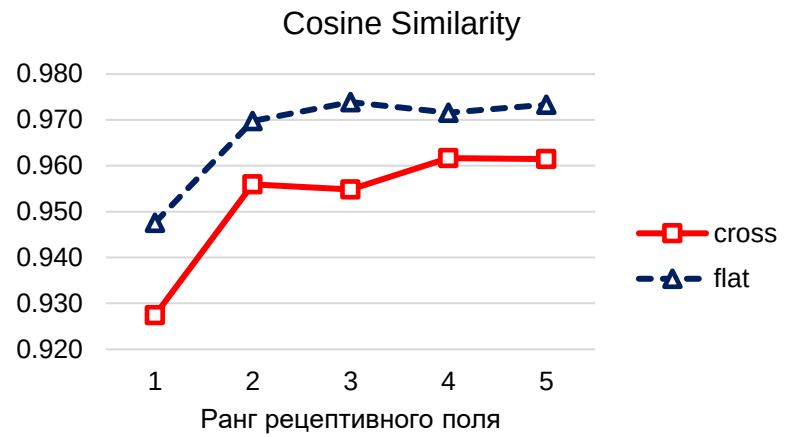
$\mathrm{CS} = 95{,}9\%$

$\mathrm{MAE} = 0{,}0499$

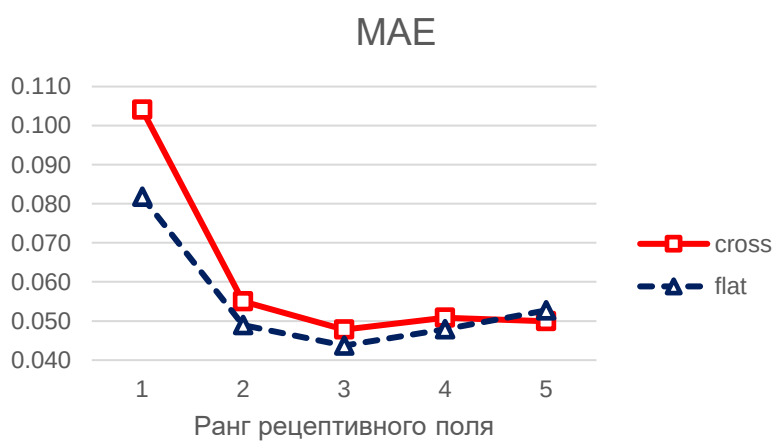
Зависимость точности от η

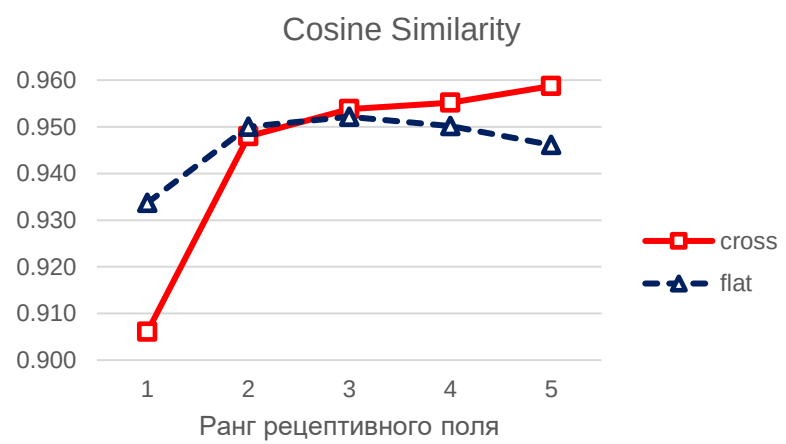
$\mathbb{R}^3$





$\mathbb{R}^4$





Спасибо за внимание!

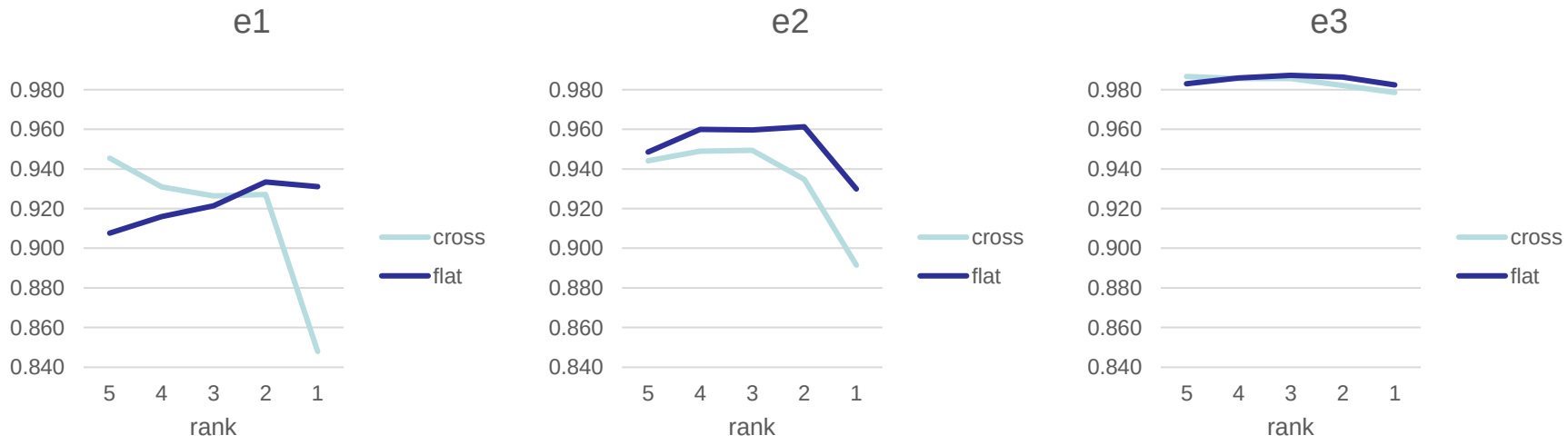
Результаты cross в $\mathbb{R}^5 - \mathbb{R}^{10}$

Cosine Similarity	Номер оси									
Размерность	1	2	3	4	5	6	7	8	9	10
5	91.91%	90.16%	96.86%	55.46%						
6	90.67%	88.33%	96.43%	52.50%	46.57%					
7	86.54%	83.52%	95.03%	64.56%	49.14%	22.13%				
8	78.71%	76.12%	91.50%	49.32%	40.64%	39.55%	23.27%			
9	52.70%	47.85%	83.84%	45.53%	42.10%	36.25%	38.42%	28.05%		
10	46.98%	45.80%	83.21%	41.63%	37.50%	37.95%	37.21%	36.48%	25.38%	

MAE	Номер оси									
Размерность	1	2	3	4	5	6	7	8	9	10
5	0.05873981	0.06582254	0.05029353	0.05081702						
6	0.06318476	0.06391414	0.06639612	0.06809764	0.04662187					
7	0.07515518	0.07796316	0.08462689	0.06014385	0.06863652	0.05404814				
8	0.10648490	0.10905033	0.11975247	0.08245312	0.08373992	0.08706762	0.07866536			
9	0.19792171	0.19899932	0.22261596	0.15107401	0.15275109	0.15594846	0.14591452	0.13258035		
10	0.20090352	0.20409746	0.24907678	0.15216495	0.15666085	0.15696955	0.15343474	0.15138265	0.12882274	

Неожиданность в $\mathbb{R}^4$

Достигаемая точность сильно различается для разных осей рецептивного поля



Эксперименты

1. Достигнутая точность
2. Оценка времени решения модельной задачи
3. Что-то из графиков:
   - Зависимость точности от плотности рецептивного поля
   - Зависимость времени от плотности рецептивного поля (аналитически)
   - Зависимость времени от размерности

Актуальность

- Отсутствуют методы решения задачи ЛП в режиме реального времени
- ИНС прямого распространения (включая сверточные сети) работают в режиме реального времени
- Критика подхода Хопфилда-Танка
- [Наш метод ориентирован на будущие мощные суперкомпьютеры. Нейронные сети заработали, когда появились GPU]

Цель исследования

Разработать модели, методы и алгоритмы позволяющие применить искусственные нейронные сети прямого распространения для решения многомерных задач линейного программирования

Предшествующие работы

Все публикации основываются на модели Хопфилда-Танка

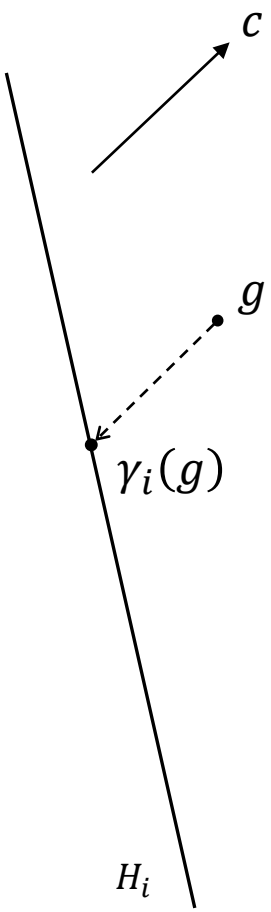
- Используется рекуррентная нейронная сеть на базе минимизации энергетической функции
- Сеть работает до достижения состояния равновесия (вход=выход)
- Архитектура нейронной сети зависит от задачи
- Необходимо придумывать энергетическую функцию и доказывать сходимость
- Нельзя предсказать количество тактов работы сети
- Достижение равновесия не гарантирует правильность ответа
- Проблемы с устойчивостью по Ляпунову

- Tank D.W., Hopfield J.J. Simple ‘neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit // IEEE transactions on circuits and systems. 1986. Vol. CAS-33, no. 5. P. 533-541. https://doi.org/10.1109/TCS.1986.1085953.
- Kennedy M.P., Chua L.O. Unifying the Tank and Hopfield Linear Programming Circuit and the Canonical Nonlinear Programming Circuit of Chua and Lin // IEEE Transactions on Circuits and Systems. 1987. Vol. 34, no. 2. P. 210-214. https://doi.org/10.1109/TCS.1987.1086095.
- Nonlinear Switched-Capacitor “Neural” Networks for Optimization Problems / Rodriguez-Vazquez A., Dominguez-Castro R., Rueda A., Huertas J., and Sanchez-Sinencio E. // IEEE Transactions on Circuits and Systems. 1990. Vol. 37, no. 3. P. 384-398. https://doi.org/10.1109/31.52732.
- Zak S.H., Upatising V. Solving Linear Programming Problems with Neural Networks: A Comparative Study // IEEE Transactions on Neural Networks. 1995. Vol. 6, no. 1. P. 94-104. https://doi.org/10.1109/72.363446.
- Malek A., Yari A. Primal-dual solution for the linear programming problems using neural networks // Applied Mathematics and Computation. 2005. Vol. 167, no. 1. P. 198-211. https://doi.org/10.1016/J.AMC.2004.06.081.
- Liu X., Zhou M. A one-layer recurrent neural network for non-smooth convex optimization subject to linear inequality constraints // Chaos, Solitons and Fractals. 2016. Vol. 87. P. 39-46. https://doi.org/10.1016/j.chaos.2016.03.009.

… … … … … … … … … … … …

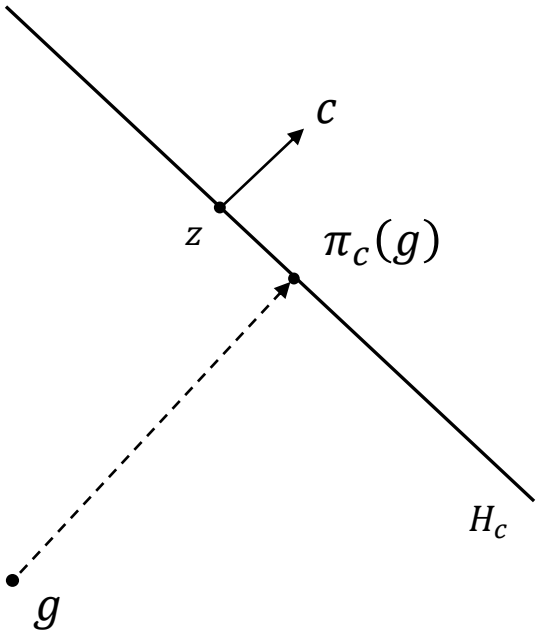
Проекция на гиперплоскость

$$\gamma_i(g) = g - \frac{\langle \vec{a}_i, g \rangle - b_i}{\langle \vec{a}_i, c \rangle} c$$



Ортогональная проекция

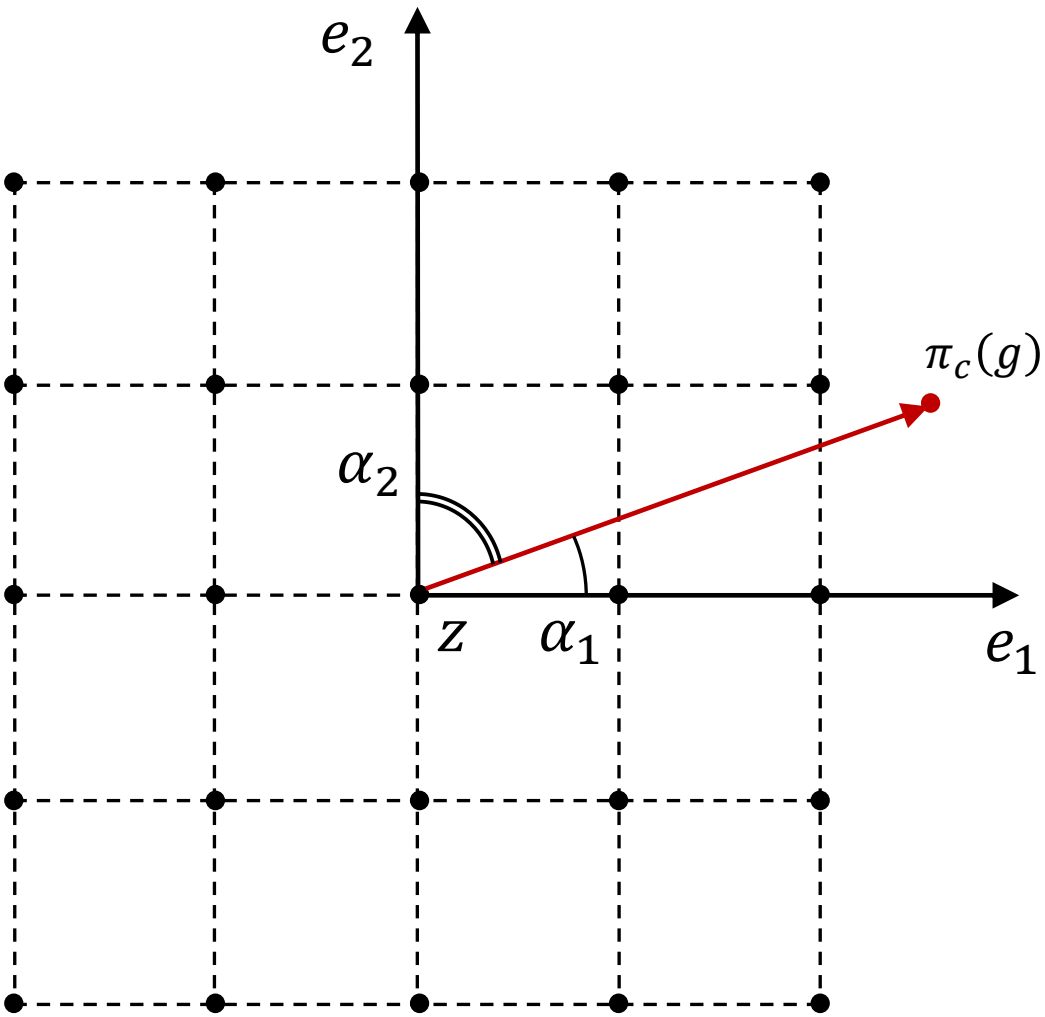
$$\pi_c(g) = g - \frac{\langle c, g - z \rangle}{\|c\|^2} c$$



$$\gamma_i(g) = g - \beta_i(g) \frac{c}{\|c\|}$$

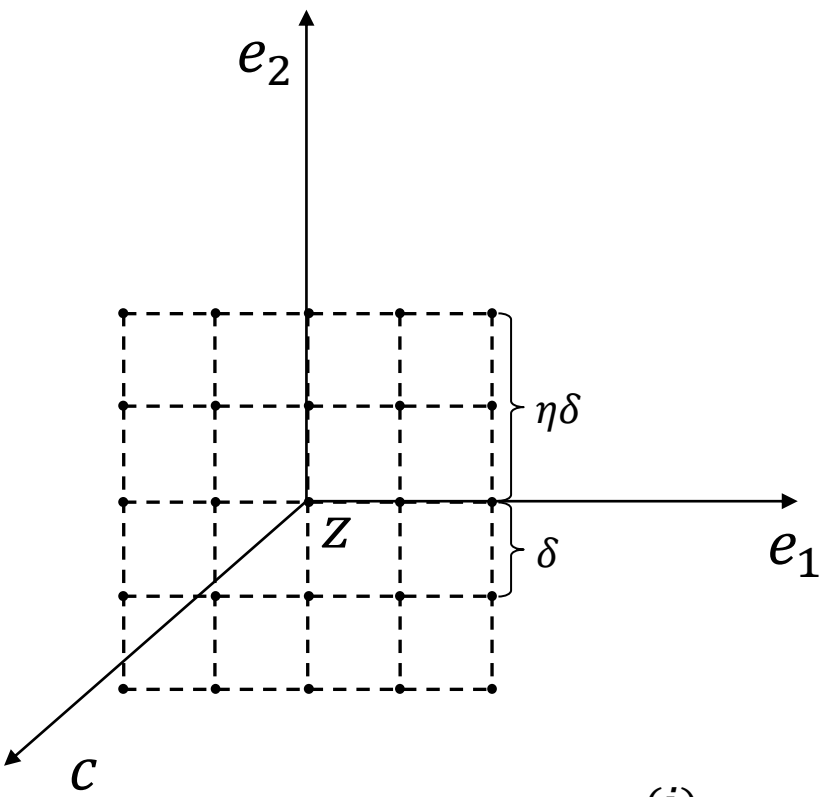
$$\pi_{\mathrm{c}}(\gamma_{\mathrm{i}}(\mathrm{g})) = \mathrm{g}$$

Coordinate angles



$$\cos\alpha_i = \frac{\langle e_i, \pi_c(g) - z\rangle}{\underbrace{\|e_i\|}_{=1} \cdot \|\pi_c(g) - z\|}$$

Размещение рецептивного поля в H_c



$$c^{(0)} = c = (c_1, c_2, c_3, c_4, \ldots, c_{n-1}, c_n);$$

$$c^{(1)} = \begin{cases} \left(-\frac{1}{c_1}\sum_{i=2}^{n} c_i^2, c_2, c_3, c_4, \ldots, c_{n-1}, c_n\right), \text{ если } c_1 \neq 0; \\ (1, 0, \ldots, 0), \text{ если } c_1 = 0; \end{cases}$$

$$c^{(2)} = \begin{cases} \left(0, -\frac{1}{c_2}\sum_{i=3}^{n} c_i^2, c_3, c_4, \ldots, c_{n-1}, c_n\right), \text{ если } c_2 \neq 0; \\ (0, 1, 0, \ldots, 0), \text{ если } c_2 = 0; \end{cases}$$

$$c^{(3)} = \begin{cases} \left(0, 0, -\frac{1}{c_3}\sum_{i=4}^{n} c_i^2, c_4, \ldots, c_{n-1}, c_n\right), \text{ если } c_3 \neq 0; \\ (0, 0, 1, 0, \ldots, 0), \text{ если } c_3 = 0; \end{cases}$$

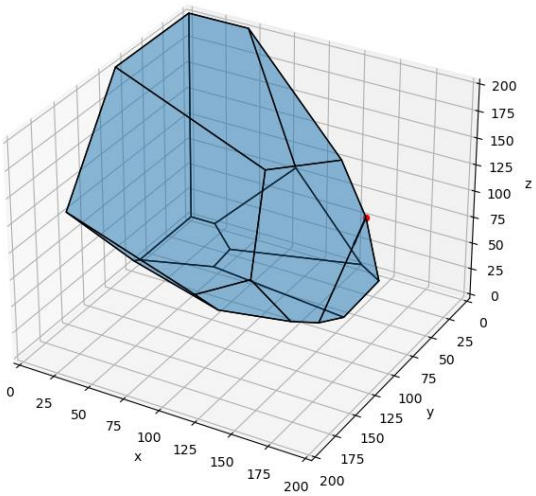
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

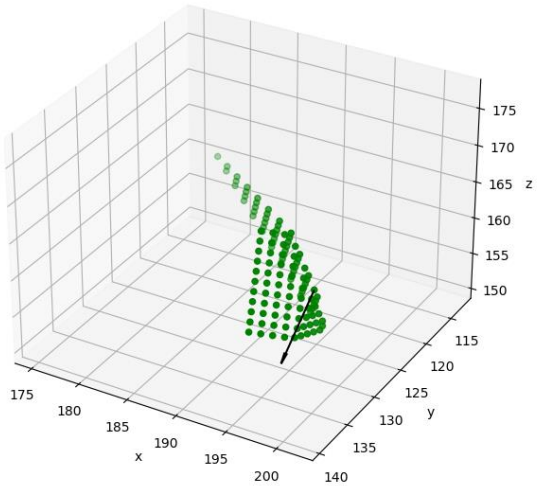
$$c^{(n-2)} = \begin{cases} \left(0, \ldots, 0, -\frac{1}{c_{n-2}}\sum_{i=n-1}^{n} c_i^2, c_{n-1}, c_n\right), \text{ если } c_{n-2} \neq 0; \\ (0, \ldots, 0, 1, 0, 0), \text{ если } c_{n-2} = 0; \end{cases}$$

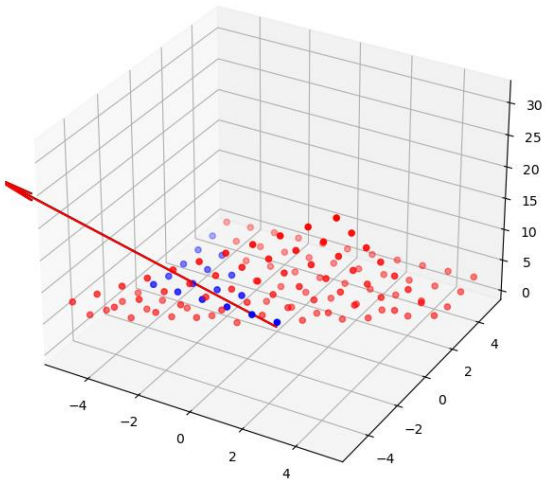
$$c^{(n-1)} = \begin{cases} \left(0, \ldots, 0, -\frac{c_n^2}{c_{n-1}}, c_n\right), \text{ если } c_{n-1} \neq 0; \\ (0, \ldots, 0, 0, 1, 0), \text{ если } c_{n-1} = 0. \end{cases}$$

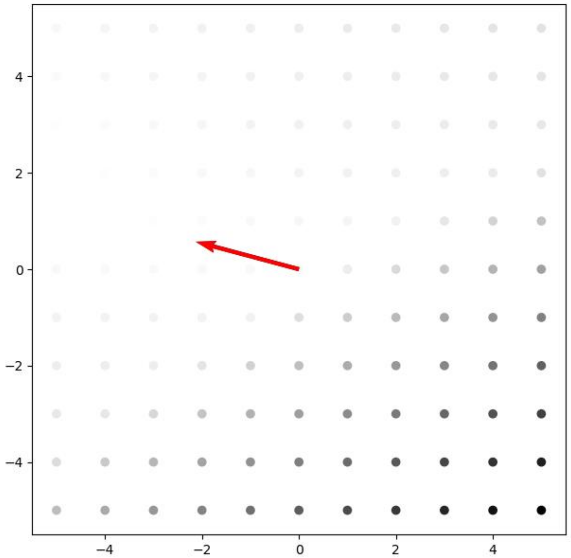
$$e^{(i)} = \frac{c^{(i)}}{\|c^{(i)}\|} \; (i = 1, \ldots, n-1)$$

Образ рецептивного поля



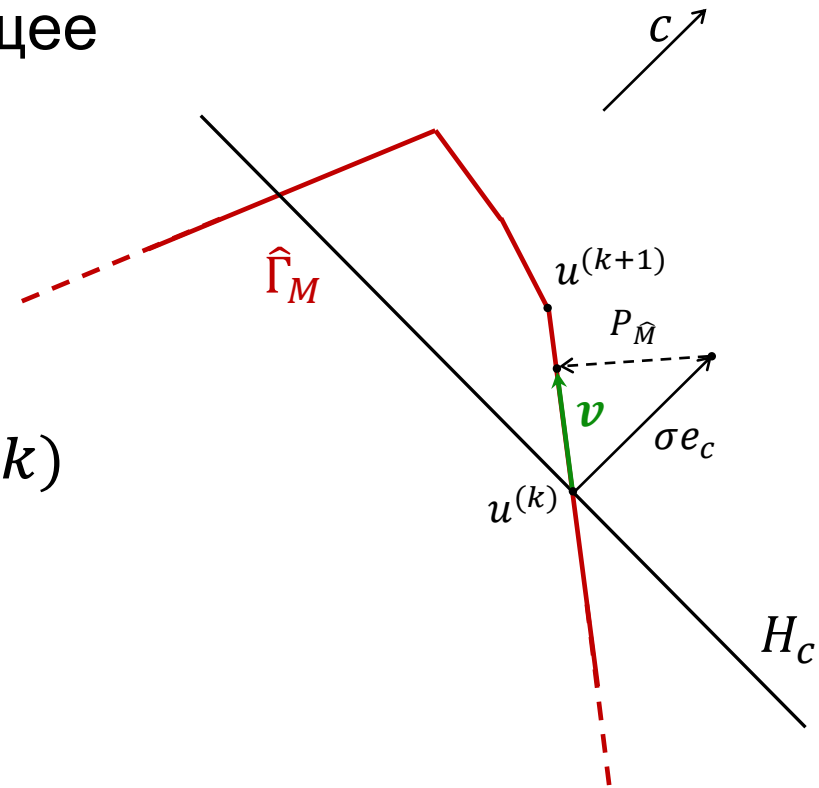






Вектор движения

- Определяет направление движения по поверхности допустимого многогранника, соответствующее максимальному увеличению значения целевой функции



$$v = P_{\widehat{M}}(u^{(k)} + \sigma e_c) - u^{(k)}$$

$P_{\widehat{M}}$ – метрическая проекция на $\widehat{M}$