Санкт-Петербургский государственный университет

Кафедра системного программирования

Мультиагентные технологии в киберфизических системах

Денис Романович Ужва, 19.М05-мм группа

Научный руководитель: д.ф.-м.н. О.Н. Граничин, профессор кафедры системного программирования

Рецензент: старший инженер ООО «Техкомпания Хуавэй» Н.В. Устюжанин

Санкт-Петербург
2021

Введение

pinterest.com

learn.g2.com

Мультиагентные технологии придумала природа — мы же их адаптируем

Цель и задачи

Цель — создание алгоритма и прототипа ПО для кластерного управления киберфизическими мультиагентными системами (МАС).
Задачи:

1. Исследовать методы управления МАС.
2. Разработать математический фреймворк для описания мультиагентных систем, формализующий явление “кластерных потоков”; исследовать с его помощью модель осцилляторов Курамото как пример киберфизической мультиагентной системы.
3. На основе данного фреймворка и метода сжатия Compressive Sensing разработать алгоритм удалённых наблюдений за МАС с синтезом управления кластерами.
4. На основе предложенного алгоритма реализовать прототип программного обеспечения для поиска кластеров в МАС.
5. Выполнить апробацию алгоритма на примере модели Курамото.

Постановка задачи управления МАС

Агент как структурная единица системы:

1. реактивность — реагирует на события во внешней среде;
2. проактивность — имеет цель и модель поведения;
3. социальность — выделяет себе подобных, общается.

Управление МАС предполагает некоторый перечень манипуляций над ней, приводящий текущее состояние системы к желаемому виду

Однако мешает *эмерджентность*

Классификация алгоритмов управления

Глобальное управление:

- одинаково для всех агентов;
- простая имплементация;
- ограниченность целей;
- хорошая изученность.

Локальное управление:

- индивидуально для каждого агента;
- сложно связать с конкретной целью;
- вариативность целей;
- также хорошая изученность.

Выводы:

- несмотря на хорошую изученность локального и глобального подходов, нельзя выделить лучший
- кластерные управляющие алгоритмы мало изучены, при этом имеют колоссальный потенциал

Фреймворк кластерных потоков

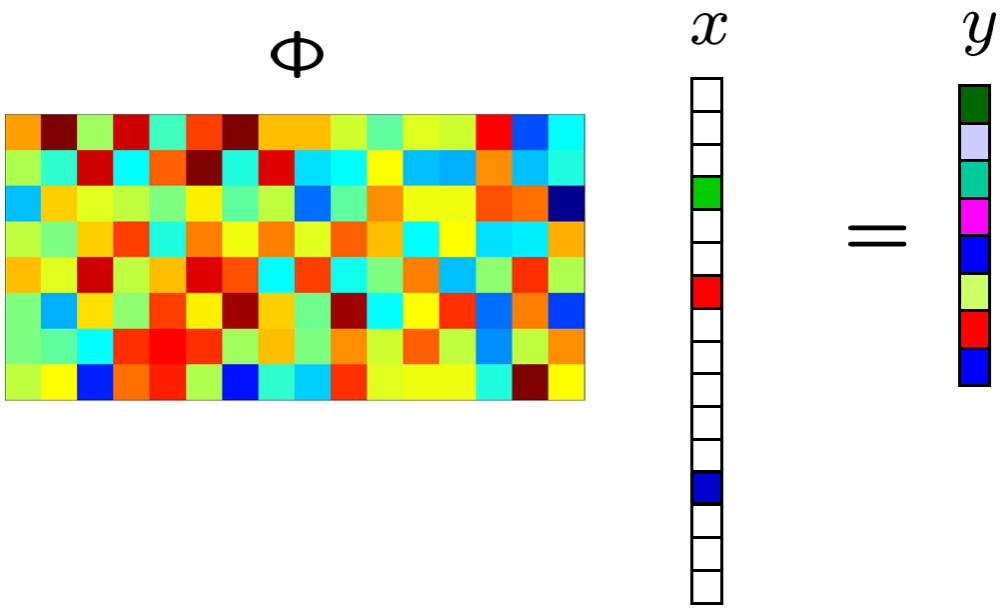
Динамика состояния агентов:

$$\dot{x}(t) = f(x(t), u(t), U_\alpha(t), \eta(t)), \tag{1}$$

- x — состояние агента;
- u — локальное управление;
- U — кластерное управление (для кластера α);
- η — помехи.

Функция $g(x(t), \nu(t))$ называется выходом агента (с помехой ν)
Кластер определяется близостью между агентами по их выходам в момент времени t

Опознавание со сжатием



informationtransfereconomics.blogspot.com

Реконструкция возможна с помощью:

- симплекс-метод;
- **метод внутренней точки**

Алгоритмы поиска кластеров

Алгоритмы кластеризации можно разделить на четыре вида:

- **иерархические** (агломеративные и дивизионные методы);
- центроидные (метод k-средних);
- кластеризация по распределению (Gaussian mixture models);
- кластеризация по плотности (DBSCAN)

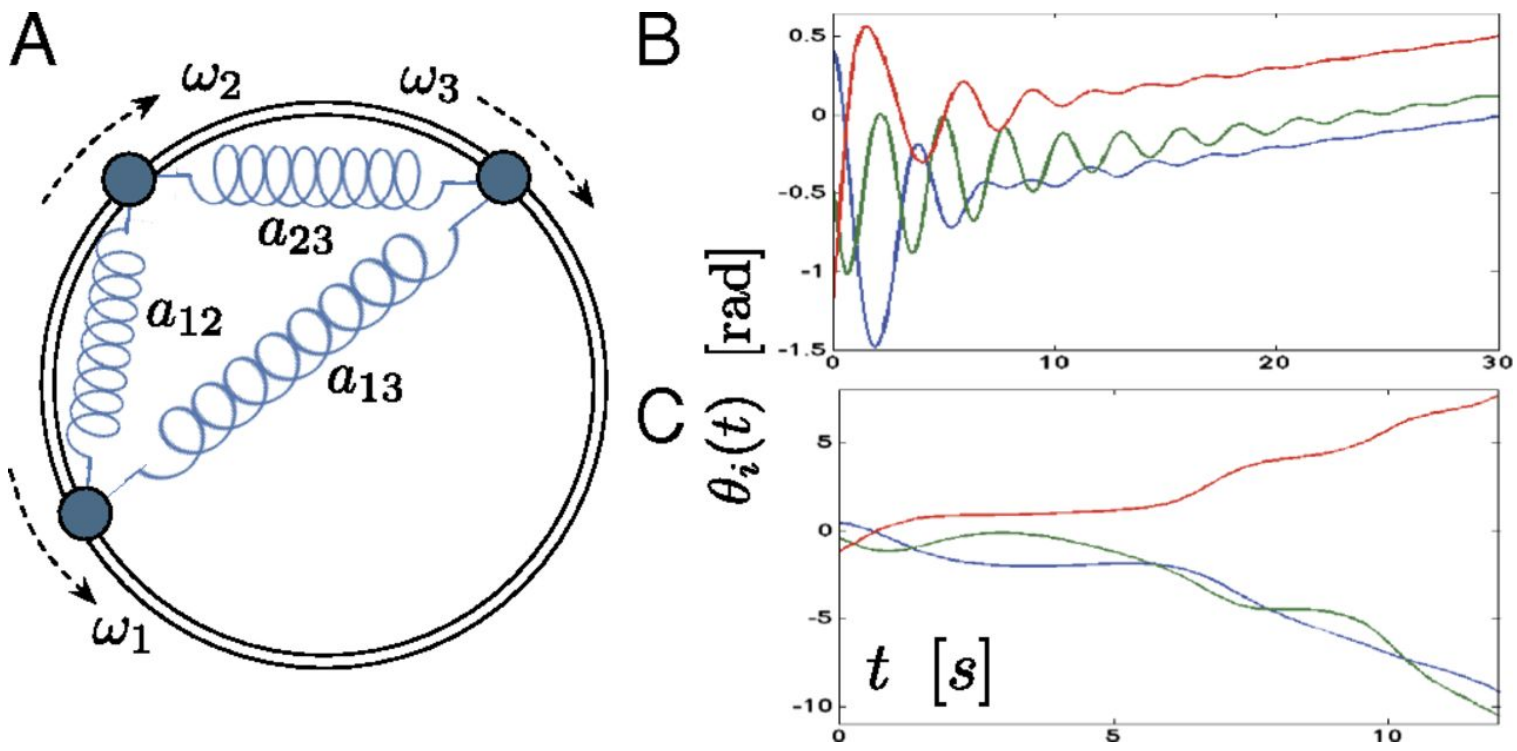
Реализация алгоритма кластерного управления с компрессией

Требования к ПО:

1. простота написания и чтения кода;
2. удобство отладки и сопровождения ПО;
3. возможность простой визуализации результатов вычислений

Технологии: Python + Jupyter Notebook + NumPy + SciPy + matplotlib + Google Colab.

Модель Курамото



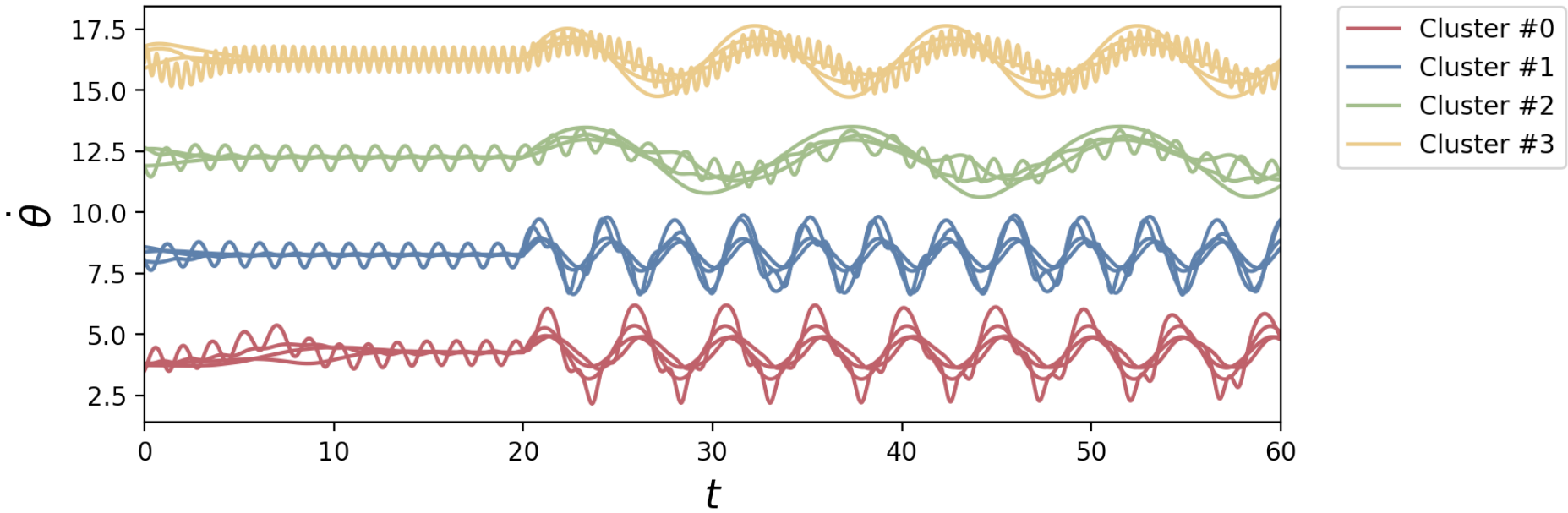
F. Dörfler 2013

Фреймворк кластерных потоков позволяет сформулировать теорему об ограничениях на параметры модели, при которых кластерная синхронизация не изменяется

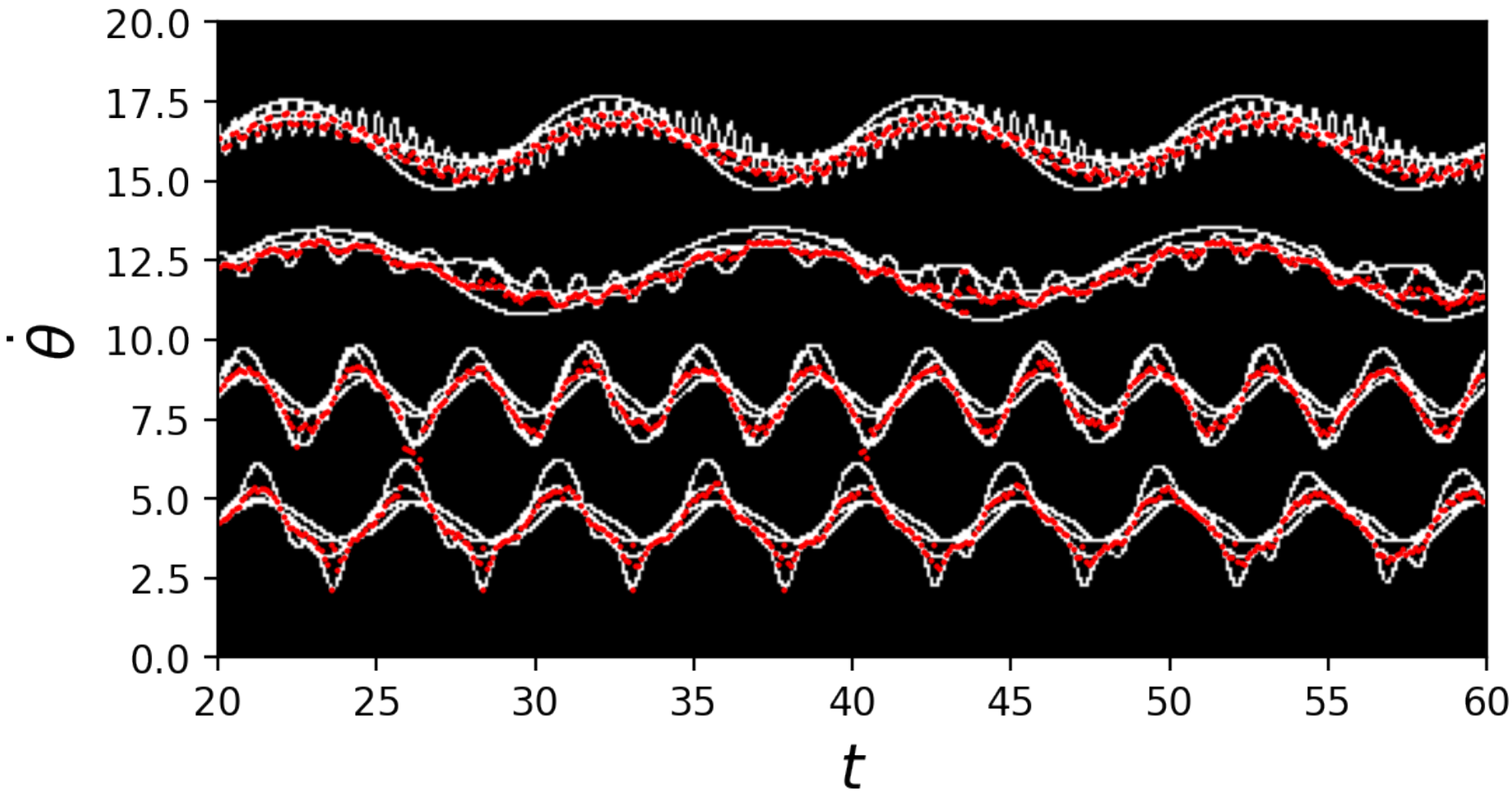
Симуляция осцилляторов

Параметры симуляции:

- 16 агентов;
- 4 искусственно созданных кластера;
- локальное управление — сумма синосув разностей фаз (модель Курамото);
- кластерное управление — синус, зависящий от времени

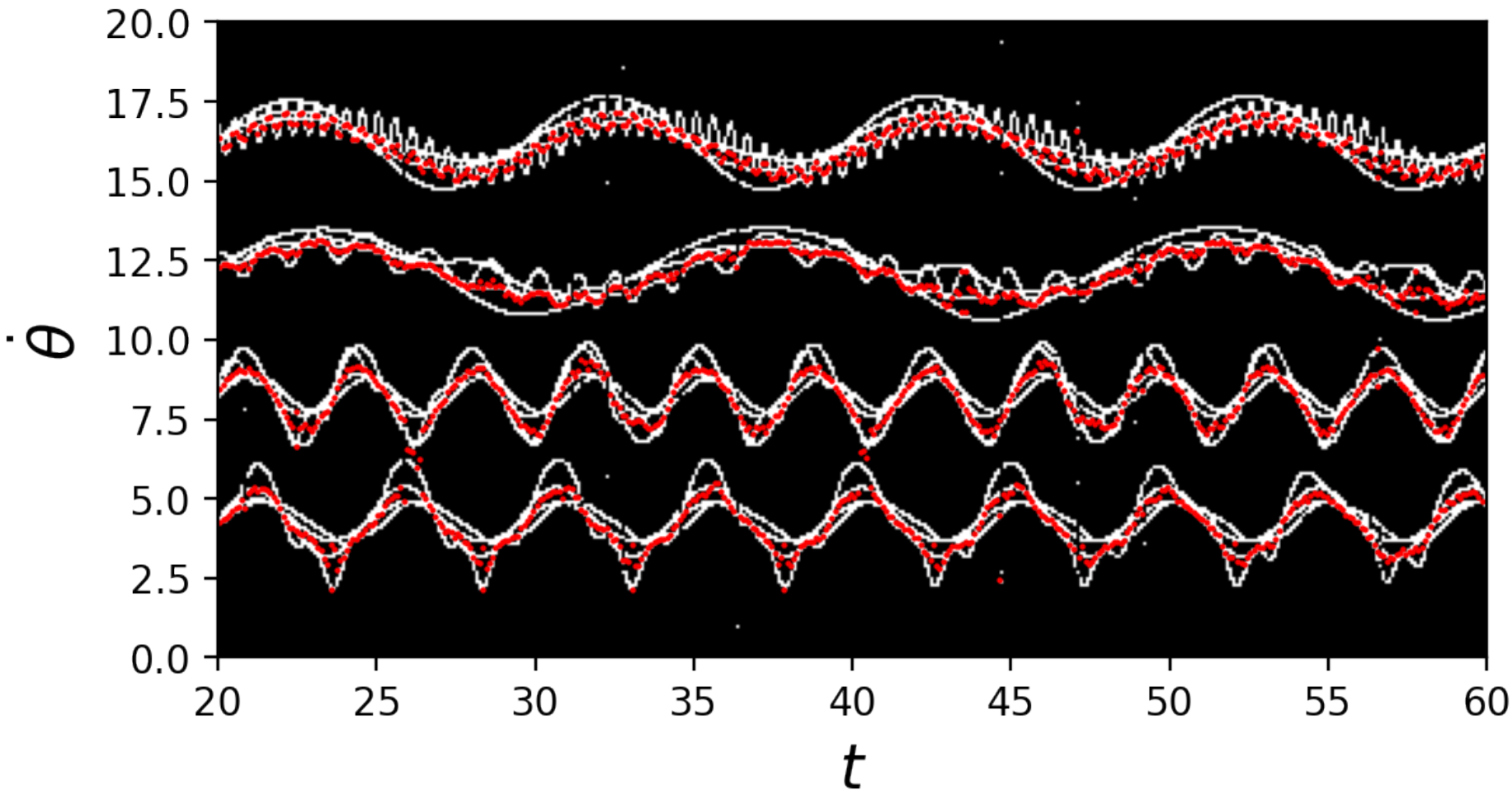


Распознавание кластеров в оригинальных данных



Шаг дискретизации по пространству: 0.1, по времени: 0.1

Распознавание кластеров в сжатых данных



Шаг дискретизации по пространству: 0.1, по времени: 0.1

Степень сжатия и точность кластеризации

Опознавание со сжатием: $y = \Phi x$, размерность y равна m, размерность x равна $p > m$. Степень сжатия

$$\gamma_c = p/m, \tag{2}$$

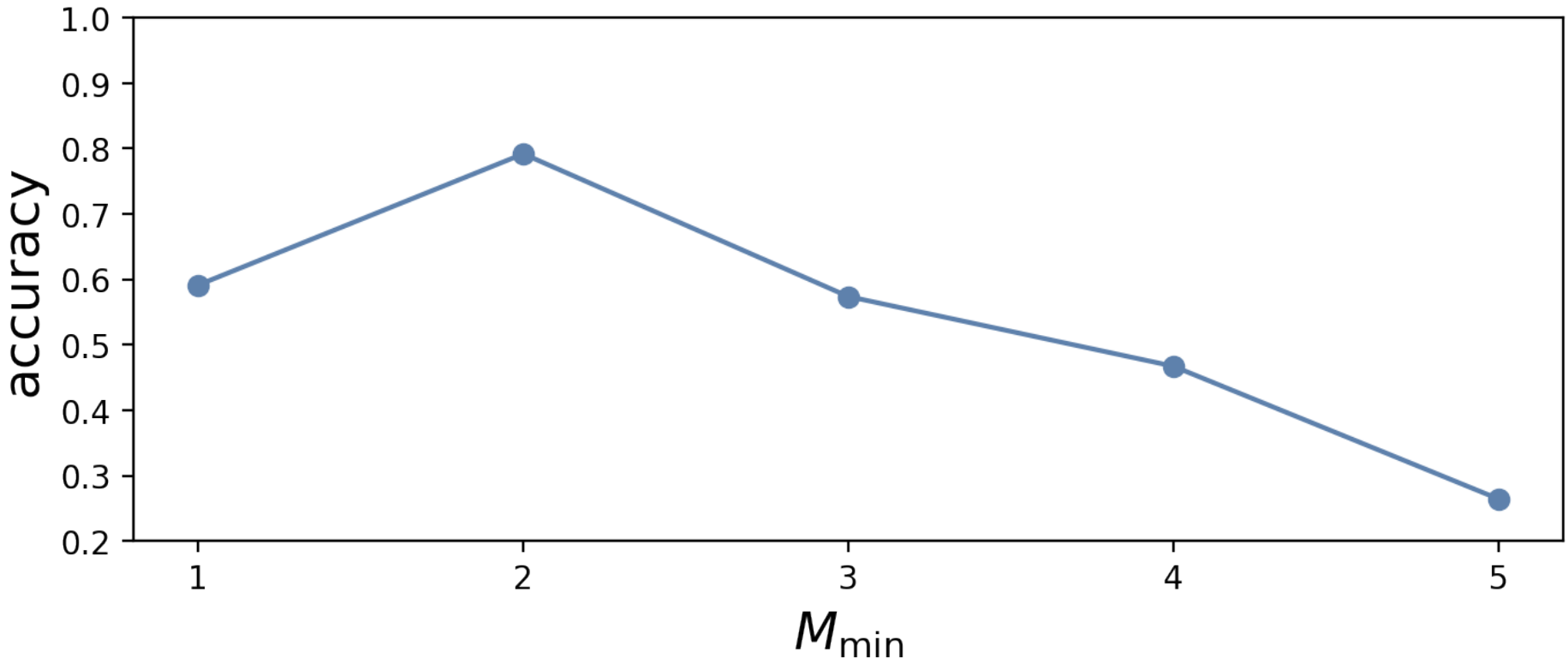
при этом $m = c \cdot s$, s — количество ненулевых элементов x.
Введём следующую переменную:

$$\chi[i] = \begin{cases} 1, & \text{если } \widehat{M}[i] = M, \\ 0, & \text{если } \widehat{M}[i] \neq M, \end{cases}$$

где i — номер наблюдения из серии (принимает значения от 1 до q), $\widehat{M}$ — предсказанное алгоритмом оценочное количество кластеров, M — истинное количество кластеров. В таком случае процентная точность по q наблюдениям

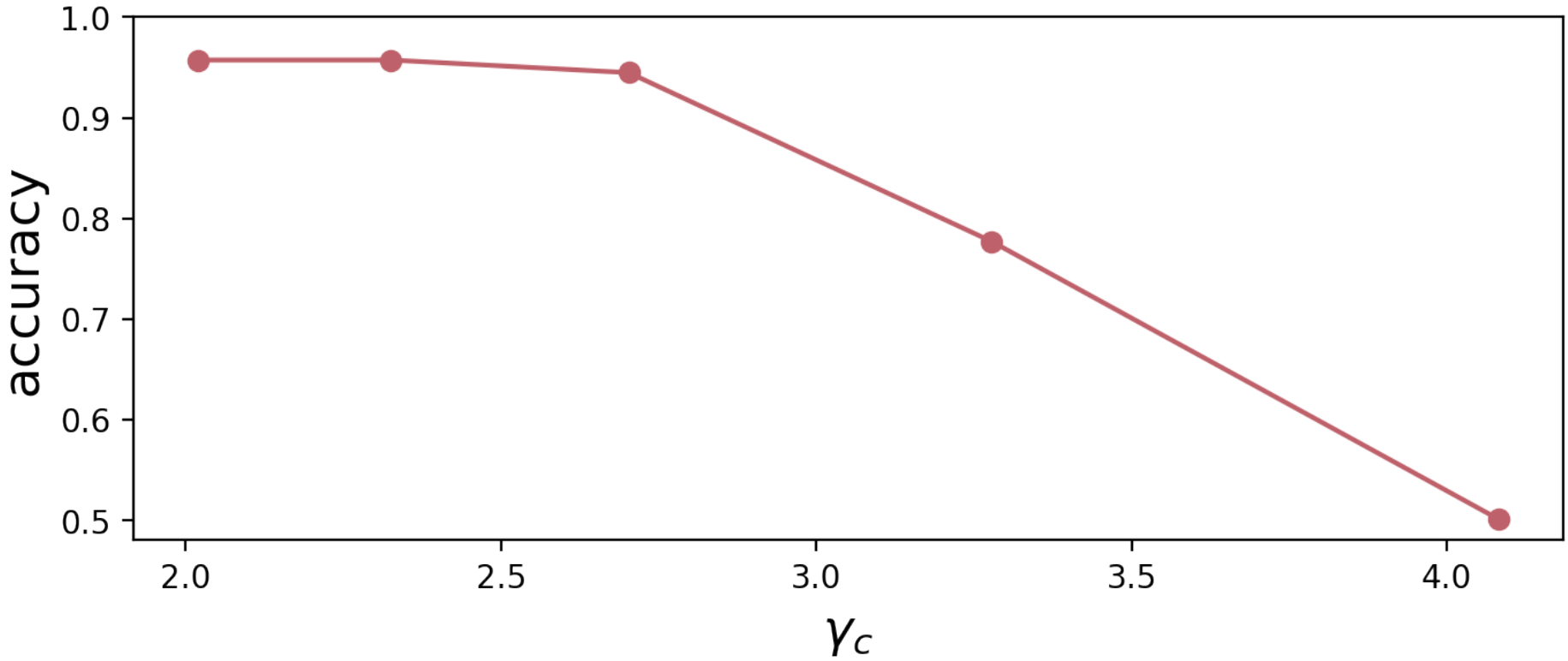
$$\epsilon = \frac{\sum_{i=1}^{q} \chi[i]}{q} \cdot 100\%. \tag{3}$$

Зависимость точности от размера минимального кластера



Точность идентификации кластеров в при $\gamma_{2.5} = 3.28$ и различных значениях минимального кластера M_{min}

Зависимость точности от степени сжатия



Точность идентификации кластеров при $M_{\min} = 2$ и различных значениях степени сжатия γ_c

Результаты

1. Выявлены преимущества и недостатки “локального” и “глобального” методов управления.
2. На основе теории динамических систем разработан математический фреймворк кластерных потоков. Сформулирована и доказана теорема об ограничениях на параметры модели Курамото необходимых для кластерной синхронизации.
3. На основе метода Compressive Sensing разработан алгоритм удалённых наблюдений за агентами в MAC с поиском кластеров. Разработан способ оценки точности алгоритма.
4. Выполнена реализация алгоритма на языке Python с использованием библиотек NumPy, SciPy, matplotlib в среде Jupyter Notebook. Симуляции развёрнуты в Google Colab.
5. На примере модели Курамото апробирован разработанный и реализованный алгоритм. При варьировании степени сжатия от 2 до 4 точность падает с 95 до 50 %.

Публикация результатов

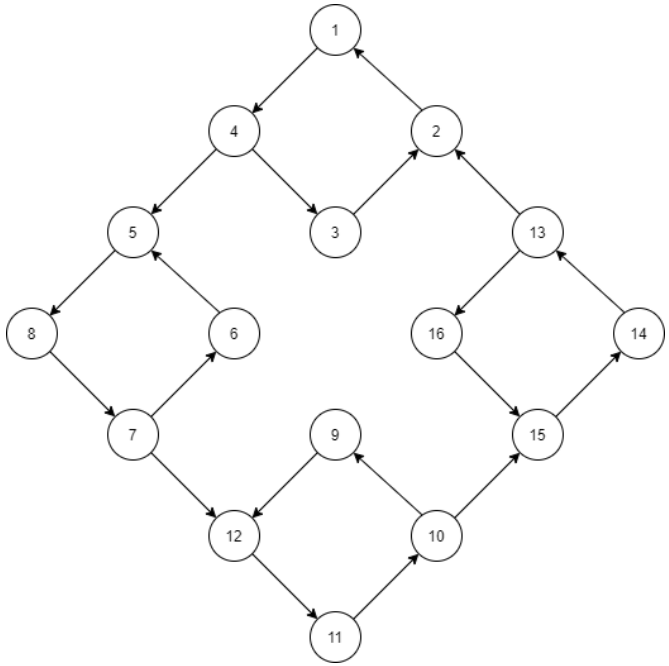
- по теоретическим результатам сделан доклад на “Восемнадцатой Национальной конференции по искусственному интеллекту“ (Россия, Москва) и опубликована статья;
- по результатам разработки алгоритма опознавания со сжатием для MAC опубликована статья в журнале Mathematics с импакт-фактором 1.747 (Q1 в JCR)

Параметры симуляции

- топология графа $\mathcal{G}$ см. ниже;
- начальные фазы $\theta_i(0)$ из равномерного распределения на окружности S^1;
- значения f_α берутся из равномерного распределения на $[0, 1]$;
- $\rho = 0.5$;
- собственные частоты w_i представлены ниже;
- значения μ_i представлены ниже.

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
4.1	4.2	4.3	4.4	8.1	8.2	8.3	8.4
w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}
12.1	12.2	12.3	12.4	16.1	16.2	16.3	16.4

μ_1	μ_5	μ_9	μ_{13}	μ_2	μ_6	μ_{10}	μ_{14}
0.375				0.75			
μ_3	μ_7	μ_{11}	μ_{15}	μ_4	μ_8	μ_{12}	μ_{16}
1.125				1.5			



Топология используемого графа

Модель Курамото

Модель с кластерным управлением:

$$\dot{\theta}_i(t) = \mu_i \mathcal{F}_\alpha(t, \overline{x}_\alpha(t)) + w_i + \rho \sum_{j=1}^{N} \Upsilon_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right).$$

Кластерное управление в симуляциях:

$$U_i = \mu_i \mathcal{F}_\alpha(t, \overline{x}_\alpha(t)) = \mu_i \sin(2\pi f_\alpha(t - 20)).$$