Многоагентное обучение с подкреплением с непрерывным пространством действий

Пшеницын А.С.
pshenitsyn.as@phystech.edu

Центр когнитивного моделирования МФТИ

2025 – Статус НИР 20.05





Мотивация



Многоагентное планирование — одна из наиболее сложных задач для навигации роботов. Существующие методы направлены на решение задачи в дискретных средах (NP-hard), потому для перехода в непрерывные состояния часто используют дискретизацию пространства, и построенные пути не являются оптимальными.

Цель работы

Разработка методов обучения с подкреплением для многоагентного планирования в непрерывных средах, способных к масштабированию и генерализации по числу агентов и сложности среды



Постановка задачи

Децентрализованный мультиагентный частично наблюдаемый процесс принятия решений - это $M = \langle S, I, (A_i), P, (\Omega_i), (O_i), (r_i), \gamma \rangle$, где:

- S - пространство состояний,
- $I = \{1, \ldots, n\}$ - агенты,
- A_i - пространство действий для агента i
- $A = \times_i A_i$ - пространство объединенных действий
- $P = P(s'|s, \vec{a})$ - функция переходов,
- Ω_i - пространство наблюдений для агента i,
- $O_i = O_i(s, a)$ - функция наблюдения для агента i,
- $r = r(s, \vec{a})$ - общая функция вознаграждения,
- γ - фактор дисконтирования

Задача - найти стратегию $\pi(O_i(s, a)) : \mathbb{E}_{s_t \sim \pi}[\sum_{t \geq 0} \gamma^t r(s_t, \vec{a_t})] \to \max_{\pi}$

Постановка задачи

В задаче планирования:

$$r_i = \begin{cases} 1, \text{если } \|x_i - goal_i\| < R_{goal} \\ -1, \text{если } \exists j: \ \|x_i - x_j\| < R_i + R_j \text{ или } dist(x_i, \text{Obstacle}) < R_i \\ 0, \text{иначе} \end{cases}$$

$x_i \in S$ - положение агента i, $goal_i \in S$ - цель агента i, R_{goal} - радиус цели, R_i - размер агента i, Obstacle $\subset S$ - множество препятствий в среде.

Метрики качества алгоритмов:

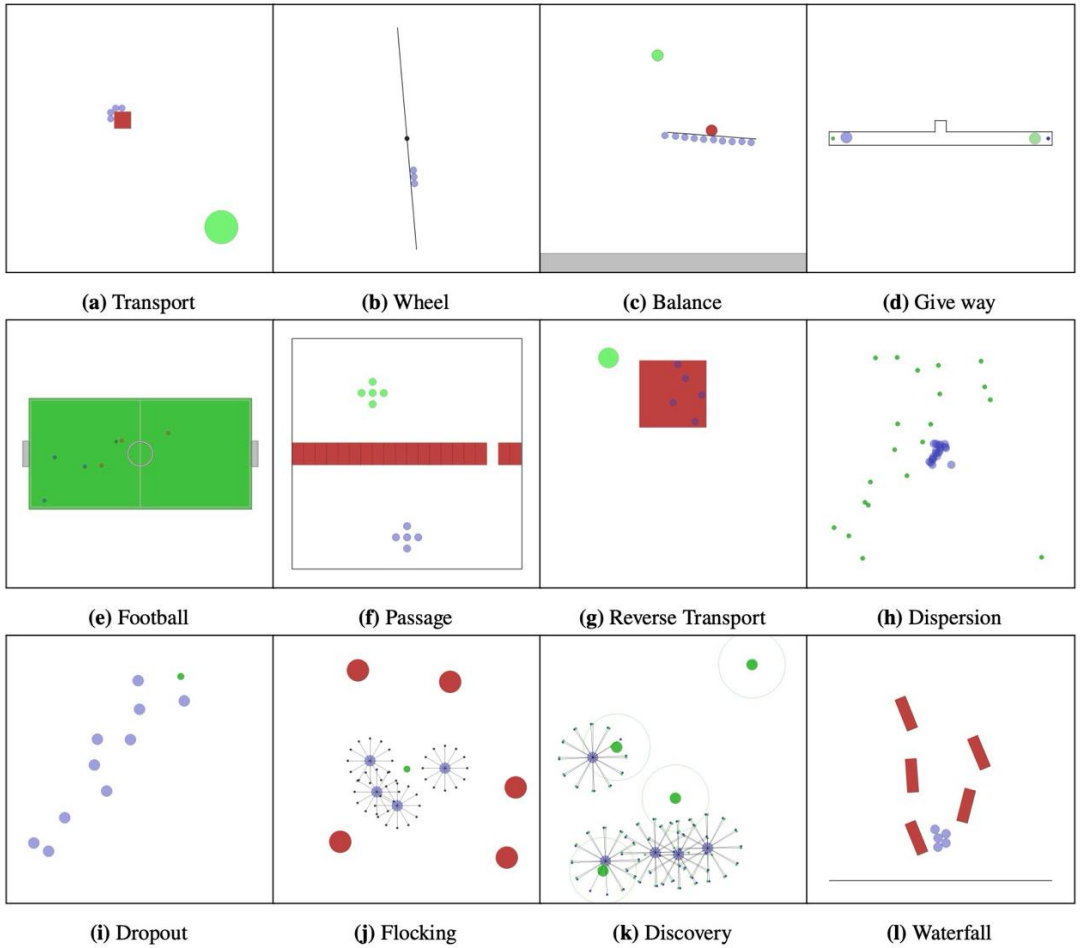
$$\begin{cases} \text{SR} = \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \mathbf{1}\{\|\vec{x}_i(T) - \vec{x}_{\text{goal}}\| \le R_{\text{goal}}\}; \\ \text{FT} = \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} t_i; \\ \text{MS} = \displaystyle\max_{i=1,\ldots,N} t_i; \\ \text{CO} = 1 - \dfrac{C}{N \times T} \end{cases}$$

Задачи

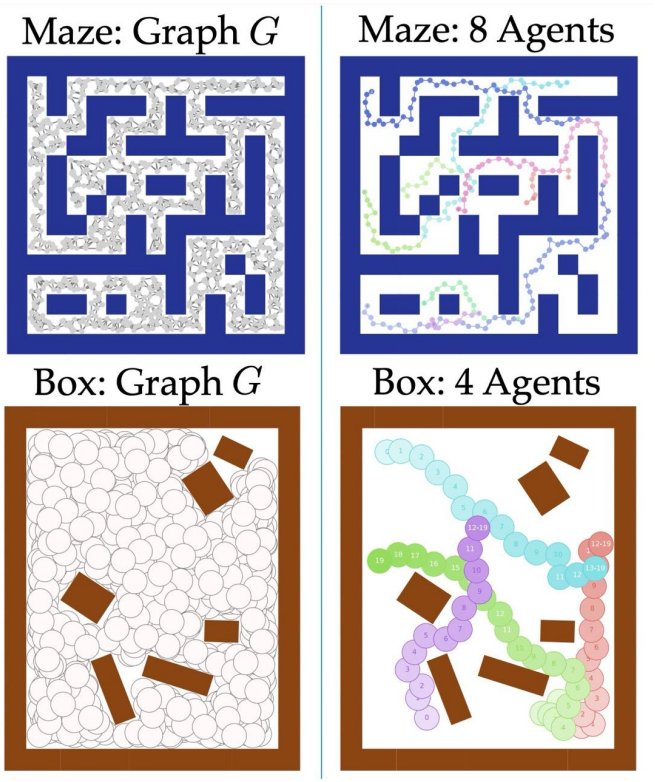
Center for Cognitive Modeling

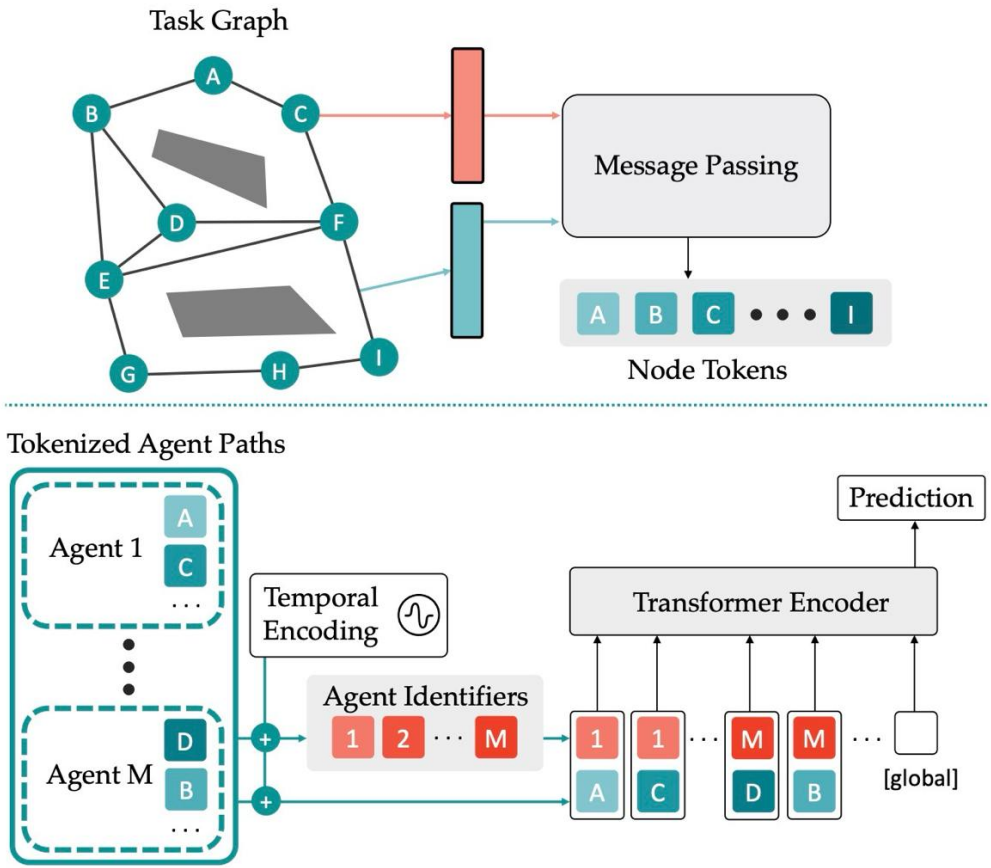
- Провести анализ существующих сред с непрерывными пространствами состояний и действий, применимых для сценариев многоагентного планирования с большим числом агентов.
- Разработать эффективную среду для задач многоагентного планирования в непрерывных пространствах состояний и действий.
- Реализовать MARL бейзлайны, а также классические алгоритмы планирования на разработанных задачах.
- Разработать новый алгоритм, улучшающий эффективность многоагентного планирования в непрерывных пространствах.

VMAS (link)


(a) Transport
(b) Wheel
(c) Balance
(d) Give way
(e) Football
(f) Passage
(g) Reverse Transport
(h) Dispersion
(i) Dropout
(j) Flocking
(k) Discovery
(l) Waterfall

Multi-Agent Planning Using Graph Transformers (link)





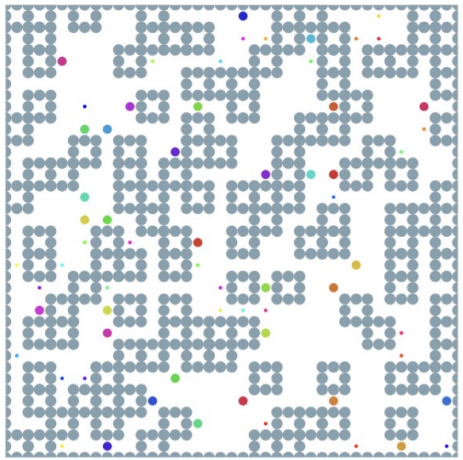
Анализ сред

Center for Cognitive Modeling

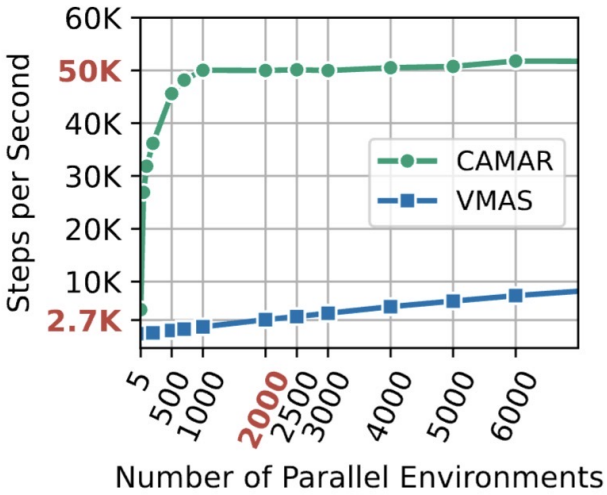
Environment / Simulator	Repository	Continuous Observations	Continuous Actions	GPU Support	Scalability >500 Agents	Partially observable	Heterogeneous agents	Performance >10K Steps/s	Python based	Procedural generation	Requires generalization	Evaluation protocols	Tests & CI	PyPi Listed
Waterworld (SISL) [13]	link	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓
RWare [20]	link	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓
RWare (Jumanji) [7]	link	✗	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓
RWare (Pufferlib) [29]	link	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✓	✓
Trash Pickup (Pufferlib) [29]	link	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✓	✓
SMAC [24]	link	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
SMACv2 [12]	link	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
SMAX (JaxMARL) [22]	link	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓
JaxNav (JaxMARL) [23]	link	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓
Nocturne [30]	link	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
POGEMA [25]	link	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
VMAS* [4]	link	✓	✓	✓	✗	✓	✓	✗ / ✓*	✓	✗	✗ / ✓*	✗	✓	✓
SMART [32]	link	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Gazebo [15]	link	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
Webots [19]	link	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
ARGoS [21]	link	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗
CAMAR (Ours)		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

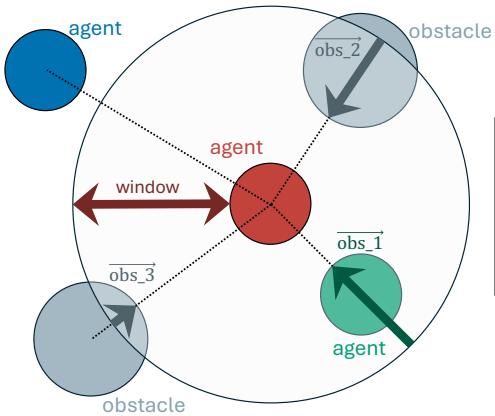
Экспериментальные результаты

$$\begin{cases} \vec{f}_i(t) = \vec{f}_i^{\,a}(t) + \sum_j \vec{f}_{ij}^{\,\text{collision}}(t) \\ \vec{v}_i(t+dt) = \vec{v}_i(t) + \dfrac{\vec{f}_i(t)}{m} \cdot dt \\ \vec{x}_i(t+dt) = \vec{x}_i(t) + \vec{v}_i(t+dt) \cdot \end{cases}$$

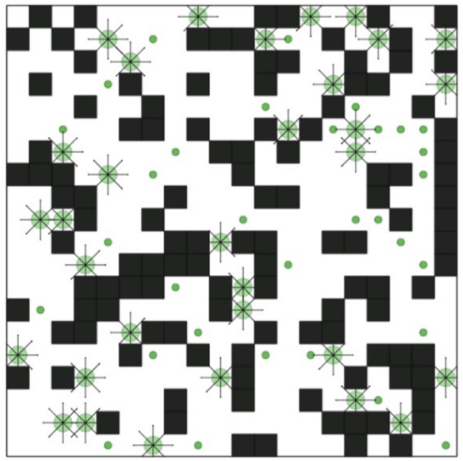


(a) CAMAR

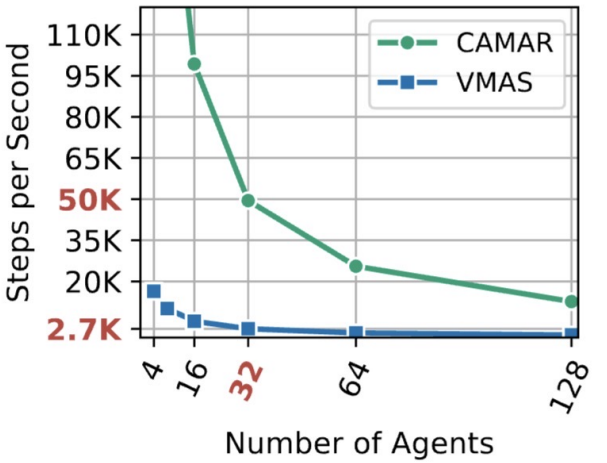




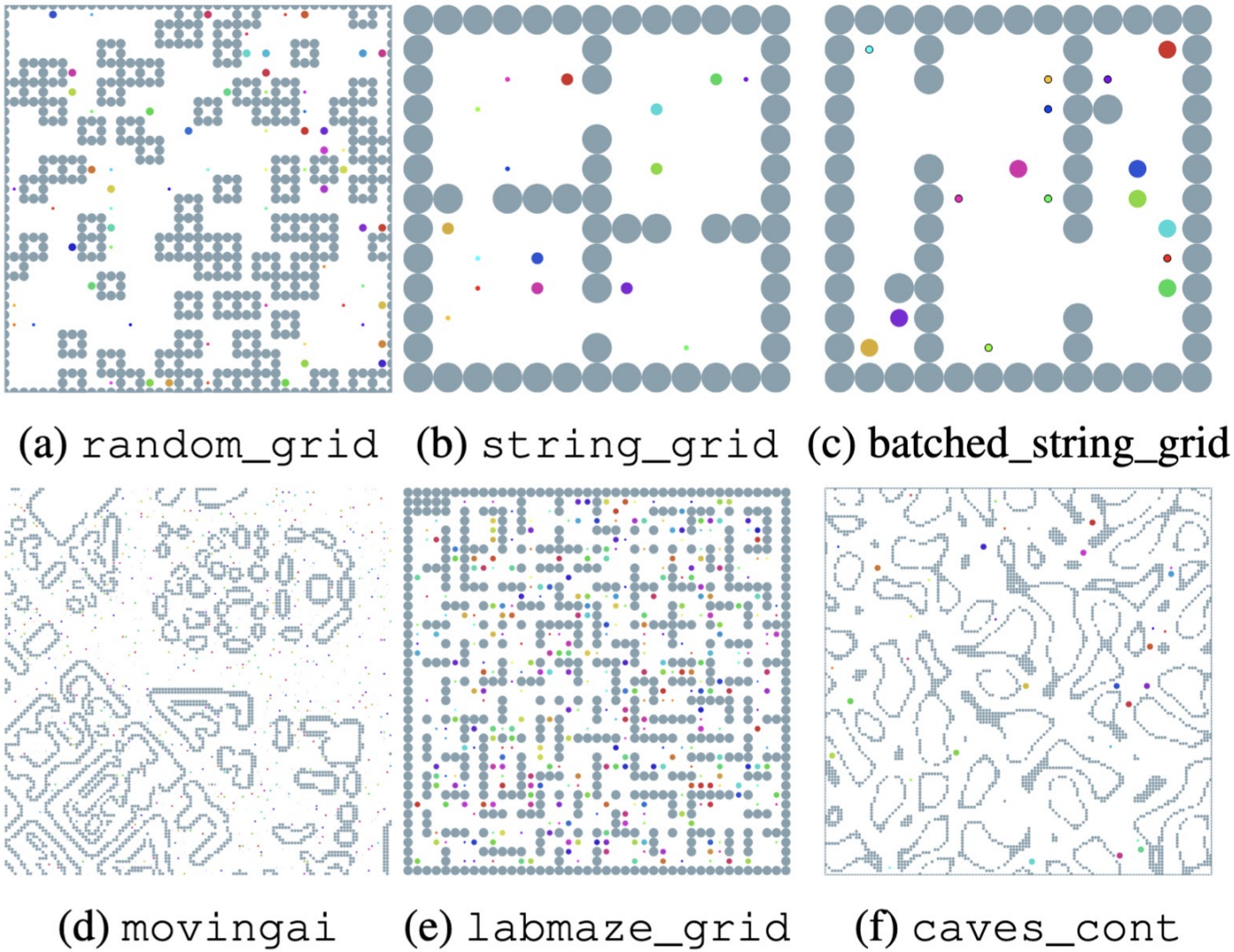
$$\text{observation} = \begin{bmatrix} \overrightarrow{obs_1} \\ \overrightarrow{obs_2} \\ \overrightarrow{obs_3} \\ \vec{0} \\ \dots \end{bmatrix}$$



(b) VMAS [4]



Карты для планирования



Экспериментальные результаты

Center for Cognitive Modeling

	random_grid				labmaze_grid			
Algorithm	SR ↑	FT ↓	MS ↓	CO ↑	SR ↑	FT ↓	MS ↓	CO ↑
IPPO	0.479±0.003	1757±14	151.0±0.3	0.932±0.002	0.330±0.003	2155±16	155.2±0.3	0.929±0.001
MAPPO	0.497±0.003	1770±14	143.8±0.4	0.924±0.002	0.300±0.003	2303±16	155.3±0.3	0.935±0.001
IDDPG	0.299±0.003	2182±16	155.1±0.2	0.970±0.001	0.337±0.003	2329±17	153.6±0.3	0.949±0.001
MADDPG	0.024±0.001	3024±19	160.0±0.0	0.880±0.002	0.005±0.000	2996±18	160.0±0.0	0.849±0.001
ISAC	0.232±0.003	2462±17	157.1±0.2	0.984±0.001	0.171±0.002	2707±18	159.6±0.1	0.957±0.001
MASAC	0.046±0.001	2782±18	160.0±0.0	0.814±0.002	0.032±0.001	2865±18	160.0±0.0	0.806±0.002
RRT+PD	0.428±0.014	2496±137	159.6±0.5	0.795±0.012	0.542±0.022	2105±136	144.5±3.3	0.593±0.018

Дальнейшая работа

Center for Cognitive Modeling

- Покрыть код среды модульными и интеграционными тестами для обеспечения корректности и стабильности разработки
- Настроить систему CI/CD для поддержки автоматизированной open-source разработки
- Разработать протоколы обучения и тестирования, обеспечивающие унифицированную и воспроизводимую оценку методов и алгоритмов
- Опубликовать разработанную среду в виде библиотеки на платформе PyPI
- Приступить к разработке метода с интегрированным RRT (RRT-based goal guidance)

Спасибо за внимание!

cogmodel.mipt.ru

pshenitsyn.as@phystech.edu