МАТЕРІАЛИ III МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ

РОЗВИТОК НАУКОВОЇ ДУМКИ ПОСТІНДУСТРІАЛЬНОГО СУСПІЛЬСТВА: СУЧАСНИЙ ДИСКУРС

28 КВІТНЯ 2023 РІК

М. ЛЬВІВ, УКРАЇНА

OPEN ACCESS

DOI 10.36074/mcnd-28.04.2023
ISBN 978-617-8126-29-2

МАТЕРІАЛИ
III МІЖНАРОДНОЇ
НАУКОВОЇ КОНФЕРЕНЦІЇ

Міжнародний Центр Наукових Досліджень

РОЗВИТОК НАУКОВОЇ ДУМКИ ПОСТІНДУСТРІАЛЬНОГО СУСПІЛЬСТВА: СУЧАСНИЙ ДИСКУРС

28 КВІТНЯ 2023 РІК
м. Львів, Україна

Вінниця, Україна
«Європейська наукова платформа»
2023

УДК 001 (08)
Р 64

https://doi.org/10.36074/mcnd-28.04.2023

Організація, від імені якої випущено видання:
ГО «Міжнародний центр наукових досліджень»

Голова оргкомітету: Рабей Н.Р.

Верстка: Зрада С.І.
Дизайн: Бондаренко І.В.

Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 58 від 17.01.2023).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Р 64 **Розвиток наукової думки постіндустріального суспільства: сучасний дискурс**: матеріали III Міжнародної наукової конференції, м. Львів, 28 квітня, 2023 р. / Міжнародний центр наукових досліджень. — Вінниця: Європейська наукова платформа, 2023. — 192 с.

ISBN 978-617-8126-29-2
DOI 10.36074/mcnd-28.04.2023

Викладено матеріали учасників III Міжнародної спеціалізованої наукової конференції «Розвиток наукової думки постіндустріального суспільства: сучасний дискурс», яка відбулася 28 квітня 2023 року у місті Львів.

УДК 001 (08)

© Колектив учасників конференції, 2023
© ГО «Європейська наукова платформа», 2023
© ГО «Міжнародний центр наукових досліджень», 2023

ISBN 978-617-8126-29-2

СЕКЦІЯ XI.
ТЕХНОЛОГІЇ ЛЕГКОЇ ТА ДЕРЕВООБРОБНОЇ ПРОМИСЛОВОСТІ

ВИЗНАЧЕННЯ РАЦІОНАЛЬНИХ ПАРАМЕТРІВ ДУБЛЮВАННЯ ТЕКСТИЛЬНИХ МАТЕРІАЛІВ
Гавриш Л. **86**

СЕКЦІЯ XII.
ЕКОЛОГІЯ ТА ТЕХНОЛОГІЇ ЗАХИСТУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА

ПОШИРЕННЯ АДВЕНТИВНИХ РОСЛИН НА ТЕРНОПІЛЬЩИНІ
Адамів С.С. **89**

СЕКЦІЯ XIII.
КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКІВ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ ДЛЯ VR-ШОЛОМІВ OCULUS
Вередін М.О. **94**

ПРОГРАМНЕ ЗАБЕСПЕЧЕННЯ ДЛЯ ЗАПУСКУ СЕРЕДОВИЩА ТА КЕРУВАННЯ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ У KUBERNETES КЛАСТЕРІ
Гнідий П.О. **98**

СЕКЦІЯ XIV.
СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ

ПОБУДОВА МОДЕЛІ ОЦІНКИ ЙМОВІРНОСТІ ДЕФОЛТУ ПОЗИЧАЛЬНИКІВ НА ОСНОВІ ЯКІСНИХ ПОКАЗНИКІВ
Савіна С.С., Водзянова Н.К. **103**

СЕКЦІЯ XV.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

ГНУЧКА ВИРОБНИЧА СИСТЕМА НА ОСНОВІ PYTHON ІЗ КЕРУВАННЯМ ЗАМОВЛЕННЯМИ ДЛЯ МУЛЬТИАГЕНТНОГО ПІДКРІПЛЕННЯ НАВЧАННЯ
Койдан А.А. **106**

МОДЕЛЮВАННЯ ПОВТОРЮВАНИХ ПОНЯТЬ У НЕЗБАЛАНСОВАНИХ ПОТОКАХ ДАНИХ
Холодов С.Є. **112**

СЕКЦІЯ XV.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

ГНУЧКА ВИРОБНИЧА СИСТЕМА НА ОСНОВІ PYTHON ІЗ КЕРУВАННЯМ ЗАМОВЛЕННЯМИ ДЛЯ МУЛЬТИАГЕНТНОГО ПІДКРІПЛЕННЯ НАВЧАННЯ

Койдан Анастасія Андріївна
здобувач вищої освіти, Факультет інформаційних
радіотехнологій та технічного захисту інформації
Харківський національний університет радіоелектроніки, Україна

Науковий керівник: Кузьомін Олександр Якович
доктор технічних наук, професор, професор кафедри інформатики,
Засновник Асоціації випускників ХНУРЕ
Харківський національний університет радіоелектроніки, Україна

Гнучка виробнича система (ГВС) - це комплексна система, що має за мету забезпечити гнучкість та ефективність у виробничих процесах.

Сучасний ринок вимагає від підприємств негайних відгуків на зміни попиту, а також більшої гнучкості виробництва та швидкості реагування на замовлення.

У цьому контексті, мультиагентне підкріплене навчання (МАПН) може забезпечити ефективну реалізацію гнучкої виробничої системи, де агенти взаємодіють між собою та з різними компонентами системи з метою максимізації певних критеріїв.

Python, як потужна та універсальна мова програмування, може бути використана для реалізації ГВС на основі МАПН. Зокрема, Python надає широкі можливості для розробки систем керування замовленнями, що є необхідним компонентом гнучкої виробничої системи.

Розроблення такої системи на основі Python може допомогти підприємствам забезпечити більш точне планування та управління замовленнями, а також зменшити час виробництва та затрати на нього.

Гнучкі виробничі системи, що базуються на програмуванні з використанням мови Python, стають все більш популярними в сучасній промисловості. Ці системи дозволяють швидко і легко змінювати параметри виробництва та пристосовуватися до змінних вимог ринку.

Управління замовленнями є важливою складовою гнучкої виробничої системи, що дозволяє підтримувати ефективний рівень виробництва і задовольняти потреби клієнтів. Використання мультиагентного підкріплення навчання дозволяє оптимізувати процес прийняття рішень у системі управління замовленнями шляхом навчання агентів взаємодіяти між собою і з виробничим процесом[1].

Проблема моделювання

Проблема моделювання виникає тоді, коли математична модель не відображає реальних процесів, які вона намагається описати. Це може бути спричинено недостатньою точністю чи складністю моделі, неправильним вибором параметрів,

або недостатнім розумінням взаємодії компонентів системи.

Також проблема моделювання може виникнути через недостатню кількість даних для побудови достовірної моделі або через недостатність знань про фізичні процеси, що відбуваються в системі.

Крім того, складність моделювання може збільшуватися при взаємодії багатьох компонентів системи, що може привести до складності у підборі параметрів та узгодженні взаємодії між компонентами.

Проблеми моделювання можуть призвести до неточностей у прогнозуванні та в управлінні процесами. Це може мати негативні наслідки для підприємств, де точність та ефективність виробничих процесів мають вирішальне значення.

У зв'язку з цим, важливо розробляти моделі, які відображають реальні процеси та враховують складність взаємодії компонентів системи[3].

Як вирішується проблема моделювання

Проблема моделювання може бути вирішена за допомогою різних підходів та методів. Основні з них наступні[2]:

–Використання більш точних та складних моделей: в деяких випадках простіші моделі можуть бути недостатньо точними, тому необхідно використовувати більш складні та точніші моделі.

–Використання більш точних даних: якщо даних недостатньо, модель може бути неправильною або недостатньо точною. Тому важливо збирати більше даних та забезпечувати їх якість.

–Використання алгоритмів машинного навчання: ці алгоритми можуть допомогти побудувати більш точні та складні моделі, а також дозволяють автоматично підбирати параметри моделі на основі даних.

–Використання експертних знань: експерти з досвідом в галузі можуть допомогти підібрати правильні параметри моделі та врахувати особливості галузі.

–Використання аналізу відхилень: аналіз відхилень може допомогти знайти проблеми з моделлю та підібрати правильні параметри.

–Узагальнюючи, вирішення проблеми моделювання вимагає поєднання різних методів та підходів, залежно від конкретної ситуації.

Засоби моделювання

Існує безліч засобів моделювання, які використовуються в різних галузях науки та техніки. Нижче перераховані деякі з них[4]:

–Математичні моделі: це формалізовані математичні описи процесів або систем, які можуть бути аналізовані та розуміні систематично.

–Фізичні моделі: це спрощені фізичні описи реальних процесів або систем, які можуть бути використані для експериментування та аналізу.

–Комп'ютерні моделі: це програмні засоби, які моделюють різні процеси та системи, використовуючи комп'ютерну технологію.

–Статистичні моделі: це моделі, які використовують статистичні методи для аналізу даних та розуміння процесів.

–Системні динамічні моделі: це моделі, які використовуються для аналізу складних систем, де взаємодія між різними елементами системи може бути динамічною.

–Моделі на основі імітації: це моделі, які імітують реальний світ, щоб досліджувати його різні властивості.

–Аналітичні моделі: це моделі, які використовують аналітичні методи для аналізу та розуміння процесів.

Ці засоби моделювання можуть бути використані окремо або поєднані для

отримання більш точних та повних результатів.

Опис завдання

Використання потужностей повністю автоматизованої лінії виробництва кузовів залежить від окремих робочих етапів на станції, які відбуваються в запланований час циклу. Системи збору виробничих даних (PDA) збирають інформацію про стан роботів у реальному часі. Однак на станціях часто виявляються різні статистично значущі неполадки. На жаль, несправності є неминучими проблемами на виробництві[5].

Наприклад, збій або помилкові функції робота призводять до збільшення витрат на виробництво.

Така поведінка може навіть спричинити тривалий період простою, доки інженери не застосують цілеспрямовані контрзаходи. Інші затримки тривають недовго і реєструються системами КПК.

Для цієї роботи використовується моделювання продуктової лінії.

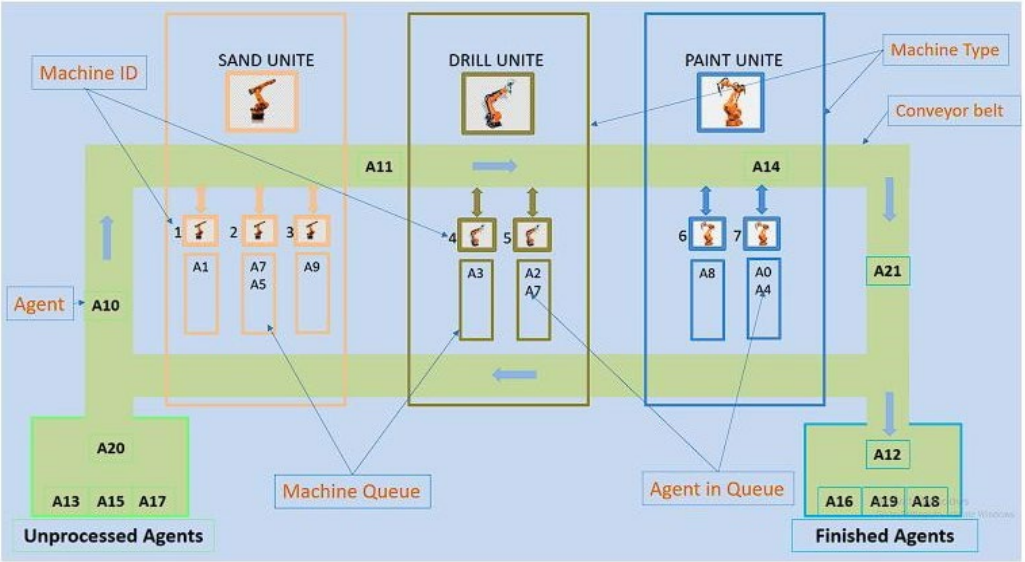


Рис. 1. **Моделювання продуктової лінії**

Під час симуляції ми створюємо функцію, яка генерує заздалегідь визначену кількість агентів.

Після цього для кожного агента зі списку завдань призначається випадкова кількість завдань (від 1 до n), і цей список завдань містить певну кількість завдань, наприклад «ПІСКУВАННЯ», «СВЕРДЛЕННЯ», «ФАРБУВАННЯ».

Більше того, випадковий час також додається разом із агентом, і це час для виконання завдань на відповідних машинах. Функція генерації агента приймає три аргументи: номер замовлення, список завдань і список пріоритетів[6].

Потім функція виконується до тих пір, поки задана кількість замовлень не буде виконана. Спочатку додайте безперервний ідентифікатор для агента, а потім час (час прибуття агента), який випадковим чином вибирається з будь-якого часу. Після цього створюється список завдань для агента, який є уніфікованим випадковим завданням із заданого списку завдань. Нарешті, для кожного агента додається випадкова властивість пріоритету, наприклад, «Істинний» або «Хибний». Якщо значення «False», відповідний агент вважається звичайним агентом, інакше він є пріоритетним агентом.

Пріоритет також має три типи, які ілюструють тип пріоритету, наприклад

високий, середній або низький. Після завершення створення кожного агента він збільшується лічильник агентів і повертає агента, доки не буде завершено створення всіх агентів.

Моделювання середовища продукту

Проект складається з двох файлів – main.py і prduct_env.py. Файл Product_env.py містить код, який імітує робоче середовище.

```
main.py   prduct_env.py
import json
from enum import Enum
import random
import collections
import numpy as np
import csv


class SimParams:
    _instance = None
    N_ORDER = 0

    MAX_NUM_OF_AGENT_GEN = 3

    AGN_GEN_INTRVL_MIN = 0
    AGN_GEN_INTRVL_MAX = 2

    AGENT_ID = 0
    TOTAL_AGENT_COMPLETED = 0

    SIM_TIME = 1

    ALL_FINISHED_AGENTS = []

    @classmethod
    def __new__(cls):
        if cls._instance is None:
            cls._instance = cls()
        return cls._instance
```

Рис. 2. **Код, який імітує робоче середовище**

Навчання з підкріпленням за допомогою ray

Файл main.py містить навчання з посиленням за допомогою бібліотеки ray python.

```
main.py   prduct_env.py
import prduct_env
import ray
from ray.rllib.agents.ppo import PPOTrainer, DEFAULT_CONFIG
from ray import tune
import pprint
ray.shutdown()
ray.init(ignore_reinit_error=True)
config = DEFAULT_CONFIG.copy()
config['env'] = prduct_env
config['framework'] = 'tf2'
stop = {
    'timesteps_total': 50000
}
analysis = tune.run('PPO', config=config, stop=stop, checkpoint_at_end=True)
```

Рис. 3. **Навчання з посиленням**

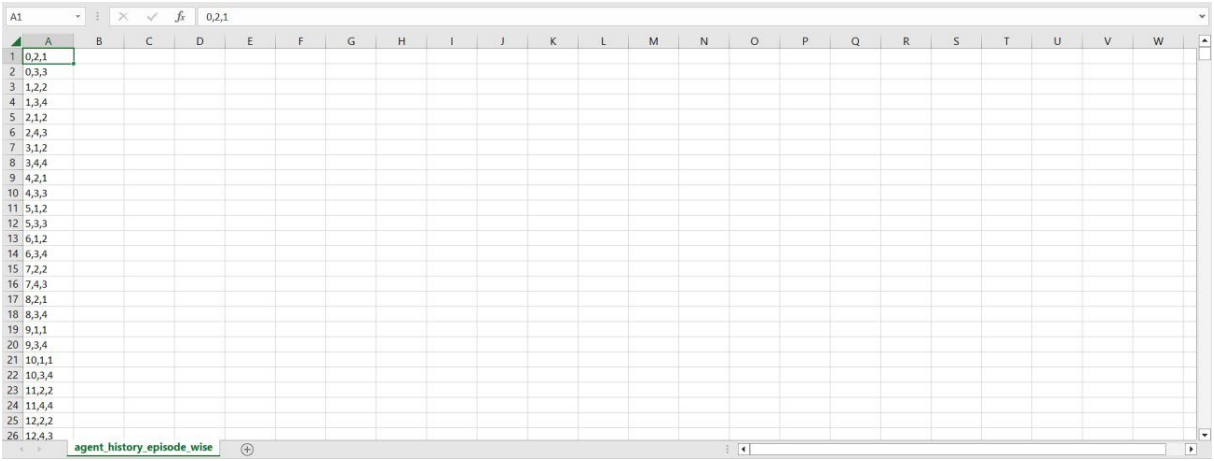
Для подальшого розвитку було вирішено впровадити в проект методи

Q-learning.

```
class Q:
    def __init__(self):
        self.gamma=0.95
        self.alpha=0.05
        self.state={}


def run_model(self, silent=1):
    self.plr.prev_state = self.plr.curr_state[:-2] + (self.plr.dx, self.plr.dy)
    self.plr.curr_state = tuple(self.plr.get_features(self.plr.x, self.plr.y)) + (
        self.plr.dx, self.plr.dy)

    if not silent:
        print(self.plr.prev_state)
        print(self.plr.curr_state)

    r = self.plr.reward

    if self.plr.prev_state not in self.state:
        self.state[self.plr.prev_state] = 0

    nvec = []
    for i in self.plr.actions:
        cstate = self.plr.curr_state[:-2] + (i[0], i[1])
        if cstate not in self.state:
            self.state[cstate] = 0
        nvec.append(self.state[cstate])

    nvec = max(nvec)
```

Рис. 4. **Q-learning**

Ми могли спостерігати ефективність агентів після завершення навчання моделі у відповідному файлі csv.



A
0,2,1
0,3,3
1,2,2
1,3,4
2,1,2
2,4,3
3,1,2
3,4,4
4,2,1
4,3,3
5,1,2
5,3,3
6,1,2
6,3,4
7,2,2
7,4,3
8,2,1
8,3,4
9,1,1
9,3,4
10,1,1
10,3,4
11,2,2
11,4,4
12,2,2
12,4,3

Рис. 5. **Файл csv**

Висновок

Цей короткий звіт містить модель для можливого використання для вирішення проблеми. Наступними кроками буде навчання цієї мережі на відповідному наборі даних друкованих текстових зображень із рукописними анотаціями, його тестування та застосування до існуючого набору даних проекту.

Гнучка виробнича система на основі Python з керуванням замовленнями для мультиагентного підкріплення навчання є потужним інструментом для покращення ефективності та продуктивності виробничих процесів. Вона дозволяє

автоматизувати процеси виробництва, зменшуючи час та зусилля, необхідні для виконання завдань.

Ця система забезпечує підтримку роботи мультиагентних систем, що розроблені для покращення продуктивності виробництва та управління ресурсами. Вона забезпечує ефективний обмін інформацією між всіма агентами та забезпечує розподіл завдань між ними, що дозволяє знизити час на виконання процесів та зменшити кількість помилок.

Крім того, гнучка виробнича система на основі Python дозволяє вирішувати проблеми на різних рівнях управління, включаючи планування виробництва, контроль якості, управління запасами та логістику. Вона також забезпечує зв'язок між виробництвом та продажами, що дозволяє підвищити ефективність роботи в цілому.

Отже, гнучка виробнича система на основі Python з керуванням замовленнями для мультиагентного підкріплення навчання є важливим інструментом для підвищення ефективності та продуктивності виробництва. Її впровадження дозволяє зменшити час та зусилля, необхідні для виконання завдань, та покращити управління ресурсами, що в свою чергу допомагає знизити витрати та підвищити якість продукції.

Список використаних джерел:

1. Liu, Y., Wang, J., Li, K., & Li, X. (2020). A flexible manufacturing system based on multi-agent reinforcement learning. Journal of Manufacturing Systems, 54, 269-280.
2. Zhang, X., Xu, H., Xu, J., & Lu, Y. (2019). Research on flexible manufacturing system based on multi-agent and intelligent scheduling. Journal of Intelligent Manufacturing, 30(3), 1089-1099.
3. Huang, H., Chen, J., & Sun, M. (2018). A hybrid model of intelligent scheduling in flexible manufacturing system using multi-agent reinforcement learning and swarm intelligence. Journal of Manufacturing Systems, 49, 30-40.
4. Xu, J., Lu, Y., Liu, H., & Zhang, X. (2018). Multi-agent-based modeling and simulation of flexible manufacturing system with hybrid manufacturing processes. IEEE Access, 6, 61361-61373.
5. Wang, Q., He, Y., & Sun, D. (2017). A multi-agent system for scheduling in flexible manufacturing systems. Computers & Industrial Engineering, 111, 524-533.
6. Shi, Y., Wei, S., & Guo, S. (2016). Intelligent scheduling in flexible manufacturing system based on multi-agent reinforcement learning. Computers & Industrial Engineering, 99, 175-184.