**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

На правах рукопису

**ДЬЯКОВ СЕРГІЙ ОЛЕКСАНДРОВИЧ**

УДК 004.896

**ДИНАМІЧНЕ КЕРУВАННЯ РУХОМИМИ ОБ'ЄКТАМИ В УМОВАХ НЕВИЗНАЧЕНОСТІ В РЕАЛЬНОМУ ЧАСІ**

**Спеціальність: 05.13.07 – Автоматизація процесів керування**

**ДИСЕРТАЦІЯ**

на здобуття вченого ступеню кандидата технічних наук

Науковий керівник:

Ямпольський Леонід Стефанович,

кандидат технічних наук, професор

Київ – 2015

ЗМІСТ

[Список скорочень 4](#_Toc434784503)

[Вступ 5](#_Toc434784504)

[1. Транспортна система ГВС як об'єкт динамічного керування 8](#_Toc434784505)

[1.1. Роль та місце транспортної системи у структурі ГВС 8](#_Toc434784506)

[1.1.1. Узагальнена структура ГВС 8](#_Toc434784507)

[1.1.2. Типи та структури транспортних систем у ГВС 10](#_Toc434784508)

[1.2. Організація управління у ГВС в умовах невизначеності 13](#_Toc434784509)

[1.2.1. Класифікація систем управління 13](#_Toc434784510)

[1.2.2. Ієрархія та задачі рівнів управління ГВС 17](#_Toc434784511)

[1.3. Види невизначеностей, що виникають у ГВС 21](#_Toc434784512)

[1.4. Система динамічного керування як підсистема СОУ ГВС 22](#_Toc434784513)

[1.4.1. Структура системи оперативного управління 22](#_Toc434784514)

[1.4.1. Система динамічного керування 25](#_Toc434784515)

[1.4.2. Основні задачі СДК 26](#_Toc434784516)

[1.4.1. Синтез параметрів СДК 41](#_Toc434784517)

[Висновки. Мета та задачі дослідження 43](#_Toc434784518)

[*2.* Узагальнена концептуальна модель системи динамічного керування в гнучких виробничих системах 44](#_Toc434784519)

[2.1. Формування набору вирішальних класифікаційних ознак та створення класифікатора СДК 44](#_Toc434784520)

[2.1.1. Огляд підходів до ДП 45](#_Toc434784521)

[2.1.2. Огляд політик часу ДП 49](#_Toc434784522)

[2.1.3. Огляд стратегій ДП 52](#_Toc434784523)

[2.1.4. Огляд архітектур СДК 53](#_Toc434784524)

[2.1.5. Огляд методів диспетчеризації 55](#_Toc434784525)

[2.2. Побудова логічної моделі поетапного синтезу СДК 92](#_Toc434784526)

[2.3. Побудова концептуальної моделі СДК 93](#_Toc434784527)

[*3.* Мультиагентне середовище поетапного синтезу системи динамічного керування у гнучкій виробничій системі 99](#_Toc434784528)

[3.1. Інтелектуалізовані агенти та мультиагентні системи 99](#_Toc434784529)

[3.1.1. Загальні риси агентно-орієнтованих систем 99](#_Toc434784530)

[3.1.2. Формальна модель інтелектуалізованого агента 102](#_Toc434784531)

[3.1.3. Формальна модель мультиагентної системи 108](#_Toc434784532)

[3.2. Гнучка інтелектуалізована система вибору СДК на основі концепції нечіткої метаідентифікації 110](#_Toc434784533)

[3.2.1. Концепція нечіткої метаідентифікації вибору СДК 110](#_Toc434784534)

[3.2.2. Компоненти ГІМАК АОП та їх взаємодія в процесі функціонування 113](#_Toc434784535)

[3.2.3. Узагальнена модель гнучкої інтелектуалізованої мультиагентної системи вибору СДК 115](#_Toc434784536)

[3.2.4. Агентно-орієнтований вибір умов 117](#_Toc434784537)

[3.3. Модуль автоматизованого синтезу параметрів у структурі СДК 123](#_Toc434784538)

[Висновки 124](#_Toc434784539)

[4. Моделювання СДК з методом прямої диспетчеризації на основі МАС 125](#_Toc434784540)

[4.1. Синтез СДК для заданої ГВС 125](#_Toc434784541)

[4.2. Метод прямої диспетчеризації на основі МАС 125](#_Toc434784542)

[4.2.1. Модель та структура мультиагентного середовища для реалізації прямої диспетчеризації 126](#_Toc434784543)

[4.2.1. Функції агентів МАС ПД та міжагентна комунікація 128](#_Toc434784544)

[4.3. Застосування СНВ для підвищення ефективності роботи методу МАС динамічного керування 135](#_Toc434784545)

[4.4. Порівняльний аналіз результатів моделювання роботи СДК на основі МАС та правил диспетчеризації 138](#_Toc434784546)

[Висновки 142](#_Toc434784547)

[Висновки 143](#_Toc434784548)

[Список використаних джерел 144](#_Toc434784549)

[Додаток А 157](#_Toc434784550)

[Додаток Б 159](#_Toc434784551)

Список скорочень

|  |  |  |
| --- | --- | --- |
| АС | - | автоматизована система |
| АСНД | - | автоматизована система наукових досліджень |
| АСУ | - | АС управління |
| АСУ ТП | - | АСУ технологічними процесами |
| АТМ | - | автономний транспортний модуль |
| АТСС | - | автоматизована транспортно-складська система |
| БД | - | база даних |
| БЗ | - | база знань |
| ГВМ | - | гнучкий виробничий модуль |
| ГВС | - | гнучка виробнича система |
| КА | - | кінцеві автомати |
| МЛВ | - | механізм логічних висновків |
| НМ | - | нейронні мережі |
| ОДК | - | об'єкт динамічного керування |
| ОУ | - | об'єкт управління |
| ПА | - | процесні алгебри |
| ПЗ | - | програмне забезпечення |
| ПП | - | продукційні правила |
| РК | - | Робокар |
| СМО | - | система масового обслуговування |
| СОУ | - | система оперативного управління |
| ТА | - | теорія агрегатів |
| ТП | - | технологічний процес |
| ЕС | - | експертна система |
| АСНД | - | автоматизовані системи наукового дослідження |
| САПР | - | система автоматизованого проектування |
| АСТПВ | - | автоматизована система технологічної підготовки виробництва |
| АСУВ | - | автоматизована система управління виробництвом |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Вступ

Науково-технічний прогрес призводить до появи нових об'єктів керування, що призначені для функціонування в апріорі невизначених умовах при висуванні підвищених вимог до їх режимів функціонування. Такі умови мають місце при виконанні операцій у виробничих середовищах, ліквідації наслідків аварій, при роботі з вибухонебезпечними предметами тощо. Застосування автоматизованих об’єктів в таких умовах повинно сприяти ефективному виконанню поставлених задач та уникненню загроз здоров’ю та життю людини-оператора.

На сьогоднішній день виробничі середовища мають високу ступінь гнучкості, оскільки стикаються з постійною зміною вимог споживачів. Гнучкі виробничі системи (ГВС) з’явилися як потужні системи, висока гнучкість яких є важливим чинником для забезпечення конкурентоздатності у високо динамічному середовищі. Гнучка виробнича системи – це складна система реального часу, що включає такі основні елементи, як виробничі модулі, автономні транспортні модулі та складські системи[].

Характерною ознакою ГВС, яка відрізняє її від звичайних виробничих систем, є принцип організації вантажопотоків, що передбачає наявність автоматизованої транспортно-складської системи (АТСС). Зазвичай об’єкти виробництва у ГВС проходять через різні виробничі модулі для здійснення відповідних операцій, що значно підвищує важливість використання транспортних модулів. Аналіз роботи звичайних комплексів устаткування з числовим програмним керуванням показує, що значні витрати робочого часу пов’язані із несвоєчасним забезпеченням цього устаткування заготовками, інструментом, оснасткою, тобто з неоптимальністю диспетчеризації матеріальних потоків. Транспортні операції являють собою прямий вираз зв’язків між окремими фазами технологічного процесу, визначаючи тим самим залежність його ефективності від реалізації міжопераційного транспортування об’єктів і засобів виробництва.

Проблеми керування ГВС пов’язані зі складністю взаємозв’язків між її компонентами. Отже, для успішного функціонування ГВС необхідно ще на стадії їх проектування розв’язувати комплекс задач, серед яких однією з найважливіших є формування науково обгрунтованих вимог до технічних характеристик елементів АТСС та структури транспортних мереж (ТрМ), які забезпечують можливість оптимального керування адресуванням матеріальних потоків між компонентами ГВС.

Таке керування на стадії функціонування ГВС є надзвичайно важливим для забезпечення вчасного переміщення об’єктів виробництва між виробничими модулями. Вирішення цієї проблеми повинне містити апріорне планування розкладу дій транспортних модулів для досягнення поставлених цілей. Однак значний вплив невизначеностей виробничого середовища на практиці робить застосування попередньо складеного розкладу неприйнятним. Натомість, для керування рухом в реальному часі застосовують диспетчеризацію.

**Актуальність теми.**

Продуктивність системи подачі матеріалів напряму впливає на продуктивність всього виробництва. Це обумовлює зростаючий інтерес до вирішення задач керування АТСС, зокрема, динамічного керування за умов невизначеності щодо параметрів власне об'єкта і навколишнього середовища, що виявляє себе у вигляді контрольованих і неконтрольованих збурень і перешкод різної природи. Вирішення цієї задачі пов'язано з розробкою систем керування, заснованих на адаптивних, робастних та інтелектуальних підходах.

При наявності достатньо великої кількості публікацій і підходів до побудови систем управління в умовах невизначеності, вирішення задачі, що задовольняють сучасним вимогам, на даний час не отримано. Таким чином, поява задач такої складності потребує розвитку засобів і методів інтелектуального керування об’єктами, застосування методів і технологій штучного інтелекту. Використання таких методів дозволить забезпечити можливість виконання необхідних функцій в умовах невизначеності.

**Об’єктом дослідження** є автоматизовані транспортні модулі у ГВС.

**Предметом дослідження** є процес керування автоматизованими транспортними модулями у ГВС в умовах невизначеності.

**Мета роботи.** Метою дисертаційної роботи є підвищення ефективності функціонування транспортної системи ГВС в умовах невизначеності, шляхом підвищення рівня системи керування.

Для реалізації поставленої мети в дисертації необхідно вирішити такі задачі:

1. Провести аналіз рівнів керування автономними транспортними модулями в ГВС.
2. Виконати аналіз виробничого середовища щодо можливих типів невизначеностей.
3. Сформувати визначення системи динамічного керування як основної підсистеми СОУ при керуванні в реальному часі.
4. Класифікувати системи динамічного керування роботи АТМ.
5. Розробити узагальнену концептуальну модель СДК на основі Ф-функції.
6. Розробити мультиагентний підхід до автоматизованого синтезу системи динамічного керування.
7. Опрацювати експертні знання щодо відповідності класифікаційних ознак СДК та можливих видів невизначеностей у ГВС.
8. Розробити детального алгоритму методу прямої диспетчеризації на основі МАС.
9. Вдосконалити метод прямої диспетчеризації на основі МАС за рахунок систем нечіткого виведення.
10. Провести імітаційне моделювання й експериментальні дослідження на базі обраних структурно-компонувальних схем.

**Методи дослідження.** У ході досліджень використовувалися методи загальної теорії автоматичного управління, теорії нечітких множин, нейронних мереж, системного аналізу, мультиагентних систем.

1. Транспортна система ГВС як об'єкт динамічного керування

Продуктивність роботи сучасних виробничих систем напряму залежить від ефективної взаємодії її складових, зокрема, системи подачі матеріалів та обробних ресурсів в умовах невизначеності. Вдало обрана схема транспортної мережі та оптимально налаштовані системи управління виробництвом на всіх рівнях можуть допомогти мінімізувати втрати від виникнення нештатних ситуацій, пов’язаних з транспортними та оброблювальними ресурсами.

Докладніше розглянемо структуру ГВС та ієрархію її керуючих систем з огляду на функціонування в умовах невизначеності.

* 1. Роль та місце транспортної системи у структурі ГВС
     1. Узагальнена структура ГВС

Інтегрована виробнича система (ІВС) – це багаторівнева ієрархічна система, що реалізує комплексно-автоматизоване групове багатономенклатурне виробництво, яке перебудовується оперативно у визначеному параметричному діапазоні продукції [].

Для комплексно автоматизованого виробництва характерною ознакою є наскрізна автоматизація виробництва продукції – починаючи із завдання на проект і закінчуючи готовою продукцією (проектування – планування – виготовлення).

При груповій технології виробництва передбачається можливість організаціх багатономенклатурного виробництва, при якому обладнання повинне автоматично переналаштовуватися на випуск нової продукції із використання необхідного набору інструментів.

Структура інтегрованої виробничої системи включає (рис. 1.1):

1) гнучкі комп’ютеризовані системи підготовки, організації та управління (ГКС ПОУ);

2) гнучку виробничу систему.

ГКС ПОУ включають:

- автоматизовані системи наукового дослідження (АСНД);

- система автоматизованого проектування (САПР);

- автоматизована система технологічної підготовки виробництва (АСТПВ);

- автоматизована система управління виробництвом (АСУВ).

Гнучка виробнича система – являє собою сукупність різних комбінацій обладнання з числовим програмним керуванням (роботизованих технологічних комплексів, виробничих модулів або іншого технологічного устаткування) і систем забезпечення їх функціонування в автоматичному режимі протягом заданого інтервалу часу, що має здатність автоматизовано переналагоджуватися на виробництво виробів довільної номенклатури у заданих межах значень їх характеристик [].

ГВС складається з наступних функціональних підсистем:

а) технологічна підсистема (комплекси основного технологічного устаткування і допоміжного обладнання);

б) керуюча підсистема (організаційно-технологічні АСУ й автоматизована система управління технологічними процесами АСУТП);

в) сервісна підсистема (діагностика і ремонт основного технологічного устаткування);

г) підсистема підготовки виробництва (підготовка технологічних процесів, управляючих програм до технологічного устаткування, інструментів та пристроїв).

Отже, ГВС розглядається як виробнича одиницю (лінія, ділянка, цех, завод), що працює на основі безлюдної технології, функціонування усіх виробничих компонентів якої координується за рахунок багаторівневої системи керування, що робить можливою як найшвидшу перебудову системи на випуск нової продукції.

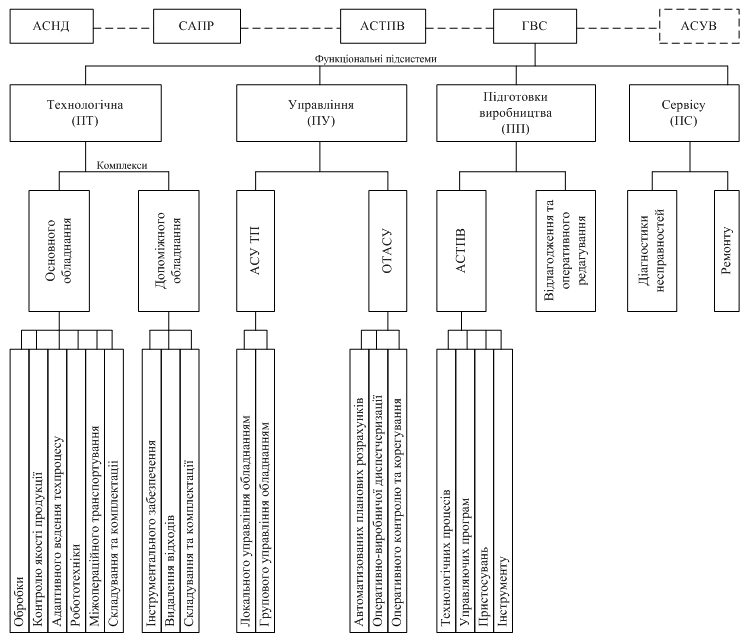


Рисунок 1.1. Узагальнена структура гнучких комп’ютеризованих систем  
в інтегрованому виробництві

* + 1. Типи та структури транспортних систем у ГВС

Компоновочні структури (схеми) ГВС залежать, зокрема, від серійності виробництва, для якого створюється ГВС []. Типові компоновочні структури можна класифікувати за наступними ознаками.

***Типи організації матеріальних потоків*:**

* з централізованим складом;
* з проміжним накопичувачем;
* з комбінованою структурою.

Централізована структура застосовується в одиничному та дрібносерійному виробництві, найчастіше для великогабаритної продукції. Таке розміщення реалізується за схемою склад – верстат - склад (Скл-В-Скл) у ГВС (рис. 1.2, а) або за схемою дільничний накопичувач - верстат - дільничний накопичувач (Н-В-Н) у ГВС, де потрібна мала місткість нагромадження керування АТСС (рис. 1.2, б).

Розміщення із проміжним нагромадженням реалізується за схемою Скл-Н-В-Н-Скл-Н-В-Н-Скл і Скл-Н-В-Н-В-Н-Скл у виробничих системах, що вимагають частих переналагоджень (рис. 1.21, в-г). Такі схеми є найпоширенішими.

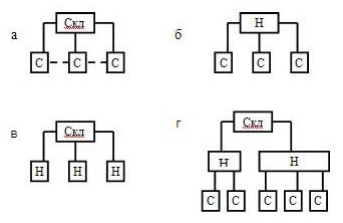


Рисунок 1.2. Типи организації матеріальних потоків

***Прийняті технології*:**

* за методом групування однотипного обладнання, що вирішує проблему максимального завантаження обладнання;
* за методом групової технології - групування різнотипного обладнання для досягнення оптимальної продуктивності, але погіршується рівномірність завантаження обладнання;
* за методом стійкої технологічної послідовності операцій.

У інтегрованому виробництві можливе поєднання зазначених методів.

***Взаємне розташування виробничих та обслуговувальних зон*** (рис. 1.3):

* фронтальне;
* поперечне;
* дипольне;
* кутове;
* кругове;
* комбінове;

Фронтальне, дипольне, поперечне та кутове компонування відносяться до лінійних.

Найбільш простою лінійною структурою компонування є фронтальна. Вона використовується у ГВС для виготовлення виробів у вигляді тіл обертання із застосуванням портальних роботів або маніпуляторів, а також для виготовлення корпусних виробів.

Дипольна структура рекомендується за необхідності перестановок і дає змогу промисловому роботу чи маніпулятору обслугувати два верстати.

Поперечна структура компонування використовується у ГВС, що виготовляють корпусні вироби або вироби типу тіл обертання. У якості транспортного модуля може бути використано конвеєр або портальний робот.

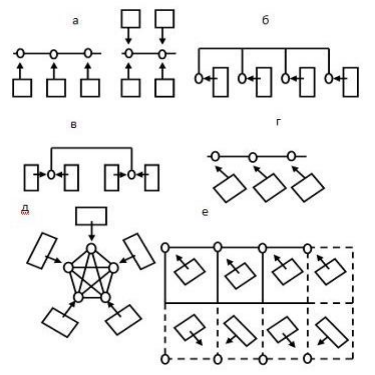


Рисунок 1.3. Схеми розміщення технологічного обладнання: а – фронтальна; б – поперечна; в – дипольна; г – кутова; д – кругова; е – комбінована

Кутове компонування ГВС дає змогу зменшити її довжину та мають сенс у випадку випуску виробів типу тіл обертання.

Кругова структура компонування застосовується у ГВС для виготовлення різних виробів при використанні маніпулятора обертального типу у якості транспортного та завантажувального засобу.

Комбіноване компонування доцільне, коли воно виконується у формі решітки із квадратами.

***Способи реалізації систем подачі інструмента*:**

* ручний візок;
* загальний автоматичний транспорт для деталей та інструмента;
* спеціальний транспорт для інструмента.

***Компонування АТСС за напрямком довжини складу*:**

* уздовж лінії з верстатами;
* перпендикулярно лінії з верстатами;
* без складу.

Склади уздовж лінії з верстататами є сенс застосовувати при розташуванні верстатів в один ряд.

Поперечні склади застосовуються при багаторядному розміщенні верстатів. Таке планування займає меншу площу, але необхідний транспорт для обслуговування верстатів.

* 1. Організація управління у ГВС в умовах невизначеності
     1. Класифікація систем управління

Усі цілеспрямовані процеси, що мають місце у ІВС, являють собою сукупність операцій, які, за видом виконуваних функцій, можна умовно розділити на два типи: виробничі операції та операції управління.

Виробничі операції виконують виробничу функцію ІВС, що полягає у реалізації технологічного процесу випуску продукції технологічною підсистемою у автоматизованому режимі.

Під технологічним процесом мається на увазі сукупність дій, що виконуються для забезпечення вирішення поставлених задач, що орієнтовані на перетворення з використанням засобів виробництва, предметів і продуктів виробництва[].

Задля досягнення цілей технологічного процесу необхідно виробничі операції організовувати та направлятися іншими діями – операціями управління. Вони виконують інформаційну функцію ІВС, що полягає у реалізації збору, обробки та видачі інформації для здійснення управління технологічним процесом в автоматизованому режимі.

Управління – це процес, що виконується у реальному часі та забезпечує хід виконання технологічного процесу. Отже система управління, в межах якої реалізується процес управління, утворює особливий підклас динамічних систем [138].

Як результат такого розподілу функцій систему управління виробництвом можна представити класичною системою організації управління, яка використовується у динамічних системах зі зворотним зв’язком за станом (рис. 1.4).

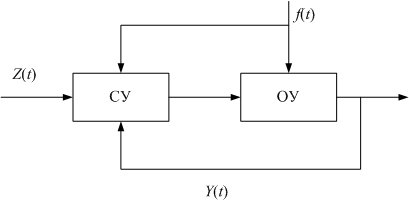


Рисунок 1.4. Загальна схема системи управління ІВС

Процес керування у системах зі зворотним зв’язком реалізується згідно з програмою управління, що задається для поточних або для всіх моментів часу. Суть процесу у визначенні відхилення оцінки поточного стану від еталонного стану на даний момент згідно з законом управління у системі, ззадля утримання її стану при умові наявних обмежень ресурсів та забезпечення показників ефективності управління.

Об’єкт управління у ІВС досить узагальнене поняття. До нього входять технологічні процеси виробництва, процеси із забезпечення функціонування виробничого обладнання та руху матеріальних потоків. Також об’єктами управління виступають окремі агрегати, верстати, цехи, ділянки, конструкторські та економічні підрозділи. Взаємодія СУ та ОУ задається інформаційними зв’язками. Межа між ОУ та СУ знаходиться там, де інформаційні процеси переходять у зміну перебігу матеріальних процесів.

Широке різноманіття об’єктів управління потребую неменшої кількості систем управління. Найросповсюдженіші системи управління можуть бути класифіковані за наступними ознаками []:

1. Рівень автоматизації інформаційної функції.

2. Рівень складності системи управління.

3. Рівень визначеності системи управління.

4. Принцип управління.

5. Структура рівнів управління.

Залежно від *ступеня автоматизації* виділяють ручне, автоматизоване і автоматичне управління. При ручному керуванні весь процес здійснюється людиною. У автоматизованих системах функції збору і обробки інформації виконуються машиною, а безпосередньо прийняття рішення – людиною. У автоматичних системах управління здійснюється взагалі без участі людини.

За *рівнем складності* системи управління поділяються на прості та складні. Поняття “складна система” містить відображення того факту, що повний опис деяких об’єктів є важким або зовсім неможливим.

Складним системам характерні такі відмінності:

–  кількість параметрів, що описують систему є досить великою, багато параметрів не може бути описано кількісно;

–  мета управління є функцією часу і не може бути формалізована без значних спрощень;

За рівнем визначеності систему поділяються на ймовірнісні та детерміновані.

Детермінована система передбачає, що згідно попередньому стану та необхідній додатковій інформації може бути сформований наступний стан.

У імовірнісній системі може бути сформована лише множина наступних станів і, відповідно їх ймовірність.

Поділ систем на прості та складні, детерміновані та ймовірнісні певною мірою є умовним, він залежить від рівня дослідженності певної конкретної системи та використовуваних засобів моделювання.

Однією із найголовніших ознак системи управління являється її структура. Вона описує внутрішню середовище системи та відображає стійкі зв’язки між елементами.

Виділяють централізовані та децентралізовані структури системи згідно із принципом управління.

Для централізованих систем характерно, що усі процеси управління зібрані в одному керуючому органі. Цей орган здійснює обробку інформації, що надходить від всіх елементів об’єкта управління.

У розподілених системах кожномуо елементу об’єкта управління відповідає власний керуючий орган, що здійснює реалізацію локальної мети функціонування елемента, а узагальнена мета системи реалізується шляхом врахування даних, що надходять від решти елементів об’єкта управління (рис. 1.5).

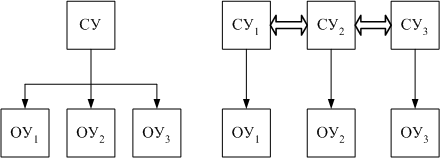


Рисунок 1.5. Централізована та розподілена системи управління

За рівнем підпорядкованості виділяють однорівневу та багаторівневу (ієрархічну) системи управління.

У системах з ієрархічною підпорядкованістю керуючі функції розподілені між кількома органами із дотриманням централізованості. У такому випадку керуюча інформація переходить з верхніх на нижні рівні, і у зворотному напрямку – інформація про стан (рис. 1.6).

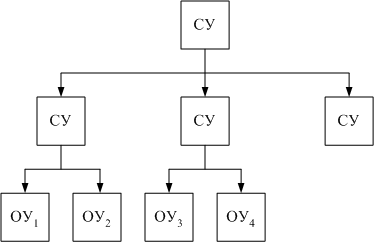


Рисунок 1.6. Ієрархічна система управління

Для інтегрованого виробництва характерні, як правило, бараторівнені гнучкі комп’ютеризовані системи керування.

* + 1. Ієрархія та задачі рівнів управління ГВС

При побудові структури систем управління ІВС зазвичай використовується ієрархічний та функціональний принципи поділу на підсистеми. Усі рівні характеризуються особливими цілями та задачами керування.

Згідно з ієрархічним принципом можуть бути виділені такі рівні та підсистеми в ІВС.

***Виконавчий*** ***рівень*** (перший) утворений з окремих систем управління виробничими ланками та обладнанням. Мета даного рівня управління – вибір та підтримка необхідних режимів здійснення технологічних операцій. Управління в основному зводиться до контролю параметрів виконання технологічної операції і до безпосереднього впливу на виконавчі механізми.

***Тактичний*** ***рівень*** (другий) складається з систем диспетчеризації виробничих ділянок і ліній. Основна мета управління на цьому рівні – оперативна диспетчеризація переміщення матеріальних потоків, що реалізується за допомогою вибору і підтримки режимів функціонування взаємозалежних виробничих ланок. На даному рівні здійснюється пристосування календарного плану технологічних операцій до поточної виробничої ситуації.

***Стратегічний*** ***рівень*** (третій) ієрархії утворений системами організаційного керування виробничими підрозділами. Основна мета даного рівня – це напрацювання узгодженого плану роботи структурних підрозділів протягом оперативного планового періоду. На цьому рівні деталізуються виробничі завдання на випуск продукції виробничими ланками.

При пооєднанні систем керування другого і третього рівнів утворюється організаційно-технологічна автоматизована система управління (ОТАСУ) виробничими підрозділами.

***Адміністративний рівень*** (четвертий) ієрархії складають системи, які повинні забезпечувати управління підприємством у цілому. Мета управління – підготовка та організація спільної роботи виробничих та інших підрозділів підприємства, що здійснюють випуск заданої кількості продукції встановлених техніко-економічних показників. Для здійснення поставленої мети в управлінні мають буьт реалізувані функції економічного характеру.

Особливості організаційно-економічних і організаційно-технологічних систем відображаються у основних задач управління (табл. 1.1), котрі забезпечують реалізацію загальної мети підприємства [138]. Цією метою є виконання планового завдання, що зазначає номенклатуру, обсяг, терміни та умови залучення виробничих ресурсів. При здійсненні планування на основі планового завдання знаходяться техніко-економічні показники підприємства в цілому і визначається деталізація показників по підрозділах, цехах і т.д.. Залежно від періоду планування поділяють на поточне та перспективне.

Перспективне планування передбачає створення планів роботи підприємства на довгий період (більше року).

Поточне планування може бути розділене на техніко-економічне (річне) та оперативне. Техніко-економічне планування конкретизує показники перспективного плану на поточний період і корегує їх згідно із вимогами завдання. Оперативним плануванням конкретизуються параметри техніко-економічного плану на малі проміжки планового періоду (місяць, декада, доба) та задаються дії щодо їх виконання. Оперативні плани враховують результати виконання попередніх планів.

У організаційно-технологічних системах разом розглядаються три задачі: оперативне планування, контроль (знаходження відхилення поточного стану від запланованого) та регулювання (диспетчеризація). Цей перелік задач лежить у основі оперативного управління. Система, яка реалізує дані задачі, називається системою оперативного управління (СОУ).

Таблиця 1.1. Ієрархія і задачі рівнів управління ІВС

|  |  |  |  |
| --- | --- | --- | --- |
| **Рівень** | **Тип системи управління** | **Мета і виконувана функція управління** | **Задачі** |
| АСУВ | Адміністративний | Техніко-економічне планування;  Економічне управління | 1. Формування підприємницької програми підприємства 2. Статистичний облік показників роботи підприємства 3. Управління збутом продукції 4. Контроль якості готової продукції 5. Управління матеріально-технічним постачанням |
| Стратегічний | ОТАСУ – система оперативного планування | Оперативне планування;  Організаційне управління | 1. Формування календарних планів змінно-добових завдань 2. Оперативний облік загальних показників роботи ІПС 3. Коректування календарних планів 4. Ведення оперативних баз даних |
| Тактичний | ОТАСУ – система оперативно- диспетчерського управління | Оперативна диспетчеризація;  Технологічне управління | 1. Диспетчеризація матеріальних потоків 2. Аналіз та оперативний контроль 3. Облік первинної інформації про стан обладнання 4. Диспетчеризація інформаційних потоків 5. Оперативне регулювання |
| Виконавчий | АСУТП СУ ГВМ СУ АТЗ СУ АНС | Логіко-програмне управління обладнанням;  Локальне управління | 1. Управління технологічними операціями 2. Управління транспортними операціями 3. Управління допоміжними операціями 4. Управління ПР |

* 1. Види невизначеностей, що виникають у ГВС

Виробничі середовища динамічні за своєю природою і схильні до виникнення різних ***невизначеностей***.

***Визначення 1.1.*** *Невизаченості* – являють собою до події в реальному часі, які можуть змінити стан системи та/або впливають на її продуктивність.

Література з динамічного керування розглядає значне число подій в реальному часі і їх вплив з урахуванням різних виробничих систем, в тому числі одно-машинних систем, систем паралельних машин, конвеєрів, цехів і гнучких виробничих систем.

Події у режимі реального часу були розділені на дві категорії: пов'язані з ресурсами та пов'язані із задачами [, , , ]. Дослідимо більш детально окремі найпоширеніші невизначеності двох зазначених типів на предмет виникнення на певному етапі керування виробничою системою, результати зведемо у таблицю 1.2.

Таблиця 1.2. Невизначеності на різних рівнях керування ГВС

|  |  |  |
| --- | --- | --- |
| **Тип невизначеності** | **Невизначеність** | **Тип системи управління** |
| Пов'язані з ресурсами | *несправність машини* | СОУ |
| *помилка оператора* | СОУ |
| *відсутність або несправність інструмента* | СОУ, АСАУ |
| *ліміти завантаження* | СОУ, АСАУ |
| *затримки у доставці матеріалів* | СОУ, АСАУ |
| *дефектність матеріалу* | СОУ |
| Пов'язані з операціями | *термінові операції* | СОУ |
| *відміна операцій* | СОУ |
| *зміни терміну виконання* | СОУ |
| *невчасне надходження операцій* | СОУ |
| *зміна пріоритету операцій* | СОУ |
| *зміна тривалості виконання операцій* | СОУ |

* 1. Система динамічного керування як підсистема СОУ ГВС

Задля забезпечення успішної взаємодії складових ГВС і збереження рівня її продуктивності необхідно здійснювати ефективне керування в умовах невизначеності.

Враховуючи значну кількість невизначених подій на оперативному рівні функціонування системи та їх вплив на роботу системи, при розгляді ієрархії системи керування ГВС, слід детальніше зупинитися на *системі оперативного управління* (СОУ) [138]. У даній роботі, зокрема, приділимо увагу підсистемі СОУ, що безпосередньо управляє ходом виробничого процесу – *системі динамічного керування* (СДК).

* + 1. Структура системи оперативного управління

Головною складовою виробничих відносин у складі ІВС являється ГВС, зв’язуючи різні підсистеми, такі як: проектування та підготовки виробництва з безпосередньо засобами обробки матеріалів. У ГВС функції керування здійснюються за допомогою ОТАСУ, головна мета якої полягає в організації оперативного управління матеріальними потоками, що полянає в упорядкуванні проходження матеріальних потоків через оброблювальні ресурси. Таким чином, ОТАСУ також називають система управління матеріальними потоками (СОУ МП).

Проаналізуємо функціональну структуру та основні принципи роботи СОУ ГВС (рис. 1.7). СОУ будується за принципом управління із зворотним зв’язком та утворюється з наступних основних блоків:

– оперативно-календарного планування (*P1*);

– оперативного обліку (*P2*);

– оперативного контролю (*P3*);

– оперативної диспетчеризації (*P4*);

– локального управління устаткуванням (*P5*);

– оперативного коректування (*P6*);

– статистичного обліку (*P7*).

До СОУ на вхід подається планове завдання, що представлене як виробнича програма, технологія обробки, фонди матеріальних ресурсів та техніко-економічні показники роботи ГВС. На основі заданої цілі Z блок P1 створює поопераційні календарні плани і складає змінно-добовий план P завантаженості обладнання.

У блоці P2 здійснюється обробка та зберігання первинних даних про стан устаткування та матеріальних потоків, що утворюють об’єкт управління (ОУ). Первинні дані надходять як сигнiали *X* щодо зміни компонентів ОУ (наприклад сигнал про закінчення запланованих робіт згідно із завданням *P*). Також блок P2 здійснює формування вектора S – поточного стану комплексу всіх компонентів ОУ.

У блоці *P3* проводиться подальший аналіз ситуацій, що мали місце в ОУ, визначається співвідношення між запланованим та реальним ходом виробництва. З цією метою розраховується ступінь неузгодженості *D* = *S* – *P*. У разі, коли дана величина не перевищує допустимого значення, дані про реальний стан ОУ надходять до *P4* задля визначення *Y –*  керуючого впливу, що покликаний усунути неузгодженості та виконати наступне завдання з плану *P*. Реалізацію керуючого впливу *Y* здійснює блок *P5*, котрий додатково користується власними вхідними даними щодо стану *К* та керуючих команд *V*.

У разі перевищення величиною неузгодженості допустимого значення управління ГВС переходить до блоку *P6*, котрий здійснює перерахунок календарного плану на залишок планового періоду з урахуванням обмежень щодо виробничих ресурсів. Згідно із результатами розрахунків формується рішення щодо подальшого функціонування ГВС:

* продовжувати роботу ГВСзгідно із перерахованим планом за умови можливості дотримання встановленого терміну виконання виробничої програми.
* надіслати повідомлення *W* про неможливість завершення виробничої програми в заданий термін без внесення змін до планового завдання та переліку наявних виробничих ресурсів.

Зберігання та статистичний аналіз інформації про стан *S*, що може бути використана на вищих рівнях управління (АСУВ), здійснюється у блоці P7. Там здійснюється обробка облікових даних щодо ОУ за зміну і формується узагальнююча інформація *F*, яка містить зокрема дані про число виготовлених деталей, рівень завантаження та простоїв обладнання, обсяг незакінченого виробництва.

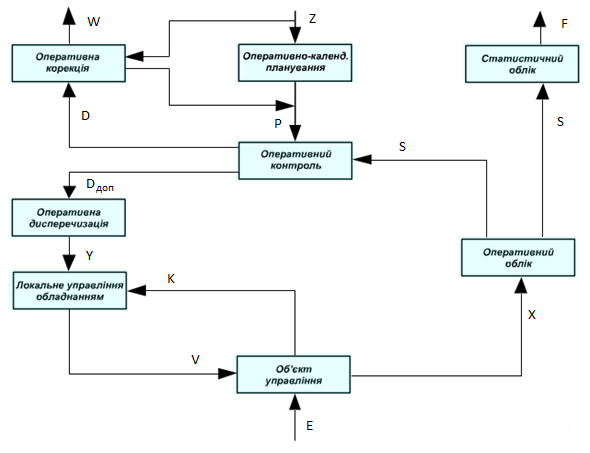


Рисунок 1.7.Функціональна структура системи оперативного керування ГВС:

|  |  |  |
| --- | --- | --- |
| *Z* | – | виробниче завдання; |
| *P* | – | календарний план; |
| *S* | – | стан усього комплексу устаткування; |
| *E* | – | збуджуючий вплив; |
| *D* | – | відхилення від планових термінів завершення технологічних операцій; |
| *D*доп | – | локальні резерви часу виконання технологічних операцій; |
| *Y* | – | керуючі завдання; |
| *V* | – | мікрокоманди на виконання елементарних операцій; |
| *K* | – | сигнал зворотного зв’язку; |
| *X* | – | інформація про завершення виконання завдання *Y*; |
| *W* | – | оповіщення системи вищого рівня про неможливість виконання поточного планового завдання; |
| *F* | – | інформація про незавершене виробництво і стан усього комплексу устаткування. |

* + 1. Система динамічного керування

***Визначення 1.1.*** *Динамічне керування ГВС* – процес керування ГВС в умовах динамічного виробничого середовища (за наявності невизначених подій в реальному часі), спрямований на дотримання строків запуску-випуску продукції на технологічне обладнання відповідно до розробленого системою оперативного планування розкладу роботи за рахунок своєчасного ***транспортного обслуговування*** заявок, що надходять від технологічного обладнання.

***Твердження 1.*** Під *динамічним виробничим середовищем* розуміється таке виробниче середовище, виробничий процес якого відбувається за наявності невизначених подій у реальному часі, що впливають на його хід та\або продуктивність.

***Визначення 1.2.*** *Система динамічного керування* – підсистема системи оперативного управління, що відповідно до обраного підходу реалізує спланований розклад роботи технологічного устаткування в умовах динамічного виробничого середовища (за наявності невизначених подій в реальному часі).

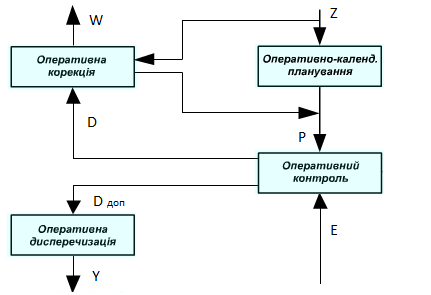


Рисунок 1.8. Структура СДК

Структура системи включає наступні модулі СОУ: оперативного планування, оперативного контролю, оперативної корекції та оперативної диспетчеризації. Структура системи та її інформаційних потоків зображена на рис. 1.8.

* + 1. Основні задачі СДК

Твердження 1. Здійснення оперативного управління ГВС полягає у подальшій деталізації за виконавцями попередньо складеного календарного плану випуску продукції у границях визначеного планового інтервалу.

Із цього випливає визначення задачі оперативного планування, як задачі просторового та часового упорядкування необхідного обсягу запланованих робіт. Під просторовим упорядкуванням розуміється призначенні всім виконавцям поопераційних планів робіт, а під часовим – визначення черговості або часу виконання операцій.

Твердження 1. Для виконання календарних планів та виробничих програм, що були розроблені, необхідна система оперативного контролю та корегування виробничого процесу.

Твердження можна вважати справедливим через неможливість виключення ряду випадкових факторів, що здійснюють вплив на виробничий процес. Серед них: поломки обладнання, брак, термінова зміна плану, перебої у постачанні ресурсів тощо.

Диспетчеризація процесу виробництва (оперативний контроль та корегування) здійснюється на всіх рівнях і підрозділах ГВС.

Отже, можна виділити наступні основні задачі системи динамічного планування:

* оперативне планування;
* оперативний контроль;
* оперативна корекція;
* оперативно-диспетчерське керування.

***Оперативне планування.***

Тривалісьть оперативного планового інтервалу у ГВС зазвичай визначається однією зміною, а його виконавцем є технологічне устаткування.

Основною математичною формою подання таких задач являється оптимізаційна модель, а серед основних методів роз'язання – дискретне лінійне програмування (ДЛП) та імітаційне моделювання.

Основними об'єктами у процесі планування виступають завдання (роботи), що являють собою сукупність технологічних операцій, партій деталей, для яких необхідно визначити послідовність обслуговування на обладнанні за фіксованих технологічних маршрутів.

Отже формалізуємо визначену задачу.

Номенклатура продукції складається з *n* найменувань (*j* = 1,*n*), що виробляються на *m* групах (*k* = 1,*m*) обладнання.

Для виготовлення кожного найменування продукції заздлегідь визначено технологічний маршрут – послідовність проходження через групи обладнання, що може бути подана у наступному вигляді: *Gj* = (*Lij* | *i* = 1,*Mj*), де *Lіj* – технологічна операція, що здійснюється *і*-ою у послідовності виготовлення *j*-ої одиниці продукції;  *Mj* – загальна кількість операцій,що здійснюються над  *j*-ою деталлю.

Технологічні операції *Lij* = (*Qij*, *Tij*), що утворюють маршрут, мають наступні характеристики:

*Qij* = *k* – номер групи налаштованого на виконання операції *Lij* обладнання;

*Tij* – нормативний час виконання операції *Lij*.

Отже, необхідно створити розклад *P* = (*Tiji* | *i* = 1,*Mj*, *j* = 1,*n*), що задає час початку виконання операцій *Lіj* (час початку обробки партії деталей на пених одиницях обладнання) і водночас задовольняє системі наступних обмежень:

– *T*н*ij* ≥ *T*к*i*-1,*j* – умова дотримання технологічної послідовності;

– умова дотримання технологічних маршрутів;

– умова безперервного виконання операцій: *T*к*ij* = *T*н*ij* + *Tij*;

– – умова одночасно можливого здійснення лише однієї операції на одному обладнані в кожен момент часу, де *T*к*ij* – момент закінчення виконання операції *Lіj*.

Вибір оптимального розкладу вионується за певним критерієм ефективності, вибір якого здійснюється індивідуально залежно від особливостей та потреб конкретної виробничої системи з метою досягнення найкращих показників економічної діяльності.

Найбільш поширеними типами критеріїв є:

1) ***мінімізація виробничого циклу*** – тобто часу, що необхідний для випуску потрібного обсягу продукці:

|  |  |
| --- | --- |
|  | (1.1) |
|  | (1.2) |
|  | (1.3) |

де *Tk*р – сумарний час виконання операцій на *k*-ій одиниці обладнання;

*Tk*п – сумарний час простоїв *k*-ої одиниці обладнання;

*Tij*оч – очікування  *j*-ої деталі перед обробкою на  *і*-й операції;

2) ***оптимізація використання обладнання*** (фондовіддачі):

– *максимізація завантаження обладнання*, а саме:

|  |  |  |
| --- | --- | --- |
| мінімального: |  | (1.4) |
| загального: |  | (1.5) |

де *Kk*з = *Tk*р / (*Tk*р + *Tk*п) – коефіцієнт завантаження *k*-ої одиниці обладнання;

– *мінімізація часу простою обладнання*, а саме:

|  |  |  |
| --- | --- | --- |
| максимального: |  | (1.6) |
| максимального  міжопераційного: |  | (1.7) |
| загального: |  | (1.8) |

де *Tij*п – простій *k*-ої одиниці обладнання (*k = Qіj*) перед виконанням операції *Lіj*;

 – сумарний простій *k*-ої одиниці обладнання;

– *мінімізація середнього міжопераційного простою обладнання*, а саме:

|  |  |  |
| --- | --- | --- |
| максимального: |  | (1.9) |
| загального: |  | (1.10) |

де *Nk* – кількість операцій (або простоїв, за рівної кількості операцій), які здійснюються на *k*-й одиниці обладнання;

3) ***мінімізація незавершеного виробництва***:

– *мінімізація очікування деталей перед обробкою*, а саме:

|  |  |  |
| --- | --- | --- |
| максимального міжопераційного: |  | (1.11) |
| максимального подетального: |  | (1.12) |
| загального: |  | (1.13) |

– *мінімізація середнього очікування деталей перед обробкою*, а саме:

|  |  |  |
| --- | --- | --- |
| максимального: |  | (1.14) |
|  | (1.15) |
| загального: |  | (1.16) |
|  | (1.17) |

де *Nj* – час очікування *j*-ої деталі перед обробкою (у разі, якщо *Mj* однакове для всіх деталей).

Кожен з наведених типів критеріїв орієнтований на задоволення тільки власного показника ефективності виробництва. Часто в реальних виробничий умовах необхідно вирішувати задачу оперативного планування, враховуючи одночасно декілька критеріїв. Одним із засобів розв’язання багатокритеріальної задачі є створення компромісного критерію.

***Оперативний (диспетчерський) контроль*** реалізується на базі оперативних даних щодо реального перебігу виробничого процесу.

Задачі оперативного контролю виконуються на трьох рівнях системи управління інтегрованим виробництвом.

На *третьому* і *четвертому рівнях* *керування* відбувається контроль працездатності устаткування, правильності дотримання програм системи ЧПУ, ефективність керуючих впливів виконавчих механізмів.

Одна з основних задач – це діагностика стану обладнання та інструментів, вирішення якої дозволяє запобігти аваріям та відмовам. Для діагностування використовується інформація від датчиків на набільш важливих елементах обладнання.

Періодично керуючою системою проводиться зчитування даних з датчиків та порівняння їх з критичними величинами, а у випадку виходу за межі – видача повідомлення оператору або аварійне завершення роботи обладнання.

Задля забезпечення своєчасності реакції оператора зміни умов роботи ГВС та запобігання виникнення нештатних ситуацій на виробництві, інформація щодо стану системи повинна надходити до оператора вчасно та у зручній та легкій для сприйняття формі. Формою представлення таких даних зазвичай є дисплеї, де зображена мнемосхема контрольованої ділянки виробництва. Для сприяння швидкому прийняттю рішень, на екран такод виводяться основні кількісні параметри виробництва. Будь-яку іншу інформацію оператор може отримати додатково сформувавши запит до системи.

На *другому* ієрархічному рівні реалізується диспетчерський контроль за ходом здійснення оперативного плану. Диспетчерський контроль характерний для всіх типів дискретного виробництва, але у ГВС він має свою специфіку. Головною особливістю є складність одержання даних щодо стану обладнання, перебіг виконання технологічних операцій через майже повну відсутність людей у виробничих приміщеннях. В умовах такого високого рівня автоматизації дані щодо стану виробництва можуть бути отримані тільки від необхідних датчиків. Проте, через обмеженість їх кількісті та надійністі може виникати нестача інформації про процес.

Використовуються методи, котрі дозволяють організувати диспетчерський контроль із використанням математичних моделей та обчислювальної техніки. Серед них метод на основі синхронної імітаційної моделі технологічного процесу. У основі такого методу лежить створення імітаційної моделі виробничої системи та синхронізувати її із процесом функціонування реального об’єкта. Таким чином можна використовувати засоби для аналізу розбіжностей моделі та реального процесу, а також функціонал, що дозволяє фіксувати на моделі та про важливі події, що не були зафіксовані на реальному об’єкті.

На рис. 1.9. наведено структуру системи диспетчерського контролю зазначеного типу, із синхронною імітаційною моделью.

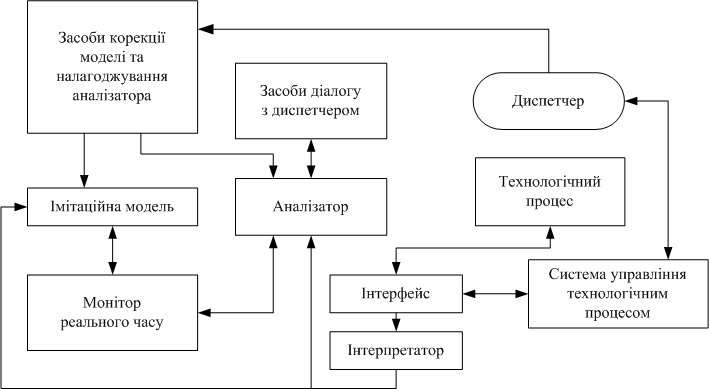


Рисунок 1.9. Структура системи оперативного контролю із синхронною моделлю

***Оперативне корегування*** реалізується за допомогою створення нового плану на повний оперативний період чи на його частину. Воно може полягати у вирішенні задач зміни частини плану за умови узгодженості з іншими частинами, що не змінилися або у зсуву часу запуску заготовок в роботу у межах допустимих інтервалів. Корегування стає необхідним зазвичай через відмову обладнання, проблеми з постачанням заготовок чи інструментів, випуск браку.

Для здійснення оперативного корегування необхідно, щоб така можливість була закладена у загальну стратегію оперативного управління і методику календарного планування.

Одна з стратегій оперативного керування – виділення понаднормових інтервалів планування, передбачає поділ оперативного місячного періоду на короткі понад оперативні інтервали (ПОІ). Вони можуть виділятися на період декади, тижні чи доби.

Наприклад, оберемо у якості ПОІ тиждень. Спочатку задача календарного планування вирішується для перших двох тижнів. По завершенню першого тижня формується календарний план на третій тиждень, після другого – на четверий.

Така система планування називається оперативним плануванням з «пересувним» горизонтом.

Розглянемо, як здійснюються корегування у такій системі. Після закінчення *k* - го ПОІ планується (*k* + 2) - й. На основі інформації щодо контролю за виконанням плану *k* - го ПОІ знаходиться множина невиконаних робіт *P*1. Після цього знаходиться множина робіт *P*2, котрі потрібно виконати для заміни браку. З урахуванням додаткової інформації диспетчер формує множину додаткових робіт *P*3, що можуть з’явитися у зв’язку із зміною замовлення на готову продукцію. Множина *Pk*+2 на (*k*+2)-му ПОІ визначається як об’єднання множин: *Pk*+2 = *P*1 http://tc.kpi.ua/content/lib/gks/book3/glav0232/img001.gif *P*2 http://tc.kpi.ua/content/lib/gks/book3/glav0232/img001.gif *P*3 http://tc.kpi.ua/content/lib/gks/book3/glav0232/img001.gif *P*4, де *P*4 – роботи, що заздалегідь заплановані на (*k*+2)-й ПОІ.

Для виконання решти робіт, крім заздалегідь запланованих (*P*1, *P*2, *P*3 ) потрібно мати у запасі необхідний резерв продуктивності, який повинен закладатися на етапі розбивання місячного плану на ПОІ.

Зазначені резерви можуть бути забезпечені наступним чином. Усі роботи можна поділити на обов’язкові *Р*об – роботи з термінами виконання в рамках ПОІ та фонові *Р*ф – терміни виконання яких поза ПОІ.Якщо видається можливим виконання множини робіт *Рk*+2 – вона приймається повністю, якщо ж ні – то *Рk*+2 розглядається як: *Р**k*+2 = *Р*об*k*+2  http://tc.kpi.ua/content/lib/gks/book3/glav0232/img001.gif  *Р*ф*k*+2. У такому випадку доступні ресурси виділяються для множини *Р*об*k*+2, а ті, що залишилися вільними – на роботи із множини *Р*ф*k*+2.

Зазначена стратегія може бути доповнена можливістю зміни календарного плану всередині одного ПОІ. Можливість такого корегування необхідно закладати при створенні календарного плану шляхом введення локальних резервів часу для найважливіших одиниць обладнання.

Локальний резерв – часовий інтервал, що дозволяє забезпечити збільшення тривалості операції без зміни часу початку наступної операції.

Локальні резерви часу можуть утворюватися самостійно, за рахунок неможливості повного завантаження обладнання, а також – вводитися у розклад спеціально. З їх допомогою можна нівелювати вплив тимчасового виходу з ладу обладнання, додати додаткові термінові замовлення без порушення структури календарного плану. Якщо локальні резерви не використовуються для компенсації збурюючи впливів – їх можна заповнити фоновими роботами для запобігання зниження завантаження обладнання.

Завдяки локальним резервам можна компенсувати незначні збурення, тоді як серйозні зміни пріоритетів, значні порушення термінів постачання матеріалів чи інструменту, тривала відмова обладнання – потребують розрахунку нового оперативного плану для залишку ПОІ. Перерахунок плану необхідно здійснити за час, не довший інтервалу, протягом якого ГВС все ще може працювати за старим планом. Необхідно обирати компромісні алгоритми між швидкістю розрахунку та якістю отриманого нового плану в залежності від ситуації.

* + 1. Синтез параметрів СДК

Достатньо широке різноманіття властивостей СДК дозволяє обрати модель, найбільш адекватну до вимог конкретної ГВС з властивими їй видами невизначених ситуацій, разом з тим ускладнюючи процес вибору його багатоваріантністю, у чому можна наочно переконатися ознайомившись із функціональною схемою СДК (рис. 1.10).



Рисунок 1.10. Функціональна схема СДК

Необхідним є вибір наступних основних параметрів, що визначають процес динамічного керування:

*- параметри оперативного планування* передбачають визначення ступеня повтоноти оперативного плану, основних критеріїв його ефективності та механізмів їх досягнення.

*- параметри оперативного контролю* передбачає визначення моменту здійснення процесу контролю та прийняття рішення про необхідність перепланування оперативної роботи виробничої системи.

*- параметри оперативної корекції* перепланування передбачає визначення обсягу змін, що вносяться до початкового або попередньо визначеного плану.

*- параметри оперативної диспетчеризації* диспетчеризації передбачає визначення основних алгоритмів утворення керуючого впливу для своєчасного обслуговування транспортними модулями задач транспортування від оброблювальних ресурсів.

Задля зниження впливу суб'єктивних факторів, пов'язаних з рівнем компетенції оператора, що здійснює налагодження роботи системи, необхідно розробити підхід до автоматизованого синтезу системи динамічного керування адекватної до заданої виробничої системи.

Висновки. Мета та задачі дослідження

Метою дисертаційної роботи є підвищення ефективності функціонування транспортної системи ГВС в умовах невизначеності, за рахунок побудови інтелектуальної системи керування.

Для реалізації поставленої мети в дисертації необхідно вирішити такі задачі:

1. Провести аналіз рівнів керування автономними транспортними модулями в ГВС.
2. Виконати аналіз виробничого середовища щодо можливих типів невизначеностей.
3. Сформувати визначення системи динамічного керування як основної підсистеми СОУ при керуванні в реальному часі.
4. Класифікувати системи динамічного керування роботи АТМ.
5. Розробити узагальнену концептуальну модель СДК на основі Ф-функції.
6. Розробити мультиагентний підхід до автоматизованого синтезу системи динамічного керування.
7. Опрацювати експертні знання щодо відповідності класифікаційних ознак СДК та можливих видів невизначеностей у ГВС.
8. Розробити детального алгоритму методу прямої диспетчеризації на основі МАС.
9. Вдосконалити метод прямої диспетчеризації на основі МАС за рахунок систем нечіткого виведення.
10. Провести імітаційне моделювання й експериментальні дослідження на базі обраних структурно-компонувальних схем.
11. Узагальнена концептуальна модель системи динамічного керування в гнучких виробничих системах

Достатньо широке різноманіття властивостей СДК дозволяє обрати модель, найбільш адекватну до вимог конкретної ГВС з властивими їй видами невизначених ситуацій, разом з тим ускладнюючи процес вибору багатоваріантністю останнього. Прийняття рішення щодо вибору параметрів СДК покладається на оператора системи.

З метою зниження впливу суб'єктивних факторів, пов'язаних з рівнем компетенції оператора, що здійснює налагодження роботи системи використаємо підхід запропонований в []. Даний підхід було застосовано для визначення топологій штучних нейронних сіток, що задовольняють вимогам певних прикладних задач.

Отже, згідно з розглянутим підходом, для синтезу моделі системи динамічного керування адекватної до заданої виробничої системи необхідно:

* сформувати *набір вирішальних класифікаційних ознак* (НВКО) і створити класифікатор СДК;
* побудувати *логічну модель поетапного синтезу* (ЛМПС) моделі СДК;
* створити *узагальнену концептуальну модель СДК* на основі набору ітераційних процедур, що забезпечуються складом і послідовністю етапів синтезу.
  1. Формування набору вирішальних класифікаційних ознак та створення класифікатора СДК

***Визначення 2.1.*** *Набір вирішальних класифікаційних ознак СДК* – це їх найменша можлива сукупність *необхідна* для формалізації процесу представлення основних властивостей і вибору задовольняючих моделей СДК і *достатня* для адекватного обслуговування вимог ГВС.

***Формування НВКО.*** На основі здійсненого аналітичного огляду, а також виходячи із результатів численних досліджень в інших джерелах (наведені в []), що присвячені даній проблемі, можемо сформувати НВКО, що дасть змогу здійснювати послідовне обґрунтування вибору відповідностей між ГВС, з властивими їй умовами й обмеженнями та СДК. Отже наведений нижче перелік НВКО є необхідним і достатнім для вирішення цієї задачі: *тип невизначеності,* *підхід до перепланування, стратегія перепланування, політика вибору часу, метод планування.*

***Класифікація СДК*.** При формуванні запропонованого НВКО було проаналізовано основні властивості системи динамічного керування (таблиця 2.1). Далі наводиться більш детальний опис та аналіз властивостей, що розглядаються.

Таблица 2.1. Класифікація СДК з необхідним НВКО

|  |  |
| --- | --- |
| Класифікаційна ознака (НВКО) | Класифікация СДК |
| Підхід до ДП | Реактивне, прогнозтично-реактивне, робастне  прогнозтично-реактивне та робастне превентивне. |
| Стратегія ДП | Повне перепланування, корекція плану |
| Політика вибору часу | Періодична, подієва і гібридна |
| Метод ДП | Правила диспетчеризації, евристики, метаевристики, ситуаційне управління, мультиагентні системи |
| Виды невизначеностей у ГВС | Пов’язані з ресурсами, пов’язані з задачами |

* + 1. Огляд підходів до ДП

Динамічне планування можна поділити на чотири категорії: реактивне планування, прогностично-реактивне планування, робастне прогностично-реактивне планування і робастне превентивне планування [, , , , ].

***Реактивне планування.*** При повністю реактивному плануванні жоден чіткий розклад не генерується заздалегідь і рішення приймаються локально в режимі реального часу. Для визначення пріоритету часто використовуються правила диспетчеризації. Правило диспетчеризації використовується для вибору для обробки наступної операції з найвищим пріоритетом з набору операцій, які очікують обслуговування на машині, що звільняється. Пріоритет операції визначається на основі параметрів операцій і машини. Правила диспетчеризації швидкі, зазвичай інтуїтивно зрозумілі та прості у реалізації. Проте, глобальне планування може значно підвищити продуктивність виробництва порівняно з локальними або короткостроковими правилами диспетчеризації, де важко передбачити продуктивність системи, оскільки рішення приймаються на локальному рівні в реальному часі. Детальний опис правил диспетчеризації представлено у підрозділі 2.3.

***Прогностично-реактивне планування.*** Прогностично-реактивне планування є найбільш поширеним підходом динамічного планування, що використовується у виробничих системах. Більшість визначень динамічного планування описаних в літературі стосуються прогностично-реактивного планування. Це є процес планування / перепланування, в якому розклад переглядається у відповідь на події в реальному часі. Прогностично-реактивне планування складається з двох етапів. По-перше, прогностичний графік генерується заздалегідь з метою оптимізації продуктивності виробництва без урахування можливих збоїв в цеху. Цей розклад модифікується під час виконання у відповідь на події в реальному часі.

***Робастне прогностично-реактивне планування.*** Більшість прогностично-реактивних стратегій планування засновані на простих корегуваннях у розкладі, які враховують тільки ефективність виробництва. Новий графік може значно відхилятися від початкового графіка, що може серйозно вплинути на планування інших видів діяльності, заснованих на початковому графіку і може привести до низької продуктивності графіка. Тому бажано генерувати прогностично-реактивні графіки, що є робастними. Робастне прогностично-реактивне планування спрямоване на побудову графіку, що мінімізував би вплив невизначеної ситуації на критерій продуктивності реалізованого графіка [, , ].

Навіть незважаючи на те, що необхідності створення робастного графіка була визнана більше двох десятиліть тому, у літературі небагато досліджень, щодо створення робастного графіка в динамічному середовищі. Типовим рішенням цієї проблеми є перепланування з одночасним урахуванням як ефективності виробництва так і відхилення від первісного графіка (стабільності). Стабільність вимірює відхилення від первісного прогностичного графіка [, , , ]. У [, ] визначено бі-критерії міри робастності для задачі перепланування у системі з однією машиною при виході з ладу машини. Критерії включають в себе мінімізацію тривалості виконання (ефективність планування) і впливу зміни розкладу (стійкість планування).

Для стійкості, вони досліджували дві міри: відхилення часу початку операцій від вихідних значень і відхилення від вихідної послідовності. Їх експериментальні результати показали ефективність робастних мір в тому, що стабільність планування може бути значно збільшена практично без втрат часу виконання. У [] автори також використовували дві міри, щоб визначити робастне планування: ефективність (час виконання) і міри стабільності (відхилення часу початку і відхилення послідовності). Мета планування полягає в тому, щоб максимізувати ефективність виробництва, і в той же час мінімізувати впливи у системі, викликані змінами розкладу. У [] автор досліджував різні міри робастності для поліпшення показників затримок і загального часу обробки при поломках машин.У [] було розроблено міри робастності і робастне планування для випадку з поломкою машин і змінним часом обробки у разі, коли для ремонту використовується стратегія зсуву вправо. Робастність визначається як мінімізація бі-критеріальної цільової функції вираженої з точки зору очікуваного часу обробки і очікуваної затримки. Очікувана затримка – це відхилення між детермінованим часом обробки до поломки, і фактичним часом обробки після застосування зміщення вправо.

Результати експерименту показали, що робастні графіки значно перевершують графіки на основі лише часу обробки. У [] визначені робастні міри для середовища з однією машиною для подолання невизначеності часу обробки, де мета планування полягає в мінімізації часу виконання операції. Робастність визначається мінімізацією як часу виконання так і абсолютного відхилення від первинного графіку. Результати розрахунків продемонстрували дієвість та ефективність запропонованих мір робастності. У [] визначили загальні міри корисності і стабільності, щоб визначити стратегію реакції на події в реальному часі, для побудови робастного графіка. Корисність вимірює зміну цільової функції планування після перегляду графіку. Це виражається різницею між значенням цільової функції нового графіка після реакції на події в реальному часі та цільовою функцією прогностичного графіку перед врахуванням подій в реальному часі. Потім вони дослідили ряд мір корисності та стабільності для моделі планування з однією машиною з метою мінімізації середнього часу завершення.

***Робастне проактивне планування.*** Робастні проактивні підходи планування зосереджені на створенні прогностичних графіків, які задовольняють вимогам до продуктивності в динамічному середовищі [, , ]. Основні труднощі цього підходу полягають у визначенні прогностичних мір. У[] запропонували прогностичну модель планування для середовища з однією машиною у випадку поломок з метою мінімізації максимального запізнення. Ефект руйнування вимірюється по відхиленню реального часу завершення роботи за реалізованим розкладом від запланованого часу завершення за прогностичним графіком. Відхилення зменшується вставлянням додаткового час у прогностичний графік з метою досягнення високої передбачуваності.

Обчислювальні експерименти показали, що прогностичне планування забезпечує значне поліпшення передбачуваності за рахунок дуже малого погіршення максимального запізнення. У [] автор розширив підхід прогностичного планування, де показником продуктивності розкладу стало запізнення виконання операцій.У [] також досліджували різні стратегії для генерації робастних проактивних графіків на основі введення тимчасового простою з метою зведення до мінімуму запізнення виконання операцій. Основна ідея полягає в забезпеченні здійснення кожної операції з додатково прогностичним часом обробки для поглинання певного рівня невизначеності без перепланування.

В табл. 2.2 зведено основні характеристики проаналізованих підходів з метою здійснення вибору останніх в умовах певної ГВС. Для відображення відповідності певних характеристик до розглянутих підходів використано наступні позначки: «+» - повністю відповідає, «+/-» - відповідає частково або на певних етапах, «-» - не відповідає.

Таблиця 2.2. Основні характеристики підходів до динамічного планування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Підхід до**  **планування** | **Наявність початкового плану** | **Мінімізація**  **відхилень від початкового плану** | **Прогностичне**  **врахування невизначеностей** | **Глобальна ефективність плану** | **Висока складність**  **критеріїв** |
| Реактивний | - | - | - | - | - |
| Прогностично-реактивний | + | - | - | +/- | - |
| Робастний  прогностично-реактивний | + | + | - | +/- | + |
| Робастний  превентивний | + | + | + | + | + |

Після вибору підходу до перепланування за наявності невизначених ситуацій в реальному часі необхідно вирішити ще дві проблем: коли і як реагувати на події в реальному часі. Перша проблема стосується часу здійснення перепланування, а друга проблема стосується визначення стратегій перепланування для реагування на події в реальному часі.

* + 1. Огляд політик часу ДП

Щодо проблеми, коли здійснювати перепланування, в літературі були запропоновані три політики поведінки [, ]: періодична, подієва і гібридна. Періодичні та гібридні політики часто визначаються як рухомий часовий горизонт [,, , , ] .

У ***періодичній політиці***, графіки генеруються через регулярні проміжки часу, за які збирається вся наявна інформація від виробничої системи. Динамічна задача планування розкладається на ряд статичних задач, які можна вирішити за допомогою класичних алгоритмів планування. Потім цей графік виконується і не переглядається, поки не почнеться наступний період, де горизонт планування оновлюється з урахуванням нової інформації, отриманої від поточного стану системи. Періодична політика дає більше стабільності розкладу і менше відхилень. На жаль, слідування встановленому графіку в умовах істотних змін стану виробничої системи може несприятливо позначитися на продуктивності. Визначення періоду перепланування також є складним завданням.

Основи застосування підходу рухомого горизонту до динамічного планування було запропоновано в []. Вони досліджували, як частота планування у динамічному виробничому середовищі впливає на продуктивність, де випадково можуть змінюватись час обробки і виникати поломки машин. На кожному періоді перепланування, за допомогою правил диспетчеризації генерується статичний графік поточних завдань. Як і слід було очікувати, продуктивність зазвичай погіршується, коли збільшується період перепланування. У [] автори використовували політику рухомого горизонту для задачі динамічного планування середовища з однією машиною з часом налаштування залежним від порядку для мінімізації максимального запізнення. Вони виявили, що за продуктивністю рухомий горизонт планування перевершує короткострокові правила диспетчеризації. У [] вивчалася частота перепланування в мульти-процесному гнучкому виробничому середовищі при поломках машин і варіації часу обробки. Продуктивність системи вимірюється по критеріям середнього запізнення і часу обробки. Їх результати дослідження частоти планування показали, що періодичної реакції з відповідною тривалістю періоду було б достатньо, щоб впоратися з подіями в реальному часі. Було відзначено, що поломки машин мають більш істотний вплив на продуктивність системи, ніж зміна часу обробки. У [] автори запропонували механізм, що періодично здійснює моніторинг системи і запускає правило диспетчеризації знайдене за допомогою моделювання роботи декількох правил. Якщо різниця між фактичною продуктивністю, що погіршується під впливом відмов машин, та очікуваною продуктивністю перевищує задану межу, то виконується нове моделювання для вибору нового правила диспетчеризації. Експерименти показують, що інтервал моніторингу та обмеження продуктивності мають бути ретельно визначені, щоб досягти кращої продуктивності.

У ***подієвій політиці*** перепланування спрацьовує у відповідь на несподівану подію, яка змінює поточний стан системи. Більшість підходів до динамічного планування використовують цю політику. У роботі [] дослідили політику подієвого перепланування для виробничих середовищ з виникненням поломок випадкових машин. Перепланування відбувається кожного разу, коли виникає поломка машини. Результати показали, що подієве перепланування з нижчим обчислювальним навантаженням і вищою передбачуваністю за продуктивністю перевершує політику періодичного перепланування та правила диспетчеризації. У роботі [] описані аналітичні моделі для оцінки ефективності системи з однією машиною при періодичній і подієвій стратегіях перепланування в умовах, коли задачі надходять динамічно. Вони запропонували оцінити роботу періодичного та подієвого перепланування за допомогою аналітичних моделей, які можуть легко і швидко оцінити важливі показники продуктивності, такі як середній час обробки і завантаженість обладнання. У [] автори розширили роботу дослідженням систем паралельних машини. Було показано, що частота перепланування може істотно вплинути на продуктивність системи (середній час обробки). Низька частота перепланування знижує кількість налаштувань. Більш висока частота перепланування дозволяє системі більш оперативно реагувати на збої, але може збільшити кількість налаштувань. Всі ці дослідження доводять, що подієве препланування набагато краще, ніж періодичне.

***Гібридна політика*** перепланує систему періодично, а також при виникненні виключних ситуацій. Події, що зазвичай розглядаються: поломки машин, надходження термінових задач, скасування задач, або зміна пріоритеів задач. У [] автори розробили гібридну подієву політику перепланування в середовищі з однією машиною і паралельними машинами з динамічними надходженням задач. Розроблена система здійснює перепланування періодично. Події, що класифікуються як періодично виникаючі між періодами перепланування ігноруються до наступного моменту перепланування. Проте, коли подія класифікується як термінова, негайно виконується повне перепланування. Результати показали, що продуктивність періодичного планування погіршується в міру збільшення тривалості періоду перепланування, у той же час подієвий метод досягає достатнього рівня продуктивності.

* + 1. Огляд стратегій ДП

Що стосується другого питання, а саме які стратегії необхідно використовувати для перепланування, у літературі вирізняються дві основні стратегії перепланування[, , ]: коректування плану і повне перепланування. Стратегія коректування плану відноситься до деякого локального врегулювання поточного розкладу і може мати перевагу завдяки потенційній економії процесорного часу і збереження стабільності системи. У Розділі 5.2 наводяться приклади стратегій коректування плану.

Стратегія повного перепланування передбачає регенерацію нового розкладу з нуля. Дана стратегія, в принципі, може бути кращою для підтримки оптимальних рішень, але ці рішення рідко досягаються на практиці і вимагають надлишкового часу обчислення. Крім того, повне перепланування може призвести до нестабільності і відсутності безперервності у деталізованих виробничих планах, що призводить до додаткових виробничих витрат, що називається виробничою нестабільністю.

У [] і [] зазначається, що більшість реактивних систем планування намагаються переглянути тільки частину первинно створеного розкладу для реагування на зміни у виробничому середовищі без регенерації нового плану з нуля. У роботі [] автори заявляли, що на практиці перепланування проводиться за допомогою корекції плану. У той же час повне перепланування у певній мірі також використовується. У [98] показано потенційну ефективність корекції розкладу з точки зору стабільності та процесорного часу в порівнянні з повним переплануванням.

Ще однією проблемою практичного значення є визначення того, чи варто здійснювати перепланування з нуля (повне перепланування), чи здійснювати корекції розкладу, і яку стратегію корекції плану обрати для того, щоб була можливість реагувати на події в реальному часі. Для вирішення даної проблеми, щоб оцінити ефективність стратегій перепланування і вибрати кращу стратегію, було застосовано моделювання та міри робастності. Для вирішення, яку стратегію перепланування краще застосовувати використовуються виміри робастності (ефективність і стабільність) [, , , , ]. У [, ] використовували міри корисності та стабільності для оцінки продуктивності різних стратегій корекції плану і повного перепланування, що допомагало обрати кращу стратегію перепланування.

* + 1. Огляд архітектур СДК

***Централізовані системи.*** Більшість систем планування та управління, розроблених в промислових умовах традиційно розглядалися з огляду на вертикальне поширення команд і відповідей, що значною мірою спирається на централізовані та ієрархічні моделі [83, 41, 105, 14, 106]. Для забезпечення узгодженості даних в масштабах всього підприємства, централізовані та ієрархічні системи планування (рис. 1 і 2) сильно залежать від центральних баз даних. Для оптимізації продуктивності рішення з планування робляться централізовано на рівні супервізора, а потім розподіляються на рівень виробничих ресурсів для виконання. Така архітектура покладає на центральний комп'ютер відповідальність за планування, диспетчеризацію ресурсів, моніторинг будь-яких відхилень і диспетчеризацію коригувальних дій.

Централізовані і ієрархічні системи планування мають ряд недоліків [83, 116, 105, 14, 16]. Основним недоліком є наявність центрального комп'ютера, який являє собою вузьке місце, яке може обмежити масштабованість виробництва і це єдина точка відмови, яка може його повністю зупинити. Крім того, зміна конфігурації ієрархічно керованих виробничих систем є дорогим і трудомістким процесом, оскільки передбачає заміну дорогого програмного забезпечення. Виробничі системи з ієрархічним плануванням набувають все більш складного рівня інтеграції компонентів. Іншим недоліком є те, що вертикальний потік інформації збільшує час прийняття рішень. Крім того практичний досвід показав, що ієрархічні централізовані системи планування, як правило, мають проблеми з реагуванням на відхилення і можуть неефективно реагувати на наявність подій в реальному часі. Коли виникає невизначена ситуація, інформація передається до вищого рівня ієрархії, і тільки після адаптації планувальника, новий графік ініціює новий потік команд, який формує реакцію на невизначеність. Це вертикальне переміщення інформації призводить до повільної реакції і, як наслідок, до низької надійності. Незважаючи на те, що централізовані та ієрархічні системи планування можуть забезпечити глобально кращі графіки в середовищах, де невизначеності в режимі реального часу виникають рідко, у динамічному середовищі у більшості випадків вони є неефективними. Таким чином, централізоване ієрархічне планування є надто складним, важко підтримуваним і переналаштовуваним, негнучким, дорогим і повільним, щоб задовольнити потреби сучасних складних виробничих середовищ.

***Розподілені системи.*** Глобальний конкурентний тиск у виробництві призвів до фундаментальних змін у роботі виробничих систем. Сучасні системи повинні швидко адаптуватися до впливу невизначеностей з підтримкою більш коротких виробничих циклів, підвищення продуктивності праці і збільшення експлуатаційної гнучкості. Для вирішення цієї проблеми застосовуються високо автоматизовані системи, що спроможні забезпечити надійність, стабільність, здатність до адаптації і ефективного використання наявних ресурсів за рахунок модульної та розподіленої архітектури [83, 16, 106]. Існує тенденція до зростання кількості розподілених виробничих систем в результаті необхідності підвищення інтенсивності реагування виробництва до змін ринків і технологій. Основною мотивацією при розробці цих систем є децентралізація контролю над виробничою системою і, як результат, зниження складності і вартості, підвищення гнучкості та відмовостійкості.

* + 1. Огляд методів диспетчеризації

***Правила диспетчеризації*** є одним з поширених методів диспетчеризації АТМ в реальному часі в динамічних умовах. Протягом багатьох років, у літературі була запропонована значна кількість простих і складних правил диспетчеризації. Такі правила диспетчеризації відображаються відносини між АТМ, як ресурсом, і набором деталей, що необхідно перемістити. В залежності від того, у який спосіб генерується запит на транспортування, правила диспетчеризації АТМ поділяються на дві категорії: правила ініційовані оброблюваними модулями і правила ініційовані транспортними модулями.

Правила ініційовані оброблюваними модулями генеруються з боку ГВМ для вибору між АТМ, що простоюють, наприклад найближчий АТМ, АТМ з найбільшим часом простою, найменше застосовуваний АТМ.

Правила ініційовані транспортними модулями генеруються з боку ГВМ для вибору, яку задачу виконати, наприклад найменший час або довжина шляху, найбільший розмір черги на обробку, найменший вільного розмір місця у черзі, модифіковане правило перший прийшов перший обслугував. Згідно з останнім правилом, коли ГВМ посилає запит на обслуговування АТМ, а воно не може бути здійснено негайно, зберігається час посилання запиту. Коли АТМ стає доступним, він направляється до ГВМ, що має найдавніше збережений час посилання запиту.

Було встановлено, що жодне з правил повністю не задовольняє всі критерії. У зв’язку з цим проводилось багато досліджень для визначення комбінації декількох правил диспетчеризації, для того щоб знайти ряд станів системи, в яких відносна продуктивність кожного правила є найвищою.

Таблиця 2.3. Правила диспетчеризації АТМ

|  |  |
| --- | --- |
| Vehicle initiated | Workstation initiated |
| Перший надійшов перший обслугований (*First-Come-First-Served – FCFS*) | Найвіддаленіший АТМ (*Farthest Vehicle – FV*) |
| Перший обраний перший обслугований (*First-Encountered-First-Served – FEFS*) | Перший доступний автомобіль (*First Available Vehicle – FAFS*) |
| Найбільший розмір черги (*Largest Queue Size – LQS*) | Найменший загальний час простою (*Least Cumulative Idle Time – LIT*) |
| Найбільший час між надходженнями (*Longest Inter-Arrival Time – LIT*) | Найменш використовуваний АТМ (*Least Utilized Vehicle – LUV*) |
| Найбільший час переміщення (*Longest Travel Time – LTT*) | Найдовший простій АТМ (*Longest Idle Vehicle – LIV*) |
| Найбільший час очікування (*LongestWaiting Time – LWT*) | Найбільший загальний час простою (*Most Cumulative Idle Vehicle – MIT*) |
| Максимальний попит (*Maximum demand – MD*) | Найближчий (вільний) АТМ (*Nearest (Idle) Vehicle – NV чи NIV*) |
| Максимальний розмір вихідної черги (*Maximum Outgoing Queue Size – MOQS*) | Випадковий АТМ (*Random Vehicle – RV*) |
| Мінімальний залишок вихідної черги (*Minimum Remaining Outgoing Queue – MROQS*) |  |
| Мінімум операцій в черзі (*MinimumWork-in-Queue – MWQ*) |  |
| Модифікований FCFS (*Modified First-Come-First-Served – MFCFS*) |  |
| Випадковий ГВМ (*RandomWorkstation – RW*) |  |
| Найкоротший час прибуття (*Shortest time to Travel First – STTF*) |  |

Для того щоб оцінити продуктивність різних правил диспетчеризації при різних динамічних і стохастичних умовах цеху, використовувалось моделювання. Моделювання дозволяє спробувати застосувати декілька правил диспетчеризації і обрати правило, яке дає найкращий результат. Моделювання правил диспетчеризації АТМ показує що у високонавантаженому виробництві переважають правила ініційовані транспортними модулями через малу ймовірність простою. Правила ініційовані оброблювальними модулями недостатньо ефективні в таких умовах.

Багато авторів використовували моделювання для оцінки ефективності правил диспетчеризації. Роботи [, ] показали докладні огляди правил диспетчеризації в дрібносерійному та крупносерійному виробництві. Вони оцінили продуктивність різноманітних правил диспетчеризації щодо деяких загальних критеріїв ефективності, які зустрічаються в літературі, таких як, зміна часу роботи, мінімальний і максимальний час роботи, середнє запізнення, максимальне запізнення, зміна запізнення і т.д. Вони класифікують ці правила на п'ять категорій:

* правила, що включають в себе час обробки;
* правила, які включають в себе строки виконання;
* прості правила, що не включають в себе ні час обробки, ні строки виконання;
* правила, пов'язані з умовами виробництва;
* правила, що включають два або більше з перших чотирьох категорій.

Було відзначено, що жодне одиничне правило не задовольняє всіх важливих критеріїв, що відносяться до часу роботи і запізнення виконання операцій. Загалом, було відзначено, що правила засновані на тривалості обробки, працюють краще в умовах високого навантаження, в той час як правила, засновані на терміні обробки працювати краще в умовах малого навантаження. У [] автор наводить аналіз на основі моделювання правил диспетчеризації для планування у виробничому середовищі з поломками машин. Що стосується критеріїв на основі часу обробки та кінцевого терміну, були оцінені відносна продуктивність добре відомих і нових запропонованих правил диспетчеризації для різних значень параметрів моделі. Результати показали, що відносна продуктивність правил планування може бути порушена при зміні параметрів поломок. У [] наводиться комплексне дослідження моделювання правил планування для гнучких виробничих систем за наявності різних рівнів поломок і змін в часі обробки. Він повідомив, що жодне правило не є найкращим при всіх можливих умовах. Також в статті представлене докладне дослідження літератури. У роботі [] автори на основі моделювання досліджували продуктивність ряду правил планування для динамічних виробничих систем. Їх мірою продуктивності був економічний критерий, який включав в себе основні витрати, пов'язані зі здійсненням планування.) У [] представлено всебічний порівняльний аналіз більше двадцяти правил диспетчеризації в динамічному виробничому середовищі зі зваженим критерієм запізнення для динамічного надходження завдань. У [] запропоновано систему планування на основі моделювання з двох основних компонентів: механізм моделювання та механізм реактивного управління. Механізм моделювання оцінює різні правила і вибирає найкраще. Механізм реактивного управління періодично контролює роботу системи і визначає термін початку нового моделювання. Також використовувалося моделювання та правила диспетчеризації для планування в режимі реального часу гнучкої виробничої системи за наявності надходження термінових задач, поломок машин та інструменту. Моделювання оцінює диспетчерські правила і на основі цієї оцінки вибирається краще правило диспетчеризації, яке задовольняє необхідні критерії. У дослідженні [] використовували моделювання для вивчення ефективності різних планових ремонтних евристик на основі перенаправлення задач на альтернативні машини (без зміни маршруту, зміна маршруту черги, зміна маршруту прибуваючих і зміна маршруту всіх) для несподіваних поломок машин в динамічних виробничих середовищах. Результати експерименту показали, що правильний вибір хорошої евристики корекції плану базується не тільки на характеристиках системи (завантаженість, час поломок машин і частота відмов машин), але і на швидкості роботи та кількості складових системи подачі матеріалу.

***Евристики*** в цьому контексті є проблемно-орієнтованими методами корекції графіку, які не гарантують знаходження оптимального розкладу, але мають можливість знайти досить хороші рішення в короткі терміни. Найбільш поширеними евристиками корекції графіку є:

* корекція графіку зсувом вправо;
* співпадаюча корекція графіка;
* часткова корекція графіка.

*Евристика зсуву вправо*, в разі відмови машини, здійснює зсув розкладу інших операцій вперед у часі на величину простою. *Евристика співпадаючої корекції графіка* здійснює перепланування зі збігом деяких моментів в майбутньому з попереднім графіком. *Часткова корекція графіка* переплановує тільки невдалі операції.

У роботі [] досліджували продуктивність *евристики зсуву вправо* у порівнянні з правилами диспетчеризації і повного перепланування за допомогою методу гілок та границь. Результати експерименту показали, що зміщення вправо перевершує за продуктивністю правила диспетчеризації і повне перепланування. У роботах [] і [] використовували евристику зсуву вправо для вставки часу простою для визначення прогностични графіків.

У роботі [] автори порівняли продуктивність *евристики часткової корекції графіка* (корекція планування невдалих операцій), повного перепланування і корекції графіка зсувом вправо відносно показників результативності (час виконання) і стабільності (відхилення від початкового графіку). Евристика часткового перепланування графіку перепланує лише ті операції, що безпосередньо або побічно піддалися впливу невизначеностей так, щоб мінімізувати як збільшення часу виконання, так і відхилення від початкового графіка. Результати показали, що евристика часткової корекції усуває більшу частину недоліків і обчислювальних складностей, пов'язаних з повним переплануванням і зсувом вправо. Зсув вправо дає гірші характеристики щодо часу виконання у зв'язку з тим, що цей спосіб є простим зміщенням графіку на величину відхилення. Таким чином, чим більше відхилення, тим більший очікуваний зсув, і тим більше зростання часу виконання.

У [] запропоновано *евристику співпадаючої корекції плану* для виробничої системи з декількома ресурсами за наявності поломок машин. Стратегія переплановує розклад в момент поломки машин так, щоб деякі моменти в майбутньому співпали з попереднім графіком. Їх експериментальні результати показали, що цей метод забезпечує практично оптимальні рішення при досягненні більш високої передбачуваності, ніж при повному переплануванні. Пізніше, у [] застосовували цей підхід для перепланування у крупносерійному виробництві. Результати показали, що евристика співпадаючої корекції є дуже ефективною з точки зору якості графіка, обчислювального часу і стабільності розкладу.

Також в літературі були запропоновані різні більш специфічні евристики корекції графіка. У [] автори запропонували кілька стратегій перепланування для зміни часу обробки, поломки машин і надходження нових операцій до виробничої системи. Існують наступні стратегії перепланування: перенаправлення задач до альтернативних машин, розділення задач (для серійного виробництва) і повне перепланування. У [] запропоновано кілька евристик корекції графіку за наявності поломок машин. Ці евристики корекції графіку ґрунтуються на перенаправленні задач до альтернативних машин. У [] розглянуто задачу мінімізації часу. Автори запропонували і оцінили роботу двох евристик корекції графіку, евристика корекції графіку із затримкою (затримує обробку партії щоб об’єднати надходження задач у найближчому майбутньому), і евристика корекції графіку з оновленням (оновлює час закінчення обробки задачі для затримки партії). Результати показали, що евристика має чудову середню продуктивність із посереднім обчислювальним навантаженням. У [] автори запропонували різні евристики корекції графіку для перепланування виробництва у гнучких виробничих системах для поломки машин, надходження термінових задач, підвищення пріоритету задач та скасування задач. При виникненні поломок машин, решта операцій виконуються на альтернативних машинах. При надходженні нових задач, якщо нова задача не термінова, то пріоритет призначається на основі EDD (Earliest Due Date – найменший строк виконання) або FCFS (First Come First Served – перший прийшов перший був обслугований) правил диспетчеризації, в іншому випадку присвоюється найвищий пріоритет і всі неочікувані задачі переміщаються вперед в часі. Коли пріоритет задачі збільшується або задача скасовується завдання, що залишилися зсуваються вперед у часі на відповідних машинах.

***Мета-евристики***: табу пошук, імітації відпалу та генетичні алгоритми.

*Мета-евристики* – евристики високого рівня, які керують евристиками локального пошуку для виходу з локальних оптимумів [, , ].

В останні роки, мета-евристики (табу пошук, імітація відпалу та генетичні алгоритми) успішно використовувалися для вирішення завдань складання виробничих розкладів. Евристики локального пошуку є методами пошуку околу. При локальному пошуку околу, пошук починається з деякого даного рішення, і намагається ітеративно рухатися до кращого рішення у відповідно визначененому околі поточного рішення з використанням рухомих операторів. Процес пошук припиняється, коли жодне краще рішення не може бути знайдено в околі поточного рішення, яке є локальним оптимумом. Мета-евристики, такі як табу пошук, імітації відпалу і генетичні алгоритми покращують алгоритми локального пошуку для уникнення локальних оптимумів або прийняттям неоптимальних рішень, або шляхом створення хороших початкових умов для локального пошуку в більш інтелектуальний спосіб, ніж просто надання випадкових початкових умов.

Пошук із заборонами, генетичні алгоритми та імітація відпалу широко використовуються для вирішення статичних детермінованих задач планування виробництва у виробничих системах. Проте, недостатньо дослідницької роботи було спрямовано на використання мета-евристик у динамічному плануванні. У [, ] розглянуто важливість використання мета-евристик для корекції розкладу замість використання локального пошуку або простих евристик, оскільки вони можуть потрапити у локальний оптимум.

***Табу пошук (ТП)*** — є мета-евристичним алгоритмом, який веде локальний пошук, що для запобігання потрапляння в пастку в передчасних локальних оптимумах, забороняє ті переміщення, які змушують повертатися до попередніх рішень і циклічної роботи. ТП починається з вихідного рішення. На кожній ітерації генерується окіл рішень, і найкраще з цієї околиці вибирається як нове рішення. Певні атрибути попередніх рішень зберігаються в табу-списку, який оновлюється в кінці кожної ітерації. Вибір кращого рішення в околиці відбувається таким чином, що він не приймає жодного з заборонених атрибутів. Краще допустиме рішення в даний час, оновлюється, якщо нове поточне рішення краще і допустиме. Процедура триває, поки не виконається будь-який з двох критеріїв зупинки, якими є максимальне число виконуваних ітерацій і максимальне число ітерацій, під час яких чинне рішення не поліпшується.

Запропонована Фредом У. Гловером схему локального пошуку має наступний вигляд. Розглянемо задачу дискретної оптимізації:

Нехай вдалося знайти деяке допустиме рішення *x0*. Розглянемо околи *N(x0)* ⊂ *D* точки *x0* і знайдемо рішення задачі:

.

Позначимо його через *x1*. Розглянемо тепер околи *N(x1),* знайдемо точку *x2* і т. д. до тих пір, поки *xk ≠ xk+1*. Якщо *xk=xk+1*, то точка *xk* є локальним оптимумом.

Ідея алгоритму ***пошуку з заборонами*** полягає в тому, щоб не зупинятися у локальному оптимумі, як це запропоновано в алгоритмах локального спуску, а продовжити пошук, керуючись тими ж правилами, забороняючи відвідування вже пройдених точок.

На *k*-му кроці алгоритму точка *xk+1* знаходиться з рішення задачі:

Множина виступає в якості списку заборон, що і послужило приводом для назви алгоритму. Цей список поповнюється на кожному кроці новою точкою. Очевидно, що якщо стара інформація не буде видалятися зі списку, то продуктивність алгоритму буде падати з ростом числа ітерацій. Тому довжина списку обмежується зверху деякою константою *l*, і список заборон містить тільки останні *l* точок. Перевірка і поповнення списку може виявитися досить трудомісткою операцією. Тому, іноді доцільно зберігати не самі рішення *xi,i=k-l+*1*,…,k*, а або функції від них (наприклад, значення цільової функції), або номера змінюваних координат, атрибути переходу від *xk* до *xk+1* . Позначимо через *N'(x)* частину околу *N(x)*, взяту випадковим чином. Тоді алгоритм пошуку із заборонами можна представити в наступному вигляді.

Алгоритм ТП.

1. Обираємо початкове рішення *x*, вважаємо *F*ПЗ:=*F(x)*. Формуємо порожній список заборон.

2. Знаходимо нове рішення *z* таке, що:

а) *z* ∈ *N'(x)*;

б) ;

в) перехід *x*→*z* не є забороненим або *F(z)* < *F*ПЗ.

3. Переходимо в нову точку *x* = *z.*

Якщо *F(x)* < *F*ПЗ, то змінюємо список заборон *F*ПЗ = *F(x)*.

4. Якщо виконано критерій зупинки, то STOP, інакше йти на п.2.

Як критерій зупинки використовується або зупинка по числу ітерацій, або необхідна точність по відношенню до заданої нижньої межі. Початкове рішення вибирається за допомогою якогось простого алгоритму. Змінюючи довжину списку заборон, можна керувати процесом пошуку. При зменшенні довжини, інтенсифікується пошук в поточній області, збільшення довжини сприяє переходу до іншої області. Як околиці можна розглядати безліч всіх рішень, які виходять з поточного рішення зміною однією з координат.

У [65] використовували ***пошук із заборонами*** для пошуку прогностичних графіків. У [28] використовували пошук з заборонами для корекції розкладу, викликаного невизначеністю часу обробки. У [129] використовували імітації відпалу для корекції розкладу для наземних операцій космічного шатла. Щоб скорегувати графік, система вибирає між п'ятьма евристиками корекції, використовуючи функцію вибору, і застосовуючи пошук за допомогою імітації відпалу для виконання декількох ітерацій корекції. Було встановлено, що пошук із заборонами та імітація відпалу генерують розклади хорошої якості в короткі терміни.

Загальну схему ***генетичних алгоритмів*** простіше всього зрозуміти, розглядаючи задачі безумовної оптимізації:

Прикладами служать задачі планування, розміщення, стандартизації, здійсненності та інші. Стандартний генетичний алгоритм починає свою роботу з формування початкової популяції – скінченного набору допустимих рішень задачі. Ці рішення можуть бути обрані випадковим чином або отримані за допомогою імовірнісних алгоритмів. Як ми побачимо нижче, вибір початкової популяції не має значення для збіжності процесу асимптотики, проте формування "хорошої" початкової популяції (наприклад з безлічі локальних оптимумів) може помітно скоротити час досягнення глобального оптимуму.

На кожному кроці еволюції за допомогою імовірнісного оператора селекції вибираються два рішення, батьки *i*1, *i*2. Оператор схрещування за рішеннями *i*1, *i*2 будує нове рішення *i'*, яке потім піддається невеликим випадковим модифікаціям, які прийнято називати мутаціями. Потім рішення додається в популяцію, а рішення з найменшим значенням цільової функції видаляється з популяції.

Завдання формалізується таким чином, щоб її рішення могло бути закодовано у вигляді вектора генів ("генотипу"). Де кожен ген може бути бітом, числом або якимсь іншим об'єктом. У класичних реалізаціях ГА передбачається, що генотип має фіксовану довжину. Однак існують варіації ГА, вільні від цього обмеження.

Деяким, звичайно випадковим, чином створюється безліч генотипів початкової популяції. Вони оцінюються з використанням "функції пристосованості", в результаті чого з кожним генотипом асоціюється певне значення ("пристосованість"), яке визначає наскільки добре фенотип їм описуваний вирішує поставлене завдання.

З отриманого безлічі рішень ("покоління") з урахуванням значення "пристосованості" обираються рішення (зазвичай кращі особини мають велику ймовірність бути вибраними), до яких застосовуються "генетичні оператори" (у більшості випадків "схрещування" - crossover і "мутація" - mutation), результатом чого є отримання нових рішень. Для них також обчислюється значення пристосованості, і потім проводиться відбір ("селекція") кращих рішень у наступне покоління.

Цей набір дій повторюється ітеративне, так моделюється «еволюційний процес», що триває декілька життєвих циклів (поколінь), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

* знаходження глобального, або субоптимального рішення;
* вичерпання числа поколінь, відпущених на еволюцію;
* вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми служать, головним чином, для пошуку рішень в багатовимірних просторах пошуку.

Таким чином, можна виділити наступні етапи генетичного алгоритму:

1. Поставити цільову функцію (пристосованості) для особин популяції.
2. Вибрати початкову популяцію *I*0 і покласти :

.

1. Якщо не виконаний критерій зупинки робити наступне.
   1. Вибрати батьків *i*1, *i*2 з популяції *Ik*.
   2. Схрещення: побудувати *i'* по *i*1, *i*2.
   3. Мутація: модифікувати *i'*.
   4. Селекція: Якщо *f\** < *f* (*I'*), то *f*\* = *f*(*I'*).
   5. Оновити популяцію і покласти *k* = *k* + 1.
2. Якщо виконуються умови зупинки, то кінець циклу, інакше до п.2.

У [18] використовувалися генетичні алгоритми для динамічного планування виробничих систем за наявності поломок машин і альтернативного маршруту вирішення задач. Були використані дві міри продуктивності, а саме: середня затримка виконання задачі і середня вартість виконання задачі. Щоразу, коли відбувається динамічна подія використовуються генетичні алгоритми, щоб запропонувати альтернативний графік. Крім того, рішення генетичних алгоритмів порівнювали з кількома поширеними правилами диспетчеризації. Результати показали, що продуктивність генетичних алгоритмів значно вище, ніж у поширених правил диспетчеризації. У [96] використовували генетичні алгоритми для динамічного планування гнучких виробничих систем. Вони розглядали наступні події в реальному часі: надходження нової партії, недоступність деталей для обробки(через відмову систем подачі, при наявності дефектів на заготовках і т.д.), і поломки машин (через відсутність інструментів, незаплановане обслуговування і т.д.). Розклади, отримані з використанням правил диспетчеризації були покращені з використанням генетичних алгоритмів. Результати показали, що генетичні алгоритми значно зменшили час виконання. У [58, 48] використовували генетичні алгоритми для генерації робастних графіків і для оцінки продуктивності різних мір робастності. У роботах [124, 125] порівняли продуктивність генетичних алгоритмів і евристик локального пошуку для генерації робастних графіків. Результати показали ефективність генетичних алгоритмів при генерації графіків з набагато кращим часом виконання і стабільністю, ніж евристики локального пошуку. Проте, у [12] автори повідомили у своїх експериментальних результатах, що можливості генетичних алгоритмів знижуються зі збільшенням розміру задачі, і вони не є ефективними для знаходження близьких до оптимальних рішень у встановлені терміни.

***Методи штучного інтелекту.*** У ряді задач динамічного планування застосовуються методи штучного інтелекту такі, як: системи знань, нейронні мережі, ситуаційне керування, нечітка логіка, сітки Петрі і т.д., які будуть розглянуті нижче.

Основна особливість ***підходів, заснованих на знаннях*** полягає у тому, що є велика різноманітність технічних знань щодо керування, яке може бути здійснене за наявності подій в реальному часі. Системи, засновані на знаннях зосереджуються на збиранні знань або досвіду експерта у конкретній області та використовують механізм виведення для отримання висновків або рекомендацій щодо здійснення корегуючих дій. Тому часто термін система, заснована на знаннях використовується в якості синоніма *експертної системи* [139].

Експертна система відрізняється від інших прикладних програм наявністю таких ознак:

1. Моделює механізм мислення людини при застосуванні для розв'язання задач в цій предметній області. Це істотно відрізняє експертні системи від систем математичного моделювання або комп'ютерної анімації. Однак, ЕС не повинні повністю відтворювати психологічну модель фахівця в цій області, а повинні лише відтворювати за допомогою комп'ютера деякі методики розв'язання проблем, що використовуються експертом.

2. Система, окрім виконання обчислювальних операцій, формує певні висновки, базуючись на тих знаннях, якими вона володіє. Знання в системі, зазвичай, описані деякою спеціалізованою мовою і зберігаються окремо від програмного коду, що формує висновки. Компонент збереження знань прийнято називати базою знань.

3. Під час розв'язання задач основну роль відіграють евристичні і наближені методи, що, на відміну від алгоритмічних, не завжди гарантують успіх. Евристика, в принципі, є правилом впливу (англ. rule of thumb), що в машинному вигляді відображає деяке знання, набуте людиною разом із накопичуванням практичного досвіду розв'язання аналогічних проблем. Такі методи є наближеними в тому сенсі, що, по-перше, вони не потребують вичерпної вихідної інформації, а, по-друге, існує певний ступінь впевненості (або невпевненості) в тому, що запропонований розв'язок є правильним.

Типові експертні системи можуть мати таку структуру (рис. 2.1):

* база даних (не обов'язкова);
* база знань;
* машина виведення (розв'язувач);
* підсистема пояснень;
* інтерфейс користувача;

База знань складається з правил аналізу інформації від користувача з конкретної проблеми. ЕС аналізує ситуацію і, залежно від спрямованості ЕС, дає рекомендації з розв'язання проблеми.

ЕС створюється за допомогою двох груп людей:

* інженери, які розробляють ядро ЕС;
* експерти за фахом.

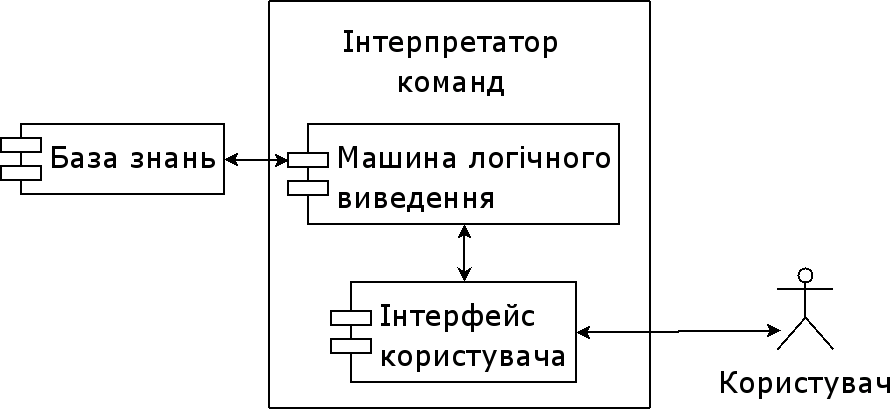


Рисунок 2.1. Структура ЕС

Представлення знань є однією з найважливіших функцій ЕС. Найбільш розповсюдженим способом представлення знань є правила (продукції). В їх основі лежить конструкція виду "ЯКЩО {умова}, ТО {наслідок}". Умовна частина називається антецедентом, а заключна - консеквентом. У загальному вигляді правило можна описати як:  де *n* – ім'я (ідентифікатор) продукції; *S* - характеристика сфери застосування продукції (область компетенції); *U* – умова застосування; *В*  *C* - ядро (власне конструкція "ЯКЩО; ТО"); *Р* - постумови продукції (котрі виконуються після її реалізації) [2, 81].

Таким чином, частина правил "ЯКЩО" має назву посилання, а "ТО" - виведення або дія. Антецедент зазвичай описується кон'юнкцією більш простих умов ; консеквент може складатися із кількох компонентів, при чому це можуть бути не лише факти, твердження, але і операції виклику спеціальних процедур. В останньому випадку правило має вигляд: "ЯКЩО { умова}, ТО {дія}". Наприклад, "ЯКЩО *В1, В2, ..., Вn*,, ТО С". Такий запис означає, що "коли всі умови від *В*1 до *Вn* являються істинними, то *С* також є істинним" або ж "коли всі умови від *В*1 до *Вn* являються істинними, то слід виконати дію *С*".

ISIS [34, 109], розроблений в Carnegie Mellon у 1982 році, була першою спробою використання системи, заснованої на знаннях у плануванні виробничої системи. ISIS виконує обмежений прямий пошук, для отримання розкладу. Динамічні ситуації обробляються переплануванням відповідних задач шляхом вибіркового послаблення деяких обмежень. OPIS [108] є наступником ISIS. OPIS це система, заснована на знаннях початково розроблена для планування промислового виробництва, яка використовує ймовірнісний підхід вирішення проблеми для поступової генерації і корекції графіка у відповідь на зміни. OPIS реалізовано на основі модульної архітектури, що поєднує багато різних евристик, що називаються джерелами знань, які вибірково використовується для генерації і перегляду загального графіка. Евристики корекції графіку визначені в OPIS:

* планувальник задач;
* планувальник ресурсів;
* зсувач вправо;
* зсувач вліво;
* планувальник попиту.

IOSS [82] є ще однією інтерактивною системою планування, заснованою на знаннях на основі ймовірністного та інтерактивного вирішення проблем. SONIA [55] це прогнозуюча реактивна система планування виробничої системи, заснована на знаннях. Були визначені різні евристики корекції графіку:

* послаблення термінів виконання;
* розширення робочих змін;
* відкладання роботи до наступної зміни;
* скорочення простоїв ресурсів за допомогою переставляння операцій.

Деякі дослідники поєднували системи, засновані на знаннях і моделювання у плануванні, для вибору найкращих корегуючих дій для обробки події в реальному часі [11]. Деякі системи, засновані на знаннях, були розроблені, щоб допомогти користувачеві реагувати інтерактивно на події в режимі реального часу [32, 101, 42, 74].

Серед інших методів штучного інтелекту, які були використані для вирішення проблеми динамічного планування є ***штучні*** ***нейронні сітки (ШНС)***.

*Штучні нейронні мережі* – це математичні моделі та їх програмні та/або апаратні реалізації, виконані за принципом організації біологічних нейросіток нервових клітин живого організму, являють собою систему з'єднаних і взаємодіючих штучних нейронів (простих процесорів) [139].

Для опису будь-якої нейронної сітки необхідно визначити основні положення:

* модель формального нейрона (функція перетворення тощо);
* структура зв'язків нейронів у сітці;
* архітектура побудови сіткових сполучень;
* метод навчання сітки.

Формально нейрон являє собою математичну модель простого процесора, який має декілька входів і один вихід. Вектор вхідних сигналів перетворюється нейроном (блок функції перетворення) у сигнал на виході з використанням трьох функціональних блоків:

* блок локальної пам'яті;
* блок підсумування;
* блок нелінійного перетворення.

Блок локальної пам'яті шляхом порозрядного зважування відповідними вагами *Wi* вхідних сигналів *Xi* формує вхідну функцію ШН.

Блок підсумування формує вхідний оператор ШН і задає вид використовуваного у ШН перетворення зважених входів:

* сума зважених входів ;
* добуток зважених входів ;
* максимальне значення зважених входів ;
* мінімальне значення зважених входів ;

*Активаційна функція*(функція перетворення)– це вхідний оператор нейрона, який задає вид використовуваного в ньому перетворення зважених входів. Найбільш розповсюджені активаційні функції:

1. сигмоїдна функція;
2. гаусівська функція;
3. інверсна мультквадратична функція;
4. трикутна функція;
5. трапацевидна функція.



Рис. 2.2. Формальне представлення функцій штучного нейрона

Формальне представлення функцій штучного нейрона наведено на рис. 2.2., де *х1, х2, х3, х4, x5* - вхідні величини; *w1, w2, w3, w4, w5* - величини синапсів (ваг), якщо сума , то нейрон збуджений, тобто *y* = 1.

При цьому для нейрона справджуються наступні кількісні співвідношення: вхідні величини *x1, х2, х3*,..., *ха*, рівні 1 або 0; вагові коефіцієнти *w1, w2, w3,..., wa* — будь-які дійсні числа.

Існуючі на даний час, нейромережі є групуванням штучних нейронів. Це групування обумовлено створенням з'єднанних між собою прошарків.

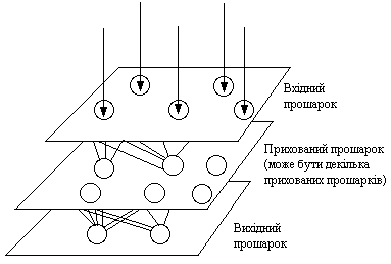


Рисунок 2.3. Діаграма простої нейронної мережі

На рис. 2.3 показана типова *структура* штучних нейромереж. Хоча існують мережі, які містять лише один прошарок, або навіть один елемент, більшість застосувань вимагають мережі, які містять як мінімум три нормальних типи прошарків - вхідний, прихований та вихідний. Прошарок вхідних нейронів отримує дані або з вхідних файлів, або безпосередньо з електронних давачів. Вихідний прошарок пересилає інформацію безпосередньо до зовнішнього середовища, до вторинного комп'ютерного процесу, або до інших пристроїв. Між цими двома прошарками може бути багато прихованих прошарків, які містять багато нейронів у різноманітних зв'язаних структурах. Входи та виходи кожного з прихованих нейронів просто йдуть до інших нейронів.

Напрямок зв'язку від одного нейрону до іншого є важливим аспектом нейромереж. У більшості мереж кожен нейрон прихованого прошарку отримує сигнали від всіх нейронів попереднього прошарку та звичайно від нейронів вхідного прошарку. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступних прошарків, забезпечуючи шлях передачі вперед (feedforward) на вихід.

Шлях, яким нейрони з'єднуються між собою має значний вплив на роботу мережі. Більшість пакетів професіональної розробки програмного забезпечення дозволяють користувачу додавати, вилучати та керувати з'єднаннями як завгодно. Постійно коректуючі параметри, зв'язки можна робити як збуджуючими так і гальмуючими.

Процес навчання може розглядатися як визначення архітектури мережі і налаштування ваг зв'язків для ефективного виконання спеціальної задачі. Нейромережа налаштовує ваги зв'язків по наявній навчальній множині. Властивість мережі навчатися на прикладах робить їх більш привабливими в порівнянні із системами, які функціонують згідно визначеній системі правил, сформульованої експертами.

Для процесу *навчання* необхідно мати модель зовнішнього середовища, у якій функціонує нейронна мережа - потрібну для вирішення задачі інформацію. По-друге, необхідно визначити, як модифікувати вагові параметри мережі. Алгоритм навчання означає процедуру, в якій використовуються правила навчання для налаштування ваг.

Існують три загальні парадигми навчання: "з вчителем", "без вчителя" (самонавчання) і змішана. У першому випадку нейромережа має у своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді як можна більш близькі до відомих правильних відповідей. Навчання без вчителя не вимагає знання правильних відповідей на кожен приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних та кореляція між зразками в навчальній множині, що дозволяє розподілити зразки по категоріях. При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час як інша визначається за допомогою самонавчання.

Широке обговорення методів нейронних мереж, зокрема вибір архітектури та методу навчання для використання у СДК можна знайти в [, , , , ].

***Нечіткі знання та системи нечіткого виведення***. Однією з найвизначніших властивостей людського інтелекту є спроможність приймати правильні рішення в умовах неповної та нечіткої інформації. Побудова моделей наближених міркувань людини та використання їх в інтелектуальних комп'ютеризованих системах є нині одним з найперспективніших напрямів розвитку сучасної обчислювальної техніки.

Якщо для класичного поняття множини характеристична функція в інтервалі (0, 1) набуває значення 0 або 1, то, поширивши поняття на нечіткі множини, говорять, що функція належності (ФН) в цьому випадку може набувати будь-яких значень в межах цього інтервалу [139]. В цій праці американський вчений Лотфі Заде визначив також ряд операцій над нечіткими множинами.

Запропонувавши поняття лінгвістичної змінної і припустивши, що її значеннями (термами) виступають нечіткі множини, Л. Заде запропонував апарат для подання процесів інтелектуальної діяльності, включаючи нечіткість і невизначеність висловлювань.

Математична теорія нечітких множин дає змогу подавати нечіткі поняття і знання, оперувати ними та робит нечіткі виведення. Нечітке управління є особиво корисним, коли досліджувані процеси є занадто складними для аналізу за допомогою загальноприйнятих методів або коли доступні джерела інформації інтерпретуються неякісно, неточно або невизначено. Характерним для нечіткої логіки і нечіткого керування є безпосереднє застосування експертних знань, які якісно формулюються для генерування керуючих впливів.

*Нечіткою множиною С* називається множина впорядкованих пар виду *C={MFc(x)/x}*, *MFc(x)* [0,1], де *x* - елемент неторої універсальної множини *X,* а *MFc(x)* - функція приналежності (Membership Function), що описує ступінь приналежності до нечіткої множини *C,* що представляє собою узагальнення поняття характеристичної функції звичайної множини. Значення *MFc(x)*=0 означає відсутність приналежності до множини, 1 - повну приналежність.

Проілюструємо це на простому прикладі. Формалізуємо неточне визначення "гарячий чай". В якості x (область міркувань) виступатиме шкала температури в градусах Цельсія. Очевидно, що вона буде змінюватися від 0 до 100 градусів. Нечітка множина для поняття "гарячий чай" може виглядати наступним чином:

*C*={0/0; 0/10; 0/20; 0,15/30; 0,30/40; 0,60/50; 0,80/60; 0,90/70; 1/80; 1/90; 1/100}.

Так, чай з температурою 60 С належить до множини "Гарячий" зі ступенем приналежності 0,80. Для однієї людини чай при температурі 60 С може виявитися гарячим, для іншої - не надто гарячим. Саме в цьому і проявляється нечіткість завдання (задания) відповідної множини.

Для нечітких множин, як і для звичайних, визначено основні логічні операції. Самими основними, необхідними для розрахунків, є перетин і об'єднання.

Перетин двох нечітких множин (нечітка "І"): *AB*:

*MFAB(x)*=*min(MFA(x), MFB(x))*.

Об'єднання двох нечітких множин (нечітка "АБО"): *AB:*

*MFAB(x)*=*max(MFA(x), MFB(x)).*

У теорії нечітких множин розроблено загальний підхід до виконання операторів перетину, об'єднання і доповнення, реалізований в так званих трикутних нормах і конормах. Наведені вище реалізації операцій перетину та об'єднання - найбільш поширені випадки *t-*норми і *t* -конорми.

Для опису нечітких множин вводяться поняття нечіткої та лінгвістичної змінних.

*Нечітка змінна* описується набором *(N, X, A),* де *N* - це назва змінної, *X* - універсальна множина (область міркувань), A - нечітка множина на *X.*

Значеннями лінгвістичної змінної можуть бути нечіткі змінні, тобто лінгвістична змінна знаходиться на більш високому рівні, ніж нечітка змінна. Кожна лінгвістична змінна складається з:

* назви;
* множини своїх значень, яка також називається базовою терм-множиною *T.* Елементи базової терм-множини являють собою назви нечітких змінних;
* універсальної множини *X;*
* синтаксичного правила *G,* по якому генеруються нові терми з застосуванням слів природної або формального мови;
* семантичного правила *P,* яке кожному значенню лінгвістичної змінної ставить у відповідність нечітку підмножину множини *X.*

Розглянемо таке нечітке поняття як "Ціна акції". Це і є назва лінгвістичної змінної. Сформуємо для неї базову терм-множину, яка складатиметься з трьох нечітких змінних: "Низька", "Помірна", "Висока" і задамо область міркувань у вигляді *X* = [100;200] (одиниць). Останнє, що залишилося зробити - побудувати функції належності для кожного лінгвістичного терма з базової терм-множини *T.*

Існує понад десяток типових форм кривих для завдання функцій приналежності. Найбільшого поширення набули: трикутна, трапецеїдальна і гауссова функції приналежності.

Трикутна функція приналежності визначається трійкою чисел (a, b, c), і її значення в точці x обчислюється згідно виразу:

|  |  |
| --- | --- |
| Треугольная функция принадлежности | (2.1) |

При (*b-a*)=(*c-b*) маємо випадок симетричної трикутної функції приналежності, яка може бути однозначно задана двома параметрами із трійки *(a, b, c).*

|  |  |
| --- | --- |
| *Трапецеидальная функция принадлежности* | (2.2) |

При (*b-a*)=(*d-c*) трапецеїдальна функція приналежності приймає симетричний вигляд (рис. 2.4).

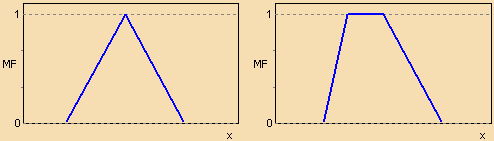


Рисунок 2.4. Типові кусочно-лінійні функції приналежності.

Функція приналежності гауссова типу описується формулою:

|  |  |
| --- | --- |
| Гауссова функция принадлежности | (2.3) |

Сукупність функцій приналежності для кожного терма з базової терм-множини T зазвичай зображується разом на одному графіку. Кількість термів в лінгвістичній змінній рідко перевищує 7.

*Нечітке логічне виведення.* Основою для проведення операції нечіткого логічного висновку є база правил, яка містить нечіткі висловлювання у формі "Якщо щось" і функції приналежності для відповідних лінгвістичних термів [139]. При цьому повинні дотримуватися наступні умови:

* Існує хоча б одне правило для кожного лінгвістичного терма вихідноїзмінної.
* Для будь-якого терма вхідної змінної існує хоча б одне правило, в якому цей терм використовується в якості передумови (ліва частина правила).

В іншому випадку має місце неповна база нечітких правил.

Нехай у базі правил мається m правил виду:

* *R* *1:* ЯКЩО *x1* це *A11* ... І ... *xn* це *A1n,* ТО *y* це *B1*
* ...
* *Ri:* ЯКЩО *x1* це *Ai1* ... І ... *xn* це *Ain,* ТО *y* це *Bi*
* ...
* *Rm:* ЯКЩО *x1* це *Ai1* ... І ... *xn* це *Amn,* ТО *y* це *Bm,*

де *xk, k* = 1 .. *n* - вхідні змінні; *y* - вихідна змінна; *Aik* - задані нечіткі множини з функціями належності.

Результатом нечіткого виводу є чітке значення змінної *y\** на основі заданих чітких значень *x k, k* = 1 .. *n.*

У загальному випадку механізм логічного висновку включає чотири етапи: введення нечіткості (фазифікація), нечіткий висновок, композиція і приведення до чіткості, або дефазифікації (рис. 2.5).

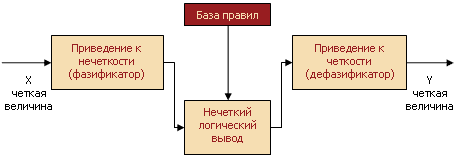


Рисунок 2.5 Система нечіткого логічного висновку.

Алгоритми нечіткого висновку розрізняються головним чином видом використовуваних правил, логічних операцій і різновидом методу дефазифікації. Розроблено моделі нечіткого висновку Мамдані, Сугено, Ларсена, Цукамото.

Розглянемо докладніше нечіткий висновок на прикладі механізму Мамдані (Mamdani). Це найбільш поширений спосіб логічного висновку в нечітких системах. У ньому використовується мінімаксна композиція нечітких множин. Даний механізм включає в себе наступну послідовність дій.

1) Процедура фаззифікації: визначаються ступені істинності, тобто значення функцій приналежності для лівих частин кожного правила (передумов). Для бази правил з m правилами позначимо ступені істинності як *Aik*(*xk*), *i*=1..*m,* *k* =1..*n*.

2) Нечіткий висновок. Спочатку визначаються рівні "відсікання" для лівої частини кожного з правил: http://www.basegroup.ru/files/image/library/analysis/fuzzylogic/math/f1.gif

Далі знаходяться "усічені" функції приналежності: http://www.basegroup.ru/files/image/library/analysis/fuzzylogic/math/f2.gif

3) Композиція, або об'єднання отриманих усічених функцій, для чого використовується максимальна композиція нечітких множин: http://www.basegroup.ru/files/image/library/analysis/fuzzylogic/math/f3.gif, де *MF (y)* - функція належності підсумковоїнечіткої множини.

Дефазифікації, або приведення до чіткості. Існує кілька методів дефазифікації. Наприклад, метод середнього центру, або центроїдний метод: http://www.basegroup.ru/files/image/library/analysis/fuzzylogic/math/f3.gif.

Геометричний зміст такого значення - центр ваги для кривої *MF (y).* Рис. 2.6 графічно показує процес нечіткого висновку по Мамдані для двох вхідних змінних і двох нечітких правил *R1* і *R2.*

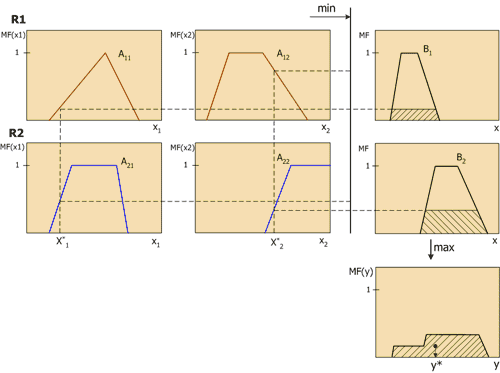


Рисунок 2.6. Схема нечіткого виводу по Мамдані.

Апарат нечітких множит, нечіткої логіки та механізми нечіткого виведення використовується у даній роботі.

Для отримання кращих динамічних систем планування, деякі дослідники розробили ***гібридні системи***, які поєднують різні методи штучного інтелекту.

У [] було розроблено фреймворк CABINS для корекції розкладу виробничого середовища на основі ситуаційного керування. Ситуації представляються змістом і відповідною корегуючою дією. Ситуаційне керування дозволяє зберігати і повторно використовувати знання для подібних ситуацій. Графік корегується поступово, в міру необхідності, з використанням ситуацій, що зберігаються в системі.

У [] та [] було розроблено гібридний фреймворк для динамічного планування, що складається з бази знань, яка описує правила диспетчеризації, модуля моделювання для оцінки продуктивності правил диспетчеризації, штучної нейронної мережі та генетичних алгоритмів, які дозволяють методами машинного навчання адаптувати підходи в окремих проблемних випадках.

У [] використовується ситуаційне керування і нечітка логіка для реактивного планування в сталеливарній промисловості. У [] використовується нечітку логіку для виявлення критичних задач для того, щоб перепланувати їх. В результаті, особа, яка приймає рішення, на виробництві отримує інформацію, які задачі повинні бути переплановані зараз, найближчим часом, пізніше або, можливо, взагалі ніколи. У [] використовувалася нечітка логіка для динамічного планування безперервного розливання сталі.

У [] була представлена система підтримки прийняття рішень на основі нечіткої логіки для планування і перепланування виробничого середовища з паралельними машинами за наявності невизначених ситуацій. Невизначеною ситуацією вважався брак матеріалу, що описувався числом невизначених подій і тривалістю ремонту. Ці параметри були визначені нечітко і моделювалися в поєднанні з використанням нечітких множин і нечітких множин другого рівня, відповідно. Нечіткі правила Сугено були запропоновані для визначення, коли і яким способом здійснювати перепланування.

У [, ] використовувалися системи на основі знань і нейронні мережі в реактивної планування. Нейронні мережі використовувалися для прийняття рішень щодо найкращого набору диспетчерських правил, за наявності подій в режимі реального часу.

У [] автор запропонував систему діагностики несправностей для реактивного планування на хімічних заводах. Система поєднує в собі процедуру адаптивного навчання нейронних мереж та експертної системи, заснованої на знаннях.

***Мультигентне динамічне планування.***

Існують вагомі докази того, що мультиагентні системи є одним з найбільш перспективних підходів до побудови складних, надійних і економічно ефективних систем планування виробництва наступного покоління з огляду на їх автономний, розподілений і динамічний характер та відмовостійкість [, , ]. Мультиагентна система являє собою мережу інтелектуальних агентів, які працюють разом задля вирішення проблем, які поза їх індивідуальними можливостями []*.* Використання мультиагентних систем для вирішення проблеми динамічного планування мотивується наступними ключовими моментами [, , , , , ]. По-перше, мультиагентні системи планування визначаються тим, що дані та управління розподілені по виробничій системі. Ці системи складаються з автономних агентів, приєднаних до кожної фізичної чи функціональної виробничої одиниці на виробництві (ресурсів, операторів, деталей, задач тощо). Локальна автономність дозволяє агентам брати відповідальність за проведення локального планування для одного або декількох об’єктів у процесі виробництва і локально та ефективно реагувати на локальні невизначеності, збільшуючи надійність і гнучкість системи. По-друге, ці окремі агенти мають значну свободу реагування на локальні умови, взаємодії один з одним для досягнення глобально оптимальних і надійних графіків. Загальна продуктивність системи глобально не планується, вона досягається через динамічну взаємодією агентів у реальному часі. Таким чином, система складається з паралельних незалежних локальних рішень агентів. По-третє, програмне забезпечення для кожного агента набагато менш об’ємне і простіше, ніж це було б для централізованого підходу, і в результаті воно простіше у написанні, установці і супроводі. Крім того, можна інтегрувати нові ресурси або видаляти існуючі зі своїми прикріпленими агентами для моделювання відповідного виробничого середовища без будь-яких змін в існуюче програмне забезпечення, просто підключивши їх до виробничої мережі.

***Архітектури планування на основі агентів***

Все більше число підприємств звертаються до агентної технології для роботи у складному і динамічному середовищі та досягають успішних результатів. Було реалізовано два основні типи мультиагентних архітектур для динамічного планування: автономні архітектури і посередницькі архітектури. Вони описані більш детально в наступних підрозділах.

***Автономні архітектури.*** В автономних архітектурах (рис. 2.7), агенти, що представляють виробничі об'єкти, такі як ресурси і задачі мають можливість визначити свої локальні графіки, реагувати на локальні зміни і безпосередньо співпрацювати один з одним для створення глобального оптимального і надійного розкладу.

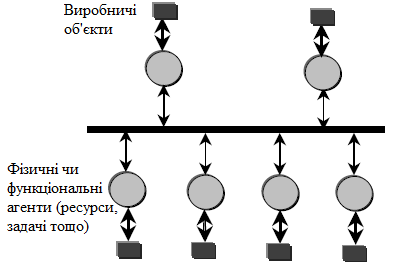


Рисунок 2.7. Автономні архітектури

Yams (Yet Another Manufacturing System) є однією з найбільш ранніх агентних виробничих систем, яка призначає агента для кожного вузла в ієрархії управління (виробництво, цех, виробничий модуль)[]. Основна ідея Yams в тому, що агенти завдань здійснюють переговори з агентами ресурсів для призначення завдань до виробничих модулів агентів за допомогою мережевого протоколу контракту (CNP – contract net protocol). У [] розробив систему динамічного планування у модульній виробничій системі. У своїй роботі агент виробничого модуля може одночасно працювати з іншими модулями через CNP []. Запит повідомлень про ставки поширюються по модулям, які оцінюють специфікацію операцій і подають заявки, що описують їх очікуваний час завершення або тривалість обробки. Модуль, який оптимізує визначений критерій обирається для виконання операції. У [] та [, , ] запропоновано просту мультиагентну архітектуру для динамічного планування в гнучких виробничих систем, включає тільки агенти ресурсів. Агенти ресурсів відповідають за динамічне планування ресурсів і вони жодним чином не контролюють один одного. Вони ведуть переговори за допомогою CNP для створення глобального розкладу. Кожен агент ресурс виконує наступні функції: планування, виявлення, діагностика та обробка помилок. Агенти ресурсів реагують локально в режимі реального часу подій, що відбуваються на відповідному ресурсі використовуючи коригувальні дії, описані у базі знань. Коли в режимі реального часу відбуваються події, такі як поломки машини, агент ресурс проводить нові переговори щодо пошуку альтернативного агенту ресурсу для поточної операції. Для підвищення гнучкості та надійності у [] автори запропонували мультиагентну архітектуру для динамічного планування у виробничих системах, яка включає агентів задач і ресурсів. Агенти задач здійснюють переговори щодо операцій задач з агентами ресурсів використовуючи CNP. Коли агент ресурс виявляє несправність, він посилає повідомлення про несправність машини до кожного агенту задачі, що є в черзі на обробку. Отримавши повідомлення про несправності машини, агент задачі повторно домовляється про виконання задачі з іншими агентами ресурсів, здатних її виконувати. У роботах [, , ] і [, ] було запропонували нову мультиагентну архітектуру для комплексного і динамічного планування у виробництва сталі. Кожен виробничий процес представлений агентом. Робастне прогностично-реактивне планування генерує надійні прогностично-реактивні графіки за наявності подій у реальному часі, використовуючи міри корисності, стабільності та надійності, а також різноманітні евристики перепланування.

Деякі мультиагентні системи планування використовують механізми торгів і ставок для переговорів між агентами. Агенти обмінюють ресурси на гроші за цінами, визначеними за допомогою ставок. У [, ] автори запропонували автономну мультиагентну архітектуру для динамічного планування виробництва на основі валютної моделі, що поєднує цілі планування і механізм цін. Їхня модель розглядає кожну задачу і ресурс в якості агента. Агенти задач ведуть переговори з агентами ресурсів через CNP механізм, щоб оптимізувати зважений критерій, який є функцією кінцевого часу виконання, ціни, якості та інших визначені користувачем параметрів. Агент деталі входить в систему з певною валютою, запитує та оцінює конкурсні пропозиції кількох агентів ресурсів, здатних виконувати вимоги до обробки і вибирає той, який оптимізує його критерій. Кожен агент ресурсу встановлює ціну на основі його статусу, потім він вирішує, яка з оголошених задач є найбільш цікавою для можливого подання пропозиції. Агент задачі намагається мінімізувати ціну, виплачену, а метою агента ресурсу є максимізація стягнутої ціни. Кожен контракт вважається завершеним після того, як агенти задач і ресурсів взаємно виконали роботу. Коли агент ресурсу вийшов з ладу, він інформує відповідний агент задачі, а останній переходить до процесу повторних переговорів щодо провалених задач з іншими агентами ресурсів. AARIA (Autonomous Agents for Rock Island Arsenal) являє собою автономну мультиагентну архітектуру розроблену для планування у оборонних виробничих об'єктах []. Виробничі ресурси, деталі і люди розглядаються як автономні агенти. Система включає в себе функції оптимізації розкладу та усунення несправностей. Агенти співпрацювати з використанням CNP.

Інші мультиагентні системи динамічного планування використовували навчальні підходи для динамічного планування. У [] запропонували динамічне планування виробництва із використання здатних до перенавчання агентів. Агент тренується за поліпшеним навчальним алгоритмом на стадії навчання, а потім успішно приймає рішення для планування операцій. Система планування складається з двох частин: середовища моделювання та інтелектуального агента. Агент вибирає найбільш відповідне правило пріоритету, для вибору задачі і присвоєння її ресурсу відповідно до виробничих умов, в той час середовища моделювання здійснює планування з використанням правила обраних агентом. У [] автор запропонував мультиагентні навчальні підходи для динамічного планування. У мультиагентній архітектурі, робочі площі контролюються агентами з базою знань, яка містить правила диспетчеризації і використовує навчання на основі генетичного алгоритму для оновлення правил у базі знань через певні проміжки часу. Вища частота навчання може допомогти агенту швидше адаптуватися до змін у виробничому середовищі.

***Медіаторні архітектури.*** Незважаючи на хороші показники автономних архітектур, як правило, вони стикаються з проблемами в забезпеченні глобально оптимізованих графіків і прогнозування за наявності великої кількості агентів [, , , , ]. Деякі дослідники запропонували медіаторні архітектури для динамічного планування у таких складних умовах щоб поєднувати надійність, оптимальність і передбачуваність.

Медіаторна архітектура має базову структуру, що складається з автономних співпрацюючих місцевих агентів, які здатні до переговорів один з одним з метою досягнення виробничих цілей [ , , ]. Ця основна структура розширюється медіаторами агентів щоб координувати поведінку місцевих агентів для виконання глобального динамічного планування, див. малюнок 4. Агенти медіатори працюють одночасно з локальними агентами над тими ж процесами прийняття рішень. Локальні агенти здійснюють прийняття рішень автономно, але можуть попросити поради у агента медіатора. Цей агент має можливість консультувати, нав'язувати або оновлювати рішення, прийняті агентами ресурсів з метою досягнення глобальних цілей і вирішення конфліктних ситуацій. Таким чином, агент медіатор може координувати і впливають на загальну поведінку системи. Агент медіатор має картину всієї системи, в той час як локальні агенти можуть мати більш детальну, більш правильну і більш актуальну картину локальних ситуацій. Таким чином, місцеві агенти можуть більш оперативно реагувати на невизначеності, в той час як агенти медіатори можуть координувати поведінку агентів і часто підвищувати глобальну продуктивність.

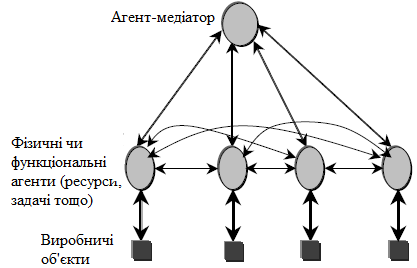


Рисунок 2.8. Медіаторні архітектури

Медіаторна архітектура забезпечують обчислювальну простоту, в той же час цілком підходять для розробки розподілених промислових систем, які є складними, динамічними і складається з великої кількості агентів ресурсів. [16, 14, 17] показали у своїх порівняльних дослідженнях, що медіаторні архітектури поліпшили продуктивність порівняно з автономними архітектурами, через їх здатність планувати далі в майбутнє, в поєднанні з їх здатністю реагувати на порушення, які можуть вплинути на глобальну продуктивність.

Дуже проста медіаторна архітектури була запропонована у [94] для динамічного планування в гнучких виробничих системах. Архітектура складається із агентів завдань, агента диспетчера задач, агентів ресурсів та агента медіатора ресурсу. Агент диспетчера завдань створює агентів завдань. Агент медіатор ресурсу веде переговори з агентами ресурсів з використанням CNP протоколу щодо виконання завдань. Коли відбуваються збої з ресурсами, повідомлення про провалені операції відправляються на агента медіатора ресурсів, який проводить повторні переговори з іншими агентами ресурсів. Цей режим обробки помилок досить є простим.

Для покращення надійності в складних виробничих системах, деякі автори запропонували інтеграцію агентів медіаторів на кожному виробничому рівні. У [64] запропоновано медіаторну архітектуру Metaphor І, для динамічного планування великих гетерогенних виробничих систем для вирішення задач, віртуальних підприємств, об'єднання підзадач і віртуальної кластеризації агентів. Задачі співпраці віртуального підприємства пов'язані з об'єднанням різнорідних виробничих підсистем у велике, динамічне віртуальне об'єднання скооперованих підсистем. Є два основних типи агентів в архітектурі: агенти ресурсів та агенти медіатори. Агенти ресурсів використовуються для представлення виробничого обладнання та операцій, в той час як агенти медіатори використовуються для координації агентів ресурсів за допомогою CNP протоколу. Збої у роботі агента ресурсів оброблюється на локальному рівні. Збій ресурсів моделюється шляхом введення періоду збою в ресурси. Кожне завдання, що включає період простою переплановується на інші доступні часові проміжки, в тому ж ресурсі (де виник збій) або в іншому ресурсі. На основі Metaphor I, у [106] розробили Metaphor II, для інтеграції інших видів діяльності виробничого підприємства, таких як проектування, планування, диспетчеризація, моделювання, обробка, постачання матеріалів, і маркетингових послуг. У цій архітектурі виробнича система організована через ієрархію підсистем медіаторів. Були введені чотири типи медіаторів: підприємства, ресурсів, маркетингу та дизайну. Кожна підсистема являє собою систему на основі агентів інтегровану в систему за допомогою спеціального медіатора. Агенти виробничих ресурсів координується відповідними медіаторами на всіх рівнях системи. Медіатори ресурсів високого рівня координують медіатори низького рівня, такі як медіатори машини, інструменту, робочих і транспорту. Співпраця між агентами ресурсів реалізується шляхом об'єднання медіаторного механізму і CNP протоколу. Кілька механізмів корекції графіку були розроблені для реагування на наявність подій у реальному часі, таких як: надходження нових завдань, скасування завдань, поломки машини і затримки в часі виконання завдання. У [113] розробили реактивну медіаторну архітектуру планування для реагування на зміни в завданнях і виробничих ресурсах. Виробничі ресурси, включаючи об'єкти і ресурси представлені агентами, які координуються двома медіаторами, а саме об'єктним і особистим, використовую CNP протокол. Реактивне планування здійснюється для зміни створеного розкладу у відповідь на зміни завдань, таких як скасування завдання або додавання термінового завдання, і виробничих умов, таких як поломки машин або раптової хвороби людини-оператора в процесі виробництва. Поєднання співпадаючої стратегії перепланування і агентів використовуються для корекції тільки частини початково створеного розкладу для підвищення ефективності реактивного планування, при збереженні якості планування.

***Порівняння методів планування***

Було проаналізовано кілька методів динамічного планування, серед яких: правила диспетчеризації, моделювання, евристики, мета-евристики, системи засновані на знаннях, нечітка логіка, нейронні мережі, гібридні методи і мультиагентні системи. У ході аналізу різних методів, було вивчено деякі публікації у яких наводиться їх порівняння. Ці порівняння дали розуміння того, які методи є найбільш придатними для динамічного планування. Переваги та недоліки цих методів були представлені у роботах [, , , ].

Правила диспетчеризація прості і швидко можуть знаходити прийнятні рішення. Проте їх основний недолік полягає в тому, що якість рішень, як правило, невисока через їх обмежену природу.

Евристики, через свою простоту, широко використовуються для реагування на наявність подій у реальному часі, але вони можуть зупинятися в локальному мінімумі. Щоб подолати цей недолік, були запропоновані мета-евристики, такі як пошук із заборонами, імітації відпалу і генетичні алгоритми. Було вивчено кілька порівняльних досліджень, щоб порівняти продуктивність пошуку із заборонами, генетичних алгоритмів і імітації відпалу. На відміну від імітації відпалу і пошуку із заборонами що маніпулюють одним допустимим рішенням, генетичні алгоритми працюють одразу з сукупністю допустимих рішень. Генетичні алгоритми було визнано неадекватними, через їх неефективність у знаходженні близьких до оптимальним рішень в розумні терміни у порівнянні з пошуком із заборонами та імітацією відпалу [, , , ]. Системи, засновані на знаннях мають потенціал у автоматизації мислення людей-експертів і евристичних знань для роботи виробничих систем планування. Вони моделюють виробничу систему за допомогою певного набору обмежень. Тим не менш, вони, як правило, не мають можливості для оптимізації системи і вимагають значних зусиль для створення і підтримки. Вони спрямовані на створення допустимих розкладів відповідно до доменних знань. З точки зору ефективності здатності до прийняття рішень, системи, засновані на знаннях обмежені якістю і цілісністю конкретних доменних знань. Потенціал нечіткої логіки до цих пір не повністю вивчені. Нейронні мережі не можуть гарантувати забезпечення оптимальних рішень, але їх здатність до навчання робить їх ідеальними для систем, що швидко змінюються. Інтеграція нейронних мереж, моделювання та експертних систем також є багатообіцяючою.

Більшість систем планування, розроблені у виробничих умовах централізовані та ієрархічної. Централізовані системи планування забезпечують послідовне глобальне відображення стану підприємства і в глобальному масштабі кращі графіки. Однак практичний досвід показує, що ці системи, як правило, мають проблеми з реактивним реагуванням на перешкоди. У даний час предметом багатьох досліджень є використання мультиагентних систем у динамічному плануванні. Основною мотивацією при розробці цих систем є децентралізація контролю виробничих систем, що призводить до зменшення складності, підвищення гнучкості та підвищення відмовостійкості. Замість традиційного уявлення про центральну систему планування, яка встановлює план виробництва для всіх машин і задач, мультиагентні системи припускають наявність декількох агентів зі значною автономією у прийнятті рішень, розподілених всередині виробничої системи. Агенти взаємодіють і співпрацюють один з одним з метою досягнення ефективних глобальних цілей. Локальна автономність дозволяє агентам приймати на себе відповідальність за проведення локального планування для одного або декількох функціональних або фізичних компонентів у процесі виробництва (наприклад, машин і завдань). Агенти мають можливість спостерігати за їх середовищем і спілкуватися та співпрацювати з іншими агентами з метою забезпечення того, щоб локальні розклади об’єднувалися у глобально допустимі графіки. Локальна автономність дозволяє агентам локально реагувати на локальні відхилення, збільшуючи міцність і гнучкість системи. Мультиагентні системи здатні успішно втілити такі можливості динамічного планування як децентралізація, інтеграції, надійність і гнучкість.

Кілька порівняльних досліджень наводять особливості мультиагентних систем, які роблять їх пріорітетними кандидатами для реалізації динамічного планування, на відміну від централізованих та ієрархічних систем планування. Результати досліджень [, ] показали, що мультиагентні системи добре підходять для застосувань на об’єктах, що є модульними, децентралізованими, часто змінюваними, погано структурованими, і складними. У роботах [, , ] автори представили у своїх порівняльних дослідженнях різні переваги мультиагентних систем, таких як гетерогенність, висока модульність, висока гнучкість, висока захищеність від збоїв, зниження складності, а також зниження витрат на розробку програмного забезпечення. Найбільш важливим моментом, який підтримує мультиагентні системи є реактивність: агенти можуть локально реагувати на локальні зміни швидше, ніж могли б централізованої системи. Мультиагентні системи забезпечують основу для створення архітектури, яка може покращити виробництво за рахунок підвищення надійності системи, ремонтопридатність, гнучкості, надійності та стабільності, а також надання можливості для планування в реальному часі і диспетчеризації. У [] наведені переваги використання мультиагентних систем у виробничому календарному плануванні, які забезпечують можливості інтеграції, надійності і реактивності, гнучкості, неоднорідності і автономності.

Було досліджено дві основні мультиагентні архітектури для динамічного планування: автономну і медіаторну архітектури. Автономні архітектури є досить поширеною формою контролю, де мережа незалежних агентів співпрацює безпосередньо для досягнення спільної мети. У медіаторній архітектурі, агенти співпрацюють через агента медіатора. Порівняльні дослідження [, , , , ] відзначають, що автономні архітектури мають перспективи зниження складності, цілісності, економічної ефективності, високої гнучкості і високої стійкості до перешкод. Вони добре підходить для застосувань з невеликою кількістю агентів. Тим не менш, у них є проблеми у забезпеченні глобально оптимальної продуктивності і поведінка системи може бути непередбачуваною в складних середовищах з великим числом агентів. На відміну від цього, медіаторні архітектури демонструють вищу продуктивність у порівнянні з автономними архітектурами у розробці розподілених виробничих систем, які є складними і складаються з великого числа агентів, таких як віртуальні підприємства. Вони поєднують в собі захищеність від перешкод з оптимізацією глобальної продуктивності і передбачуваністю.

* 1. Побудова логічної моделі поетапного синтезу СДК

***Визначення 2.2.*** *Модель поетапного синтезу СДК* – така послідовність перебирання її властивостей в просторі НВКО, яка дозволяє виокремити модель СДК, здатну задовольнити критерії обслуговування властивостей ГВС.

Наведена на рис. 2.9 схема відображає послідовність врахування властивостей згідно запропонованій НВКО при ітераційній процедурі синтезу СДК інтелектуальними агентами (ІА) у процесі функціонування МАПАВ. В завершенні процедури ІА проводять перевірку виконання умов обслуговуваності поточним вектором можливостей конкретної СДК вимог конкретної ГВС.

*Конкретна унікальна послідовність дій*, що приводить до мети, шукається агентом кожного разу з урахуванням відповідних *критеріїв обслуговуваності* властивостей розв’язуваної задачі.

***Визначення 2.*3.** *Критерій обслуговуваності* – показник задовольняння системою динамічного керування з боку ГВС.

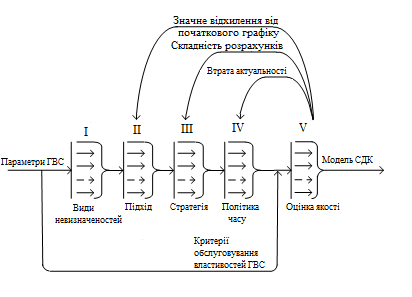


Рисунок 2.9. Логічна модель поетапного синтезу СДК

* 1. Побудова концептуальної моделі СДК

Обрана логічна послідовність синтезу СДК для здійснення динамічного керування у певній ГВС, використовується для визначення: виду невизначеності (ВН), підходу (ПДК), стратегії (СДК), політики часу (ПЧДК) та методу (МДК) динамічного керування. Склад і послідовність етапів синтезу повинні забезпечити набір ітераційних процедур, які й визначатимуть вибір такої моделі СДК, яка здатна адекватно відбивати *властивості та обмеження ГВС* (ВОГВС).

Для цього побудуємо так звану *віртуальну модель СДК*, що узагальнює в собі всі характеристики останньої, спроможні обслуговувати кожні з ВОГВС.

Використаємо за основу підхід, запропонований в роботі [137]. Згідно із цим підходом на узагальненому верхньому рівні абстрагування концептуально функція СДК (ФСДК-*функція*) (як і будь-якої організаційно-технічної системи) як *об’єкта керування* (ОК) подається декартовим добутком множин: .

|  |  |
| --- | --- |
|  | (2.4) |

Ф-функція відображає постулат діалектичного матеріалізму про існування матерії у просторі та часі у своїх різноманітних формах та проявах [136]. *Метод**-функцій* за якими установлюється відповідність множин рухомих об’єктів, технологій динамічного керування, що відбувається у просторових координатах ГВС у відповідні часові інтервали. Таке представлення може використовуватися як основа формалізації при кінематичному, геометричному, технологічному та структурному описах СДК. Отже метою синтезу/аналізу СДК є визначення складу та закономірностей організації окремих компонентів в єдину систему, з урахуванням того, що*функція останнього визначає його структуру*.

При виборі та дослідженні СДК вирізняють дві постановки задачі:

• *пряму* (*задача* *аналізу*) – відомі складові моделі СДК. Треба визначити ВОГВС – властивості та обмеження ГВС;

• *обернену* (*задача* *синтезу*) – відомі ВОГВС (властивості та обмеження прикладної ГВС). Треба визначити складові моделі ГВС, що адекватно реалізує ВОГВС.

Пряма задача виникає при перевірці можливості використання певної моделі СДК для обслугування властивостей конкретної ГВС, а обернена – при створенні нової або суттєвій модифікації існуючої моделі СДК під певні ВОГВС.

Отже, верхній рівень абстрагування визначає перебирання всіх варіантів складових моделі СДК.

За основу формалізації опису функціональної бази ОК – моделі СДК та її підфункціональних компонентів – елементарних складових (ЕлС) на виділених ієрархічних рівнях використовуються скінчені графи, що відображають через НВКО властивості та обмеження задачі з ОК і його складовими, а також метод Ф-функцій [136], за якими установлюється відповідність множин у виразі (2.4).

Отже, якщо загальна ФСДК-*функція* як ОК може бути подана на вищому ієрархічному рівні абстрагування виразом (2.4), то сукупність підфункцій складових процесів (Фсклад-*процеси*) перебирання на ***другому рівні абстрагування*** має вигляд: , , , , . Дана сукупність підфункцій призначена для варіативного формування траєкторій *tr* руху в межах змінних кожної з наведених складових до кінцевої мети – оптимального шляху *tr*opt. Іншими словами, варіативний перетин всіх підфункцій в процесі реалізації ЛМПС СДК являє собою багатоваріантну ієрархічну ітераційно-пошукову оптимізаційну задачу щодо послідовного перебирання складових моделі СДК, що задовольняють ГВС:

|  |  |
| --- | --- |
|  | (2.5) |

***Визначення 2.*4.** *Траєкторія tr руху* – це слід у послідовності етапів синтезу моделей СДК, що визначається перетином складових моделі СДК з показниками відповідності відповідно до ВОГВС [137].

Таким чином, вирази (2.5) являють собою проекції функціональних залежностей, що описують Фсклад-*процеси*, на відповідні координатні гіперплощини. Зокрема, серед виразів (2.5) є подання всіх складових, що утворюють моделі СДК і формують НВКО останніх.

***Визначення 2.*5.** *Оптимальна траєкторія tr*opt *руху* – слід у послідовності етапів синтезу моделей СДК, що визначається перетином складових моделей СДК з максимальними показниками відповідності до ВОГВС на кожному з етапів [137].

На подальших ***рівнях абстрагування (третьому – шостому)*** відповідно до моделі рис. 1 і залежностей (2.5) реалізується ітеративне перебирання можливих варіацій моделей СДК при відповідних фіксованих (за кращими результатами попереднього ієрархічного рівня) сполучень складових:

|  |  |
| --- | --- |
|  | (2.6) |

Виходячи з функції (2.4) послідовність реалізацій ФСДК може бути представлена орграфом (рис. 3), що являє собою ***нижній* *ієрархічний рівень*** подання функцій ОК.

Виділення цього рівня ієрархії, що характеризує узагальнену процедурну частину, є надзвичайно важливим етапом системного аналізу/синтезу моделі СДК. Синтезовані на цьому етапі типи функціональних сполучень Фсклад-*процесів* (, , , , ) є *необхідними і достатніми* для розв’язання прикладних задач побудови *системи автоматизованого вибору* структури СДК. На рис. 2 використано, відповідно, наступні скорочення: ВН – види невизначеностей (Р – пов’язані з ресурсами, З – з задачами), П – підходи до перепланування (Р – реактивний, ПР – прогностично-реактивний, РПР – робастний прогностично-реактивний, РП – робастний превентивний), С – стратегія перепланування (ПП – повне перепланування, КП – корекція плану), ПЧ – політика вбору часу перепланування (П – періодична, ПД – подієва, Г - гібридна), М – метод перепланування (ПД – правила диспетчеризації, Е – евристики, МЕ – мета евристики, СУ – ситуаційне управління, МАС - мультиагентні системи), СДК, що задовольняє умовам ВОГВС.



*Рисунок 2.10. Повний функціональний орграф ОК – процесу синтезу необхідної СДК*

Як вже вище наголошувалося, серед траєкторій *tr* процедурного руху за орграфом, наведеним на рис. 2.10, що відбиває реалізацію залежностей (2.6), є й оптимальні *tr*опт за умов відповідності до певних ВОГВС. Зокрема, на рис. 2 червоною лінією виділений слід умовно оптимальної траєкторії *tr*опт ум, яка в результаті багатоітераційного перебирання визначає складові умовної бажаної моделі СДК:

|  |  |
| --- | --- |
|  | (2.7) |

Подальше вдосконалення процесу синтезу СДК ГВС вбачається в автоматизації вибору параметрів шляхом використання інтелектуалізованих систем для здійснення послідовності ітераційних процедур перебору їх можливих варіантів з метою пошуку*tr*опт, що дозволить сформувати СДК здатну задовольнити ВОГВС.

**Висновки*.*** Динамічне керування є однією з ключових складових процесу керування виробничими системами, що включають виробничі ресурси і транспортні модулі. Пропонується використати узагальнену концептуальну модель та визначені в процесі її реалізації НВКО та ЛМПС для формалізації процесу синтезу СДК.

Визначено НВКО необхідний для формалізації процесу вибору методу динамічного планування і достатній для адекватного задоволення вимог обслужіваемості з боку розглянутої ГВС. В рамках НВКО проаналізовані види невизначеності в ГВС, чотири підходи до динамічного планування: реактивне, прогнозно-реактивне, робастне, робастне превентивне планування, основні політики часу і стратегії вирішення проблем оновлення розкладів, а також основні методи динамічного планування.

Запропонована узагальнена концептуальна модель та визначені в процесі її реалізації НВКО та ЛМПС чітко структурують процес синтезу СДК та дають можливість у подальшому здійснити автоматизацію цього процесу, що дасть змогу уникнути суб’єктивних факторів, пов’язаних з роботою оператора.

1. Мультиагентне середовище поетапного синтезу системи динамічного керування у гнучкій виробничій системі

Наведені у розділі 2 кроки щодо побудови узагальненої концептуальної моделі СДК дозволили отримати набір всіх можливих варіантів поєднань відповідних значень НВКО, серед яких шляхом багатоітераційного перебирання можна визначити складові умовної бажаної моделі СДК. Враховуючи значну кількість необхідних ітерацій та можливість впливу суб'єктивних людських факторів на даному етапі синтезу СДК виникає необхідність у його автоматизації.

Для цього необхідно розробити підхід до автоматизованого синтезу на основі набору ітераційних процедур, такої СДК, яка здатна адекватно задовольняти властивостям та обмеженням певної ГВС.

У [] була запропонована можливість використання мультиагентноїмоделі із вбудованим реляційним механізмом для перебирання на основі ЛМПС *функціонально-спеціалізованими інтелектуалізованими агентами* (ФСІА) критеріїв обслуговуваності з боку прикладних задач для вибору необхідної топології *штучної нейронної сітки* (ШНС).

Використаємо наведений у [] підхід, щоб на основі створеної у попередньому розділі *віртуальної моделі* [] створити строгу *узагальнену модель вибору* (УМВ) СДК, що базується на *гнучких інтелектуалізованих мультиагентних конфігураціях* (ГІМАК) *агентно-орієнтованих підсистем* (АОП) для конкретних ГВС.

* 1. Інтелектуалізовані агенти та мультиагентні системи
     1. Загальні риси агентно-орієнтованих систем

Однією з основних особливостей багатьох сучасних інформаційних систем є те, що вони не призначені для самостійного прийняття рішень. Імовірно, всі можливі варіанти поведінки таких систем мають бути спроектовані людиною і закладені в них на етапі розробки. Попадання подібної системи в умови, не враховані її розробниками, може призводити до аварійного завершення або більш тяжких наслідків. Одним з підходів, направлених на вирішення цієї проблеми, є застосування агентно-орієнтованого методу проектування та розробки.

Відмітною властивістю концепції агента (програмного агента) є наявність зовнішнього середовища, з яким агент здатний взаємодіяти, але не володіє можливістю його контролювати і тому завжди повинен бути готовий до того, що вжиті ним дії не призведуть до бажаних результатів. Ця властивість робить концепцію агента привабливим інструментом для вирішення багатьох завдань, у тому числі, для створення систем управління складними пристроями і комплексами.

Для сучасних інформаційних систем все частіше постає вимога здатності до оптимізації поведінки в умовах мінливого зовнішнього середовища, а також здатності до накопичення та аналізу досвіду, що, певною мірою, входить у конфлікт з жорсткими обмеженнями на час реакції і обчислювальну потужність обладнання. Крім того, для подібних систем особливо гостро стоїть питання коректності їх поведінки.

Іншим завданням, що виникає при розробці самоналагоджувальних мультиагентних систем, є реалізація методів накопичення та аналізу досвіду. Набір методів, запропонованих для вирішення цього завдання дуже широкий, активно розвивається донині і включає найрізноманітніші методи - від нейронних мереж до складних баз знань.

Однією із значущих тенденцій в сучасних інформаційних системах є паралелізація обчислень та інтеграція окремих інформаційних систем в більш потужні обчислювальні комплекси. Вплив цих тенденцій на розвиток самоналагоджувальних систем призвів до формування концепції соціальних агентів і мультиагентних систем. Перші роботи, які пропонують декомпозицію системи у вигляді набору автономних взаємодіючих сутностей, з'явилися в 70-х роках 20-го століття [117, 44, 57], проте термін агент вперше з'явився лише через десять років [37].

На сьогоднішній день мультиагентні системи використовуються для розробки широкого спектра інформаційних систем, серед яких умовно можна виділити три основні класи.

Відкриті системи. Системи, структура яких може змінюватися у процесі їх функціонування. Найбільша та відкрита на сьогоднішній день система - Інтернет. Соціальність і автономність агента дозволяють ефективно застосовувати його в якості елемента відкритої системи.

Складні розподілені системи. Найкращими сучасними методами боротьби зі складністю, є модульність і абстракція. Завдяки високому ступеню автономності, агент дозволяє зменшити кількість залежностей між частинами системи і, отже, спростити її проектування і реалізацію.

Інтерактивні системи. Більшість існуючих систем, незважаючи на графічний інтерфейс і потужну систему довідки, вимагають серйозних зусиль з боку потенційного користувача для їх освоєння. За допомогою агентів можна побудувати інтерактивну систему, яка буде не просто приймати і виконувати команди користувача, а активно й інтелектуально взаємодіяти з ним, прагнучи до досягнення спільних цілей.

У промисловості мультиагентні системи найбільш поширені в наступних областях.

*Автоматизація управління складними системами*. Область, в якій давно і ефективно застосовуються інтелектуальні агенти. В якості прикладів можна навести платформу ARCHON [118], систему управління виробництвом YAMS [84] і систему управління повітряним рухом OASIS [63].

*Збір та обробка інформації*. Агенти часто використовуються для реалізації систем, що збирають і обробляють інформацію з всесвітньої мережі Інтернет. Більшість сучасних пошукових машин реалізовано з використанням агентів.

*Ігри*. Сьогодні в комп'ютерних іграх противниками гравця людини часто стають гравці, реалізовані як інтелектуальні агенти.

Хороший огляд сучасних теоретичних і практичних підходів до побудови мультиагентних систем можна знайти в роботі [123]. Робота [67] містить більш критичний анліз, а також задається питанням, чому агентно-орієнтовані методи не настільки популярні, якими могли б бути.

Крім того, що мультиагентні системи активно застосовуються на практиці, вони формують великий напрямок сучасної науки, що активно розвивається. У різних частинах світу регулярно проводяться конференції та семінари, присвячені практичним і теоретичним аспектам розробки МАС.

Підходи до розробки мультиагентних систем можуть сильно відрізнятися в залежності від класу завдань, для вирішення яких ця система призначена.

* + 1. Формальна модель інтелектуалізованого агента

Поняття "агент" останнім часом було адаптовано до багатьох областей як прикладного та системного програмування, так і до досліджень в областях штучного інтелекту і розподілених інтелектуальних систем. Причому в кожному конкретному випадку поняттю надавалося дещо різне значення. У цьому розділі ми зафіксуємо поняття "агента", яке буде використовуватися далі в роботі.

Основною відмітною властивістю агентів є те, що агенти здійснюють дії. Далі часто стверджується, що агенти не просто здійснюють дії, але вони діють автономно і раціонально. Під автономністю зазвичай розуміють, що агент діє без прямого втручання людини чи іншої керуючої сутності.

З поняттям раціональності все не так просто, але часто під раціональністю розуміють прагнення агента оптимізувати значення деякої оціночної функції. Міра раціональності неявно вказує на те, що агент має цілі, яких агент "хоче" досягти, і уявлення про зовнішній світ, на які агент спирається при виборі дії.

Ще однією важливою властивістю агента є те, що він поміщений в зовнішнє середовище, з яким він здатний взаємодіяти. Як правило, середовище не контролюється агентом, тобто він здатний лише впливати на нього, але не контролювати повністю.

У підсумку, можна сформувати наступне визначення агента, адаптоване багатьма сучасними дослідниками.

***Визначення 3.1.*** *Агент* - обчислювальна система, поміщена в зовнішнє середовище, здатна взаємодіяти з ним, здійснюючи автономні раціональні дії для досягнення цілей.

Очевидно, що перераховані вище властивості не є достатніми для визначення інтелектуального агента, оскільки не передбачають явно гнучкості поведінки. Зазвичай для того, щоб вважатися "інтелектуальним" агент повинен володіти такими властивостями.

* Реактивність (reactivity) - агент повинен відчувати зовнішнє середовище і реагувати на зміни в ньому, здійснюючи дії, спрямовані на досягнення цілей.
* Проактивність (pro-activeness) - агент повинен показувати керовану цілями поведінку, проявляючи ініціативу, здійснюючи дії спрямовані на досягнення цілей.
* Соціальність (social ability) - агент повинен взаємодіяти з іншими сутностями зовнішнього середовища (іншими агентами, людьми тощо) для досягнення цілей.

Дамо визначення однієї з найбільш абстрактних моделей інтелектуального агента зі станом. У цьому випадку агент розглядається як набір

*AG* = *(S, A, env, I, refine, action),* де

• S є непорожня скінченна множина станів зовнішнього середовища;

• A є непорожня скінченна множина дій агента;

• env: *S* × *A* -> 2 *S* є функція поведінки зовнішнього середовища, що зіставляє поточному стану зовнішнього середовища і заданої агентом дії непорожню множину можливих наступних станів зовнішнього середовища. Таким чином, дії агента можуть впливати на навколишнє середовище, але не контролювати його повністю;

• *I* є непорожня скінченна множина внутрішніх станів агента;

• refine: *I* × *S* ->*I* є функція оновлення стану, що зіставляє попередньому внутрішньому стану і новому стану зовнішнього середовища новий внутрішній стан агента;

• action: *I* -> *A* є функція прийняття рішення, що зіставляє поточному внутрішньому стану агента деяку дію.

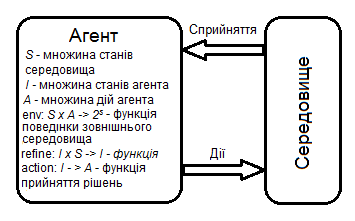


Рисунок 3.1. Модель агента зі станом

Опис агента в такому вигляді не дає ніякої інформації про його внутрішню структуру і зводить опис агента практично до моделі кінцевого автомата з вхідним алфавітом *S,* множиною станів *I* і вихідним алфавітом *A.* Далі в цьому розділі ми уточнимо структуру внутрішнього стану агента і процеси його перетворень.

Інтелектуальний агент володіє деякою інформацією про зовнішнє середовище, а також активно використовує цю інформацію при взаємодії із зовнішнім середовищем. Саме цю інформацію і називають знаннями або уявленнями агента. У разі ізольованого інтелектуального агента (діючого окремо, а не в складі мультиагентної системи) можна виділити два основних блоки уявлень:

• уявлення про поточний стан зовнішнього середовища, часто іменовані *сприйняття;*

• *уявлення* про закономірності поведінки зовнішнього середовища, що дозволяють агенту прогнозувати наслідки своїх дій;

*Сприйняття* описується відношенням , визначаючим для даного стану *s* 2 *S* множина станів невідмітних від s для даного агента:

Для моделювання *уявлень* агента про поведінку зовнішнього середовища введемо відношення . Якщо деяка трійка *(s, a, s ')* входить у відношення, то, за уявленнями агента, при виконанні дії *a* в стані зовнішнього середовища *s* зовнішнє середовище може перейти в стан *s'.*

Таким чином, уявлення агента за своєю структурою аналогічні опису поведінки зовнішнього середовища *env* і, по суті, є моделлю зовнішнього середовища з точки зору агента. Якщо для будь-яких  и виконано тотожність  тоді і тільки тоді, коли, можна сказати, що уявлення агента повністю відповідають реальності (в цьому випадку уявлення можна вважати знаннями). Множину всіх можливих уявлень позначимо:

.

Раціональність поведінки є основною властивістю агента, в той же час саме ця властивість найскладніше піддається формалізації. Багато дослідників ототожнюють раціональність з керованою цілями поведінкою, маючи на увазі, що агент не просто взаємодіє із зовнішнім середовищем, а намагається при цьому досягти деяких цілей.

До моделюванню поняття мети агента також існує кілька підходів. Найпростішим прикладом мети може служити деяка підмножина станів зовнішнього середовища , одного з яких агенту необхідно досягти. Розширеними цілями часто називають множину кінцевих ланцюжків станів зовнішнього середовища *.* Розширена мета вважається досягнутою в тому випадку, якщо історія взаємодії агента з зовнішнім середовищем має кінцевий префікс, стан зовнішнього середовища в якому збігається зі станами одного з ланцюжків множини *G*.

Одним з найбільш загальних описів мети агента є опис функції-критерію goal: *(P* × *Bel (S, A))* \_! {completed, continue, failure}, співставляючої ланцюжку пар *(p, bel)* 2 *P* × *Bel (S, A)* одне з наступних значень:

• completed в тому випадку, якщо мета остаточно досягнута;

• continue в тому випадку, якщо досі історія взаємодії відповідала меті, але остаточно мета не досягнута;

• failure в тому випадку, якщо мета не була досягнута і не може бути досягнута в майбутньому;

На вхід функції-критерію подається частина історії взаємодії агента з середовищем, що містить інформацію про сприйняття відповідних станів зовнішнього середовища та інформацію про уявленнях агента на даний момент (ця частина необхідна для моделювання епістемологічних цілей). Результатом функції є прапор, що показує досягнута мета і чи можливо її досягнення в майбутньому.

Позначимо множину всіх можливих функцій-критеріїв для множини сприйнять *P,* множини станів зовнішнього середовища *S* і множини дій агента *A* як . У цьому випадку бажання агента є кінцевою підмножиною множини всіх можливих функцій-критеріїв.

Одним з характерних властивостей інтелектуального агента є проактивність, що передбачає здатність агента до побудови планів взаємодії із зовнішнім середовищем.

План агента можна розглядати як кінцевий *автомат:*

*plan = (P, A, I pln,* ** *pln, i pln, 0),* де

• *P* є вхідний алфавіт автомата, що співпадає з множиною можливих сприймань агентом станів зовнішнього середовища;

• *A* є вихідний алфавіт автомата, що співпадає з множиною дій агента;

• *Ipln* є множина внутрішніх станів автомата, що є частиною множини внутрішніх станів агента *I* в тому сенсі, що *I = I pln × I';*

• ** *pln* ** *P × I pln × I pln × A* є відношення переходів, що визначає по сприйняттю p поточного стану зовнішнього середовища і поточного внутрішнього стану плану *i pln* наступний стан для плану *i 'pln* і дію, яку слід виконати агенту;

• *i pln, 0* ** *I pln* є початковий стан автомата.

Можливим результатом плану plan в початковому стані зовнішнього середовища *s0* і початковому стані *i0,pln = i0|Ipln ∈ Ipln* за уявленнями агента *bel = i0|Bel(S,A) ∈ Bel(S,A)* назвемо множину ланцюжків *outplan,bel(s0, i0,pln) = {λ ∈ (S × Ipln × A)+},* де для кожного ланцюжка *λ ∈ outplan,bel(s0, i0,pln)* виконано, *λ [0]|S = s0 и λ [0]|Ipln = i0,pln*  і для будь-якого *j <|* ** *|* якщо існують такі *s ∈ S, a ∈ A* і *I’pln ∈ Ipln*, що *(λ [j]|S, a, s) ∈ bel* і *(see(λ [j]|S), λ [j]|Ipln, I’pln, a) ∈ σpln*, то *j +1 < |λ|* і існує таке *a’ ∈ A,* що

• *(λ [j]|S, a’, λ [j + 1]|S) ∈ bel*.

• *(see(λ [j]|S), λ [j]|Ipln, λ [j + 1]|Ipln, a’) ∈ σpln*.

Зауважимо, що результат плану за уявленнями агента може відрізнятися від реальних результатів плану в тому випадку, якщо уявлення агента розходяться з реальністю.

Будемо говорити, що план *plan* реалізує мету *goal* за уявленнями агента *bel* ** *IB* в поточному стані середовища, що сприймається як *p0,* якщо для будь-якого ланцюжка ** ** *out (plan, bel, p0)* виконано *goal |P×IB) ≠ failure.*

Множину всіх можливих планів позначимо як 2Plan. У тих випадках, коли це не буде вести до двозначності, будемо ототожнювати план з його відношенням переходів.

План агента не є статичним - при зміні бажань чи уявлень агента він може бути перебудований. Цей процес ми змоделюємо функцією оновлення плану (plan revision function): *prf: IB × ID × 2Plan × P –> 2Plan*

співставляючу поточним уявленням агента *bel* ** *IB,* поточним бажанням агента *des* ** *ID,* поточному плану *plan ∈ 2Plan* і сприйняттю поточного стану зовнішнього середовища *p* ** *P* новий план *plan’ ∈ 2Plan.*. У більшості випадків новий план буде збігатися зі старим, проте, якщо бажання або подання агента радикально змінилися або зміни, що відбулися у зовнішньому середовищі, не були передбачені вихідним планом, може бути побудований новий план.

Крім того, з кожним планом можна асоціювати деяку множину цілей *int* ** *Goal(P, S, A),* які реалізуються планом за поданнями агента і які містилися в множині бажань агента *des* на момент побудови плану. Такі цілі ми будемо називати намірами агента.

* + 1. Формальна модель мультиагентної системи

Мультиагентна система представляється трійкою *MAS = (S, AG, env),* де

• *S* є кінцева множина станів зовнішнього середовища;

• *AG = {ag1, . . . , agn}* є скінченна множина агентів, кожен з яких представлений розширеною моделлю інтелектуального агента;

• *env : S×Aag1 ×. . .×Aagn –> 2S* є функція, що описує можливу

реакцію зовнішнього середовища на дії всіх агентів системи. Множину всіх можливих спільних дій системи позначимо *ACS = Aag1 × . . . × Aagn*.

Оперативна комунікація - це вид комунікації, який використовується агентами для координації своїх дій у поточний момент. В рамках оперативної комунікації агентам необхідно обмінюватися відносно невеликими обсягами інформації. Для моделювання цього виду комунікації найкраще підійде модель сигналів.

Для кожного з агентів *agi* системи визначено скінченну множину сигналів *Sigagi,* а також функцію Sendagi: *Pagi × AG –> Sigagi,* що описує які сигнали агент agi надішле кожному з інших агентів в поточній ситуації. Зауважимо, що кожен з агентів може послати кожному іншому агенту не більш, ніж однин сигнал. Для моделювання ситуації, коли агент не посилає сигнал іншому агентові, введемо виділений фіктивний *сигнал sg° ∈ Sigagi*. Поведінка функції Sendagi змінюється в процесі взаємодії із зовнішнім середовищем, а послані і отримані сигнали впливають на прийняття іншими агентами рішення про дію. Отже, функцію Sendagi зручно моделювати в контексті поточного плану агента. У цьому випадку план агента представляється як взаємодіючий кінцевий автомат

*planagi = (Pagi ,Aagi , Sigag1 ×. . . × Sigagn*, *Sigagi*, *Ipln*, *sendpln*, *σpln*, *ipln*, *0)*, де

• *Pagi* є вхідний алфавіт автомата, що співпадає з множиною можливих сприймань агентом agi станів зовнішнього середовища;

• *A agi* є вихідний алфавіт автомата, що співпадає з множиною дій агента agi;

• *Sigag1 ×.* *.* *.* *× Sigagn* є множина вхідних сигналів автомата, що збігається з декартовим добуком множин сигналів всіх агентів системи;

• *Sigagi* є множина вихідних сигналів автомата, що збігається з множиною сигналів агента AG I ;

• *Ipln*є множина внутрішніх станів автомата;

• *sendpln : Pagi × Ipln × AG –> Sigagi*є функція посилки сигналів, що визначає за поточним сприйняттям зовнішнього середовища і станом автомата сигнал, який буде посланий кожному з агентів системи;

• *σpln ⊆ Pagi × Sigag1 × . . . × Sigagn × Ipln × Ipln × A* є відношення переходів, що визначає по сприйняттю *p* поточного стану зовнішнього середовища, набору отриманих сигналів *(sg1, . . . , sgn) 2 Sigag1 × . . . × Sigagn* та поточному внутрішньому стану плану *ipln* наступний стан плану *i'pln* і дію, яку слід виконати агенту;

• *ipln,0 ∈ I*pln є початковий стан автомата.

Таким чином, рішення про вибір дії і зміну стану автомата відбувається в два етапи.

1. Визначення та розсилка сигналів - на цьому етапі кожен з агентів системи визначає за своїм сприйняттям поточного стану зовнішнього середовища і поточного стану свого плану, які сигнали він повинен послати іншим агентам і розсилає їх.

2. Вибір дії - на цьому етапі, ґрунтуючись на сприйнятті поточного стану зовнішнього середовища, поточний стан плану й отриманих від інших агентів сигналах, агент вибирає свою дію і наступний стан для свого плану.

Запропонована модель комунікації досить проста в реалізації і дозволяє забезпечити швидке прийняття кожним з агентів системи рішення про поточну дію. Зауважимо, що в запропонованій моделі агент може відправити сигнал сам собі, в чому, як правило, не виникає необхідності. Ця можливість збережена з метою спрощення опису.

Мультиагентний підхід надалі застосовується у роботі для реалізації автоматизованого синтезу СДК для перебору інтелектуалізованими агентами критеріїв обслуговуваності з боку ГВС. Також мультиагентний підхід розглядається як один з основних методів здійснення динамічного керування роботи АТМ у ГВС.

* 1. Гнучка інтелектуалізована система вибору СДК на основі концепції нечіткої метаідентифікації
     1. Концепція нечіткої метаідентифікації вибору СДК

Основна ідея запропонованого підходу полягає у тому, щоб, розглядаючи нечіткі ідентифікуючі компоненти як складні *агентно-орієнтовані підсистеми* (АОП), використати переваги нечіткого підходу до процесу ідентифікації моделі СДК самими АОП.

***Визначення* 3.2.** *Метаідентифікеація СДК* – це ітераційна процедура синтезу такої СДК, яка виявиться спроможною найкращим чином задоволняти умови обслуговуваної ГВС.

При цьому *завдання нечіткої метаідентифікації* полягає в динамічному побудуванні з існуючих *функціонально-спеціалізованих інтелектуалізованих агентів* (*ФСІА*)таких *гнучких інтелектуалізованих мультиагентних конфігурацій* (*ГІМАК*) *АОП,* які найкращим чином задовольняють умови обслуговуваної ГВС.

***Визначення 3.3*.** *Інтелектуалізований* *агент/мультиагент* – це програмно-апаратний або програмно-емуляційний автономний компонент АОП, який функціонує за певним сценарієм/алгоритмом на основі КУПД в інтересах досягнення поставленої користувачем перед нею мети.

***Твердження 3.*1.***Алгоритм дії ФСІА* може мінятися і коректуватися по ходу виконання завдання заради досягнення мети.

***Визначення 3.4*.** *Агентно-орієнтована підсистема* – це складна підсистема ГІМАК, в якій функціонують два або більше ФСІА, орієнтованих на розв’язання задач ідентифікації за певною вирішальною класифікаційною ознакою СДК і утворюючих *агентно-орієнтоване середовище*.

***Визначення 3.5*.** *Функціонально-спеціалізований інтелектуалізований агент* – це ІА з функціями метаідентифікації в просторі *вирішальних класифікаційних ознак СДК*.

Реалізація вищеозначеної ідеї можлива за допомогою побудування ГІМАК, особливостями якої є: використання об’єктно-орієнтованих ФСІА всіляких типів, реалізуючих складові класифікатора СДК; високий ступінь паралелізму; децентралізована структурна і параметрична метаідентифікація в межах АОП (рис. 3).

Для узагальнення підходу розглядається більш складний випадок нечіткої метаідентифікації класифікаційних ознак СДК щодо вимог з боку прикладної задачі. В разі чіткої взаємозалежності “вимога – ознака” реалізація процесу спрощується.

***Визначення* 3.6.** *Гнучка інтелектуалізована мультиагентна конфігурація* – така мультиагентна конфігурація, яка: містить агенти  з *функціями метаідентифікації*, які реалізують механізм розподіленого динамічного виявлення “ступеня важливості” інших агентів із всілякою природою; формує різні закони ідентифікації; забезпечує паралельність роботи агентів різнорідних “шарів”; реагує на зміни стану зовнішнього середовища (вихідних умов задачі) шляхом піднастроювання загального виходу у відповідності з ідентифікацією, задовольняючою поточний набір умов на вході.

***Визначення 3.7*.** *Агенти з функціями метаідентифікації* – такі ФСІА, які спроможні приймати рішення щодо: активації інших агентів ГІМАК; формуванню виведень щодо задоволення поточного набору умов на вході ГІМАК.

***Твердження* 3.2.** Сукупність певним чином організованих ФСІА, забезпечуючих визначений НВКО рівень ідентифікації СДК, утворюють *мультиагентне* *середовище* відповідної АОП.

Мультиагентна структура ГІМАК АОП (рис. 2.10) формується з множини  зв’язаних між собою ФСІА. На вхід АОП від зовнішніх джерел (в тому числі, і від користувача) надходить множина  значень вхідних змінних  які відображують умови обслуговуваності властивостей (інформаційне поле) прикладної задачі.

Фаззі (*F*)-перетворювач (“чіткий → нечіткий”) трансформує *Ux* у множину факторів  (є нечіткими множинами, заданими на значеннях вхідних змінних) з відповідними факторами достовірності (ступенями приналежності за експертними оцінками) 

Дефаззі (*D*)-перетворювач (“нечіткий → чіткий”) трансформує множину своїх вхідних факторів  і відповідних факторів достовірності  у множину  значень умов сумісності  пропоновуваної ГІМАК АОП моделі СДК із заданим на вході набором властивостей ГВС. (детально процедури фаззі-перетворення наведено в розділі 2.2.).



Рисунок 2.10. Структура ГІМАК АОП

* + 1. Компоненти ГІМАК АОП та їх взаємодія в процесі функціонування

Функціонування ГІМАК АОП формується із загального “внеска” її компонент – ФСІА (див. рис. 3): продукційних правил (ПП), експертних оцінок (ЕО), об’єктів інтелектуального вибору ОІВ – штучних нейросіток (ИНС), чисельних процедур (ЧП), об’єктів вибору з табличних даних (ОВТД), об’єктів побудування графіків (ОПГ), реалізації (АР) – якщо експерт або відповідний агент робить висновок щодо необхідності уведення нових правил, обмежень тощо. Базуючись на вхідних даних і меті ідентифікації в залежності від етапу послідовності синтезу моделі СДК, ГІМАК АОП реалізує на виході піхід, стратегію, політику часу, метод і як наслідок – модель СДК, що задовольняє умовам і обмеженням ГВС, точності апроксимації, складності реалізації, розмірності та швидкодії, рівню кваліфікації користувача.

Прийнявши за *H, G* відповідно набори (імена) змінних на вході і виході, а за Z — набір з усіх змінних (вхідних і на виході включно), приймаючих участь у міжагентному інформаційному обміні  а також враховуючи, що структура ГІМАК АОП формується з множини  взаємозв’язаних ФСІА, тоді, використавши множину компонент зі складових  (вхідних відносно цього агента  і дій  (на виході відносно формальний опис *і*-го агента  можна подати у вигляді:



де *Ti* — тип агента (ЕС, НС, ЧП тощо); *Di* — тип умови для активації (наприклад: *D*1 — зміною вхідних даних агента; *D*2 — естафетною умовою, тобто завершенням поточного прогону визначених попередніх агентів; *D*3 — незалежною активацією, в тому числі відповідно до часового закону; *D*4 — у відповідь на запит від іншого агента, і так далі); *Ui* , *Wi* — набори вхідних і вихідних змінних агента *Аi* відповідно; *Bi* , *Ei* — набори розташованих вище відносно *Аi* (чиї вказівки він виконує) і підлеглих (відносно *Аi*) агентів відповідно. При цьому:



а реалізація ⎜*Bi* ⎜>0 вказівок вищерозташованого агента є припустимою. Для відображення факту можливості активізації  рішенням інших ФСІА вважаємо, що значення спеціальної “перемикаючої” змінної  визначає стан (“активний”, “очікування”) агента *Ai*.

Агенти мультиагентного середовища ГІМАК АОП можуть характеризуватися: використовуваною моделлю – *N*1 (тобто НС, ПП, ЧП тощо); типом подання — *N*2 (наприклад, числовий, символьний або більш детально: фреймовий, семантичний и т. д. Ясно, що *N*1 и *N*2 тісно зв’язані); особливостями виконання — *N*3 (як окремі фізичні (мікросхеми, комп’ютери) або віртуальні об’єкти (в межах цієї ж програми), і так далі).

Тип *N*2 передбачає необхідність проміжних перетворень даних, інтеграцію множин чисельних значень і формування символьних умов для логічного виведення, а також навпаки — перетворення фактів (виведень) у чисельні значення (у випадку нечітких уявлень це відповідає перетворенням “чіткий → нечіткий” і “нечіткий → чіткий” відповідно). Ці перетворення можуть бути або “вбудованими” у загальне функціонування ФСІА, або виконуються спеціальними процедурами трансформацій. Сам ФСІА, у свою чергу, може також мати мультиагентну структуру. Так, для реалізації перетворень“нечіткий → чіткий”, наприклад, за наявності невизначеності при ідентифікації етапів синтезу ШНС такий ФСІА може виконуватися у вигляді фаззі-контролера []. При цьому кожний такий агент  формується з множини специфікацій *SP* якісних правил  наприклад, у формі: “ЯКЩО вхідній вимозі відповідає множина *Х* значень змінних, ТО відгуковій реакції вектора ідентифікації у фазовому просторі відповідає множина *Y*”. Для математичного опрацювання таких правил необхідні операції взаємодії між окремими нечіткими множинами, які належать до частини “ТО”. Тоді продукування рішень для фаззі-контролерів характеризується взаємодією задаваних експертом вхідних і вихідних функцій належності, причому, операції “І” и “АБО” використовуються як зв’язка в правилах у множинах, елементи яких належать до різних основних множин.

* + 1. Узагальнена модель гнучкої інтелектуалізованої мультиагентної системи вибору СДК

Наведена на рис. 2.11 узагальнена модель *гнучкої інтелектуалізованої мультиагентної системи* (ГІМАС) об’єднує *необхідну* для формування алгоритму синтезу СДК і *достатню* для задоволення вимог з боку обслуговуваної ГВС сукупність цільових компонент метаідентифікації: АОПВН, АОПП, АОПС, АОППЧ, АОПМ – видів неаизначеності, підходів, стратегії, метода планування, моделі СДК та інших (за переліком НВКО з розділу 2).



Рисунок 2.11. Узагальнена модель ГІМАС вибору СДК

***Визначення 3.8*.** *Гнучка інтелектуалізована мультиагентна система* – це сукупність ГІМАК АОП, в якій реалізується *модель поетапного синтезу* СДК з такою послідовністю їх перебирання в просторі НВКО, яка, будучи виконувана користувачем і/або внутрішнім ініціюючим джерелом, відтворює принципи агентно-орієнтованого підходу та автономно дозволяє виокремити модель/моделі СДК, здатні задовольнити критерії обслуговування властивостей ГВС.

Заключна процедура поетапного синтезу СДК в ГІМАС зводиться до реляційного перебирання ФСІА умов виконання критеріїв обслуговуваності поточним вектором можливостей конкретної СДК вимог з боку ГВС.

Саме перевірка при реалізації покрокового алгоритму синтезу СДК виконання умов задоволення певною компонентою АОП вимог/обмежень поставленої задачі найчастіше виявляється причиною появи лінгвістичних невизначеностей, опрацювання яких і потребує використання методів і засобів нейро-фаззі-технологій із залученням для розв’язання проблеми *процедур фаззіфікації та дефаззіфікації* [].

Ітераційні процедури дозволяють більш “тонко” відфільтровувати в процесі ідентифікації такі остаточні рішення, які у визначеній обмеженнями задачі мірі задовольняють критерій узгодженості за даною властивістю синтезованої СДК.

Необхідні компоненти для підтримки процесів ідентифікації на кожному з кроків алгоритму синтезу СДК надходять з відповідних баз знань (БЗ), а після узгодження з умовами задачі нові реалізації поповнюють ці БЗ, розширюючи таким чином коло ГВС, для яких є вже готові розв’язки.

* + 1. Агентно-орієнтований вибір умов "обмеження ГВС – модель СДК"

На рис. 2.12 приведена УМВ, що заснована на реляційних відношеннях між окремими компонентами НВКО, формування якої являє першу складність реалізації моделі.



Рисунок 2.12. Інтерпретаційна УМВ СДК

*Друга складність* полягає у кількісному визначенні вагомості реляційних зв'язків між визначальними класифікаційними ознаками і реалізується експертним рейтиновим оцінюванням альтернативних варіантів з використанням методів ранжування і попарних порівнянь [].Відповідні дослідження вимагають глибоких і ретельних прпрацювань щодо однозначності вирішення цієї частини задачі автоматизованого синтезу СДК. На рис. 2.12 наведені умовні кількісні результати попередніх обробок даних від джерел, що не претендуючи на строгість експертного оцінювання, можна розглядати в якості прикладу узагальненої методики при вирішенні задачі адекватного синтезу СДК.

***Екпертні технології оцінювання класифікаційних ознак.***

Кількісна оцінка вагомості реляційних зв'язків між визначальними класифікаційними ознаками здійснюється на основі експертного рейтингового оцінювання. Екперти оцінюють кожну ознаку окремим числом балів виходячи зі степеня ефективності функціонування СДК. Нижче наведені два методи визначення узгодженності експертів: метод ранжування та метод парних порівнянь [].

***Метод ранжування.*** Припустимо, що ми маємо експертів, котрі оцінюють n-критеріїв. Кожен експерт проводить ранжування критеріїв шляхом присвоєння номерів від 1 до  у відповідності зі зменшенням степеня важливості цих критеріїв. Результати зводимо в таблицю 2.4.



Таблиця 2.4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Ознаки | | | |
|  |  | 1 | 2 | … |  |
| Експерти | 1 |  |  | … |  |
| 2 |  |  | … |  |
| … | … | … | … | … |
|  |  |  | … |  |

Тут **– номер, котрий **–й експерт присвоїв **–му критерію. У подальшій роботі зручно перерахувати цю таблицю, так щоб присвоїти більш важливому критерію більше значення . Введемо нові вагові коефіцієнти. Тепер найбільше важливий критерій має оцінку *n*-1, а найменш важливий – 0. Вагові коефіцієнти в адитивному критерії визначаються наступним чином: , .

Степінь узгодженості результатів роботи експертів визначається коефіцієнтом:

,



де: , , .

Значення змінюються в межах 

Якщо: - усі експерти дають однакові оцінки;  - думки експертів неузгоджені. В залежності від величини коефіцієнта  результати оцінки критеріїв або приймаються або відхиляються.

***Метод парних порівнянь.*** Цей метод застосовується коли експерти не в змозі оцінити важливість ознаки у балах. У цьому випадку кожен -й експерт заповнює наступну таблицю 2.5.:

Таблица 2.5. Таблиця парних порівнянь експерта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | *j* |  |
| 1 |  |  | . . . |  |
| 2 |  |  | . . . |  |
| *i* | . . . | . . . | . . . | . . . |
|  |  |  | . . . |  |

Тут ,, якщо *i*-й критерій має перевагу над *j-м* критерієм, , в протилежному випадку*.*

Для кожного експерта визначаєм таку таблицю, тобто всього таблиць. Далі здійснюється об'єднання таблиць в одну шляхом сумування чисел у відповідних клітинках:.

Визначимо вагові коефіцієнти: , 

Цей метод також передбачає визначення степеня узгодженості експертів:, где  - число ν поєднань по *r*. Максимальне значення коефіцієнта узгодженості . Мінімальне значення  для  – парного і  - для  непарного.



Отже, при залученні 15 експертів було проведено опитування оцінок ефективності поєднання ознак СДК наведеними експертними методами. Також, після отримання результатів оцінювання було проведено аналіз на узгодженість експертів у даній області (коректність відповідей щодо відповідності властивостей із НВКО). Після проведення обчислень було отримано наступний результат: степінь узгодженості методом ранжування (); степінь узгодженості методом парних порівнянь (). У результаті бачимо, що степені узгодженості, отримані двома методами не менше ***0,5***, тому можемо записати значення в таблиці порівняння ефективності поєднання ознак СДК (Додаток А).

***Використання ШНС для синтезу СДК.***

Згідно із [137] АОП функції ФСІА можна реалізувати певними топологіями ШНС. Таким чином, кожний з етапів ЛМПС концептуальної моделі процесу вибору моделі з послідовностями реалізацій у вигляді траєкторій *tr*opt руху, визначених повним функціональним орграфом процесу синтезу СДК (у розділі 2), можна відтворювати топологіями ШНС у вигляді нейросіткової моделі реалізації такого графа (рис. 4).



Рисунок 4. – Нейросіткова модель повного функціонального орграфа синтезу СДК

За таких умов процедура синтезу бажаної моделі СДК буде полягати у настроюванні ваг кожної зі складових ФВН, ФП, ФС, ФПЧ, ФМ нейросіткової моделі на поточні значення експертних рейтингових оцінювань альтернативних варіантів з урахуванням вагомості реляційних зв’язків (див. рис. 3) згідно вимог ВОМЗ для відповідного етапу синтезу. В решті решт, виконуючи ітераційні процедури настроювання, можна отримати траєкторію *tr*opt руху [137], визначувану перетином складових моделей СДК з максимальними до ВОМЗ показниками відповідності. Ті самі міркування в процесі синтезу бажаної моделі СДК стосуються і процедур керування етапами узагальненої моделі ГІМАС (рис. 2).

Як приклад, на рис. 4 червоною лінією виділений слід умовно оптимальної траєкторії trопт ум, яка в результаті багатоітераційного перебирання визначає складові умовної бажаної моделі СДК (рис. 3):

|  |  |
| --- | --- |
|  | (2) |

На початкових етапах вибір можливого методу із альтернативних варіантів обумовлений об'єктивними і суб'єктивними факторами невизначеності. До об'єктивних факторів належать: неповнота множини вихідних даних, відсутність кваліфікованих спеціалістів, обмеження, що обумовлені зовнішнім середовищем. Суб'єктивні фактори визначаються досвідом і особистою зацікавленістю у варіантах вибору. Аналітичними методами визначити вплив таких причин на кінцевий вибір доволі складно. Разом з тим, використовуючи мультиагентний підхід, таку невизначеність можна значно зменшить, приймаючи раціональні рішення щодо вибору НВКО, що задовольнятиме вимоги ГВС.

* 1. Модуль автоматизованого синтезу параметрів у структурі СДК

Для впровадження процедури автоматизованого синтезу пропонується включити до СДК окремий модуль автоматизованого синтезу параметрів динамічного керування (Рис. 5).

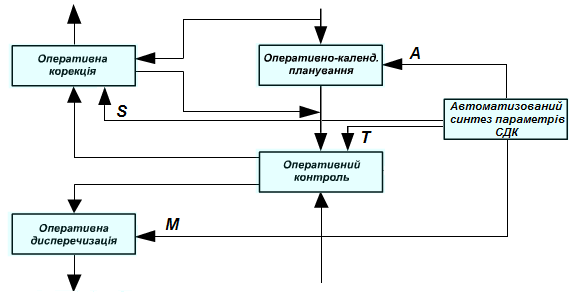


Рисунок 5. – Модуль автоматизованого синтезу параметрів у структурі СДК

Даний модуль з заданою періодичністю опрацьовуватиме отримані від виробничої системи дані про її роботу за попередній період та синтезуватиме адекватні до опрацьованих даних параметри роботи СДК. Відповідні параметри будуть передаватися іншим модулям СДК, що визначатиме їх роботу на наступний період.

До модуля оперативно-календарного планування передається підхід до динамічного планування (A), до модуля оперативної корекції – стратегія перепланування (S), до модуля оперативного контролю – політика вибору часу перепланування (T) і до модуля диспетчеризації – метод перепланування (M).

Висновки

Запропонована у попередньому розділі узагальнена концептуальна модель, а також реалізована за допомогою ГІМАС, що містить АОП для кожної властивості з НВКО, інтерпретаційна модель вибору адекватної СДК є основою для реалізації автоматизації цього процесу. Це дозволяє усунути суб’єктивний фактор некомпетентності користувача і реалізувати в автоматизованому виді логічну схему поетапного синтезу СДК.

1. Моделювання СДК з методом прямої диспетчеризації на основі МАС
   1. Синтез СДК для заданої ГВС

Розглянемо ГВС з однією з типових структурно-компонувальних схем.

Дана ГВС характеризується наступними видами невизначеностей:

Пов’язані з ресурсами:

* несправність виробничих модулів;
* затримки у доставці матеріалів.

Пов’язані з задачами:

* надходження термінових задач;
* зміна тривалості виконання операцій.

За допомогою мультиагентного підходу до поетапного синтезу було обрано СДК з наступними властивостями:

*Підхід*: прогнозтично-реактивне.

*Стратегія*: корекція плану.

*Політика вибору часу*: подієва.

*Метод календарного планування*:

*Метод прямої диспетчеризації*: метод на основі мультиагентних систем.

[Розширити пункт покроковим синтезом параметрів СДК]

* 1. Метод прямої диспетчеризації на основі МАС

Як уже зазначалося, зі зростанням складності виробничих систем задачі керування в реальному часі стають занадто складними для одного контролера. У такому випадку доцільно застосовувати децентралізовану систему, де кожна підсистема керується окремим контролером, що незалежно базується на локальній інформації і керує локальними діями.

Використання ***мультиагентного середовища*** (МАС) – приклад формування децентралізованої системи керування на основі застосування наборів впорядкованих інтелектуальних і автономних програмних агентів, що співпрацюють або змагаються для знаходження розв’язку задачі, що є занадто складною для розв’язання одним програмним агентом. Автономність тут означає, що агент є активною сутністю, яка може приймати власні рішення. Агент відрізняється від об’єкта в об’єктно-орієнтованому підході, який виконує заздалегідь визначені операції, на які хтось інший відправив запит, тим, що сам вирішує виконувати чи ні запитувану операцію, враховуючи власні цілі, пріоритети та умови. Зв’язок між агентами зазвичай відбувається завдяки переговорам – процесу комунікації, для подальшої координації та взаємодії [52]. Популярність децентралізованих систем, які базуються на МАС, зростає через те, що ці системи спроектовані для розподіленої і автономної роботи.

Не дивлячись на деяку кількість наявних досліджень щодо керування АТМ з використанням МАС, для розв’язання задачі прямої диспетчеризації даний метод використовується значно рідше ніж традиційні правила диспетчеризації. Однією з причин є недостатня ступіть чіткості формулювань у літературі узагальненого процесу переговорів і призначення задач. Тому задачею даного дослідження є розробка моделі та архітектури розподіленої системи реального часу – МАС прямої диспетчеризації (МАС ПД).

* + 1. Модель та структура мультиагентного середовища для реалізації прямої диспетчеризації

***Модель МАС ПД*** має наступний вигляд *,* де:

• *AG = {ag1, . . . , agn}* – скінченна множина агентів, кожен із яких представлений моделлю інтелектуального агента;

• *S* – скінченна множина станів зовнішнього середовища;

• *env : S×Aag1 ×. . .×Aagn* – функція, що описує можливу реакцію зовнішнього середовища на дії всіх агентів системи.

Запропонована ***структура МАС ПД*** АТМ і ГВМ зображена на рис. 4.1. Вона включає наступні агенти та метаагенти:

*agМ* – агент-менеджер;

*ag\*АТМ* – метаагент системи АТМ;

*ag\*ГВМ* – метаагент системи ГВМ;

*ag\*З* – метаагент системи замовлення.

***Метаагент (****ag\*i****)*** – це агент, що складаються з інших агентів.

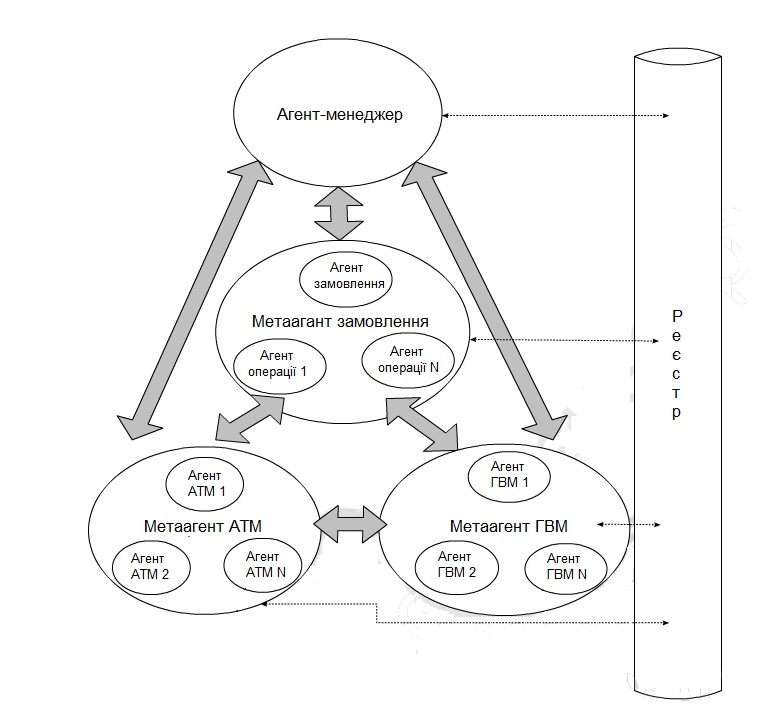


Рисунок 4.1. Архітектура МАС диспетчеризації

АТМ у виробничій системі відображений двома окремими типами агентів:

*agДАТМ* – агент диспетчеризації АТМ;

*agРАТМ* – агент ресурсів АТМ.

Коли в ГВС додається АТМi, то одночасно в МАС додається два агенти *agiДАТМ* і *agiРАТМ*. Ці два типи агентів формують метаагент АТМ (*ag\*АТМ*).

Схожим чином, ГВМ у ГВС також відображається двома окремими типами агентів, що утворюють метаагент ГВМ (*ag\*ГВМ*):

*agДГВМ* – агент диспетчеризації ГВМ;

*agДГВМ* – агент ресурсів ГВМ.

Метаагент замовлення *ag\*З* складається з множини агентів:

*agЗ* – агент замовлення, кожен з яких в свою чергу створює множину агентів:

*agО* – агент операції, що входить до складу замовлення.

Агенти *agО* створюються агентом *agЗ* використовуючи інформацію отриману від агента *agМ*.

Сховище даних Реєстр (*Reg)* – зберігає інформацію про *ag\*АТМ* і *ag\*ГВМ*. Використовуючи сховище, агенти знають про доступність інших агентів. Використовуючи канали інформаційного обміну у сховищі, агенти знають про доступність інших агентів.

Далі більш детально визначається механізм і поведінка всіх типів агентів.

* + 1. Функції агентів МАС ПД та міжагентна комунікація

**Агент-менеджер**. Агент *agМ* має дві основні функції: ініціалізація систем і створення задач. Для ініціалізації системи потрібні два агенти: *agМ* і *agЗ*, решта агентів ними створюються динамічно. Агент *agМ* створює агенти у *ag\*АТМ*, *ag\*ГВМ* та *agЗ*. Кількість транспортних і виробничих модулів передається до *agМ* через сховище даних *Reg*. Архітектура агента наведена у Додатку Б1.

У МАС ПД виділяють два типи сигналів: *внутрішні сигнали* та *зовнішні сигнали*. Внутрішні сигнали, які агент відсилає сам собі зазвичай є результатом виконання відповідних методів у планах агента, тоді як зовнішні сигнали – це повідомлення від інших агентів або середовища. У роботі розглядаються обидва типи сигналів. Агент *agМ* створює наступні агенти – *agЗ*, *agДАТМ*, *agРАТМ*, *agДГВМ* і *agРГВМ* – за допомогою відсилання сигналу “Ініціювати створення Агентів ГВМ, АТМ і замовлення” (*Initiate\_creation\_of\_machine\_AGV\_and\_staff\_agents*) до своїх локальних планів: “Створити замовлення” (*CreateStaff*), “Створити ГВМ” (*CreateMachine*), “Створити АТМ” (*CreateAGV*).

Агент *agМ* відсилає керуючі сигнали і чекає на відповідь від створених агентів. Сховище даних *Reg* надає доступ всім агентам до інформації про всіх зареєстрованих агентів. Це дозволяє іншим агентам системи знати про статус системи. Якщо *ag\*АТМ* чи *ag\*ГВМ* додаються чи видаляються з системи відповідні зміни вносяться до *Reg*. Це супроводжується відповідним планом АМ.

Інша важлива функція *agМ* – це створення задач у певний визначений час. Він визначає кількість операцій які має задача, їх послідовність і час виконання. Коли створюється нове замовлення – воно додається до списку *agМ* "Замовлення" (Orders). Коли у список додається новий запис генерується внутрішній сигнал “Сповістити про нове замовлення” (*Alert\_new\_order*), що містить інформацію про замовлення і оброблюється планом “Відіслати нове замовлення до Агенту замовлення” (*Send\_new\_order\_to\_staff\_agent*). Ця інформація про задачу пересилається агенту *agЗ* за допомогою зазначеного плану і генерації сигналу “Надіслати подію замовлення” (*Send\_order\_event*). Цей цикл повторюється щоразу, коли надходить нове замовлення.

**Агент замовлення.** Коли нове замовлення надходить до *agЗ* від *agМ*, він отримує сигнал “Надіслати подію замовлення” (*Send\_order\_event*)та ітеративно створює відповідні *agО* згідно з планом “Створення Агентів операцій” (*CreateOperationAgents*) (Додаток Б2). Агенти *agО* зберігають власну інформацію і автономно шукають найефективніший *agДАТМ*, що перемістить їх до бажаного ГВМ.

**Агент операції.** *agО*, що свторені *agЗ*, поділяються на два підтипи: активні і пасивні. Оскільки *agО* залежні від черги, то вони стають активними коли попередні операції завершені. Кожен *agО* має дві функції: Транспортування (*transportation*) і Обробка (*processing*) (Додаток Б3).

Агент *agО* має два приватних списки: один з них для зберігання операцій, а інший – для ставок, що надходять від *agДАТМ*. Після отримання інформації про операцію від *agЗ*, *agО* записує її до списку "Інформація про операції" (*OperationInfo*) згідно з планом “Запис до списку операцій” (*RecordIntoBeliefSet*). Після додавання інформації автоматично генерується сигнал “Зберегти інформацію про операцію” (*Post\_operation\_iformation*), що оброблюється планом “Відіслати запит до агентів АТМ і ГВС” (*Send\_a\_request\_to\_AGV\_machine\_agents*). Цей план спочатку перевіряє чергу операцій. Якщо це перша операція – план надсилає запит на транспортування всім *agДАТМ*, що містяться у *Reg*. У іншому разі план призупиняється до моменту, коли будуть здійснені всі попердні операції, за допомогою функції “Чекати чергу” (*waitfor*). Ця функція перевіряє чергу у списку "Інформація про операції" (*OperationInfo*). Коли попередні операції завершено, *agО* отримує сигнал “Активувати наступний Агент операції” (*Activate\_next\_operation\_agent*). Цей сигнал оброблюється планом “Оновити статус Агента операції” (*Update\_operation\_agent\_status*), що змінює статус операції у списку "Інформація про операції" (*OperationInfo*). Потім план розсилає запит на транспотрування всім *agДАТМ*і цей запит записується до їх робочих списків. На цьому етапі план знову призупиняється за допомогою функції “Чекати чергу” (*waitfor*), доки не отримає ставок на задачу транспортування від будь-якого *agДАТМ*. Як тільки перша ставка в переговорах про здійснення транспортування замовлення зроблена *agДАТМ*за допомогою сигнал “Зробити ставку” (Submit\_bid), процес переговорів починає керуватися планом *agО* “Оцінка ставок” (*Assesment\_of\_biddings*) заснованому на Contract Net Protocol (CNP) зображеному на рисунку 4.2.

Після отримання агентом *agДАТМ* першої ставки план стартує. В першу чергу іншим *agДАТМ* надсилається сигнал “Запит на ставку” (*Request\_a\_bid*). Позитивні відповіді, що приходять від *agДАТМ* через сигнал “Зробити ставку” (Submit*\_bid*) додаються до списку "Ставки"(*Bidding*). Найкраща пропозиція визначає *agДАТМ*-переможця через сигнал “Оголосити переможця” (*Declare\_winner*), а інші *agДАТМ* -учасники переговорів інформуються про результати і отримують запит на видалення задачі із робочих списків через сигнал “Видалити операцію” (*Delete\_event*). Це останній крок плану “Оцінка ставок” (*Assesment\_of\_biddings*). Після того, як транспортування здійснено *agДАТМ* -переможцем, ця інформація надсилається через сигнал “Транспортування завершено” (*Transportation\_completed*) і оброблюється планом АО “Оновити інформацію про транспортування” (*Update\_transportation\_information*).

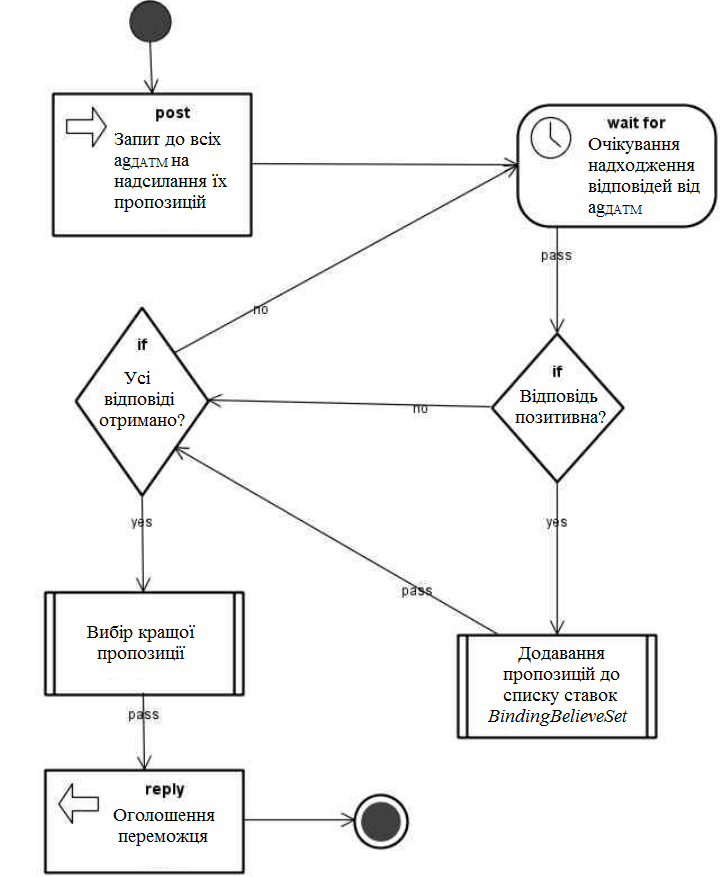


Рисунок 4.2. План “Оцінка ставок” (Assesment\_of\_biddings) агента *agО*

**Агент диспетчеризації АТМ.** У архітекрурі *agДАТМ* є деякі сигнали, що його активують (Додаток Б4).

Як тільки *agРАТМ* стає вільним він інформує *agДАТМ* сигналом “Інформація про простій” (*IdleInfo*). Агент *agДАТМ* відсилає сигнал “Транспортування завершено” (*Transportation\_completed*) до *agО* згідно з планом “Перевірити робочий список” (*Check\_the\_blackboard*). Він визначає наступну задачу. Рис. 4.3 підсумовує чей процес.

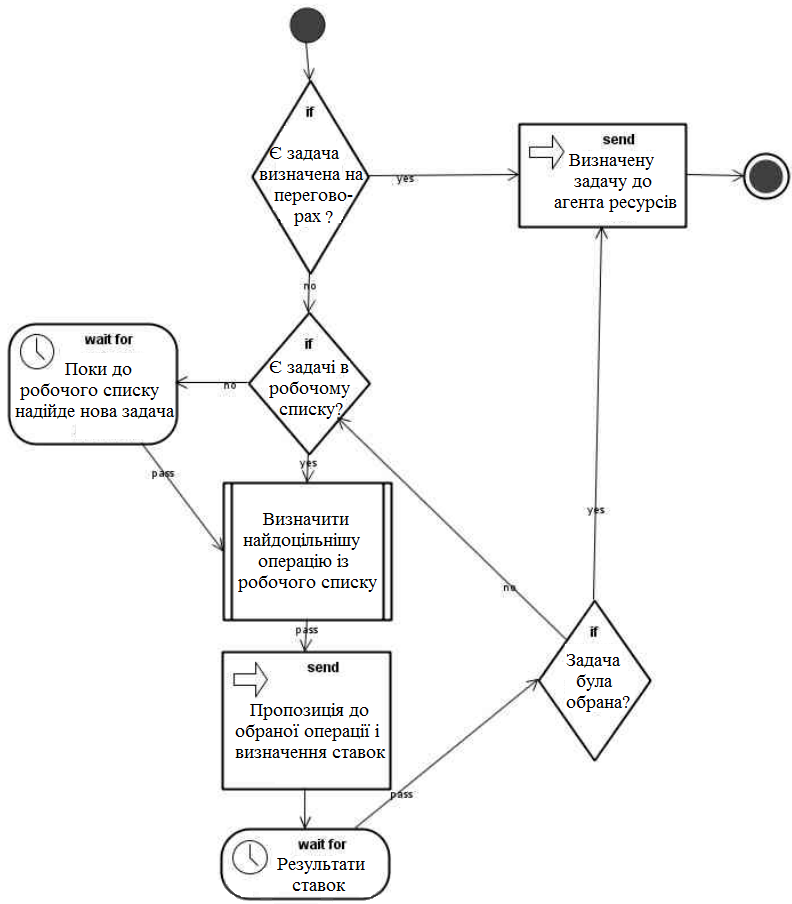


Рисунок 4.3. План “Перевірити робочий список” (*Check\_the\_blackboard*)

агента *agДАТМ*

Коли операція розпочалася на ГВМ, *agДГВМ* відсилає сигнал “Активувати наступний Агент операції” (*Activate\_next\_operation\_agent*) наступному *agО* . Після цього надходить запит на транспортування від наступного *agО* через сигнал “Відіслати запит до АТМ” (*Send\_request\_to\_AGV*). Агент *agО* також повідомляє про мінімальний можливий час початку обробки транспортної задачі. Ця інформація записується планом *agДАТМ* “Записати до робочого списку” (*Record\_in\_blackboard*) і використовується планом “Перевірити робочий список” (*Check\_the\_blackboard*) для визначення найбільш підходящої задачі.

Під час визначення задачі транспортування *agДАТМ* робить висновки, що базуються на оцінці часу переміщення і часу власного очікування найближчого можливого початку обробки відповідної операції з робочого списку. Мета цих висновків у виборі задачі з найближчим часом початку з робочого списку використовуючи наступне рівняння:

 (4.1)

де *ELTi* — найближчий час початку опрацювання задачі *i*; *CL* — поточне розташування АТМ; *PCPi* — розташування точки початку обробки задачі *i*; *t* — поточний момент часу; Δ*t*(..., ...) — час переміщення між двома точками; *EPTi* — найближчий час можливого початку обробки задачі *i*.

Далі обирається *ELTs* за допомогою рівняння:

 (4.2)

Час *PR* виконання відповідної операції, запропонований *agДАТМ* визначається як сума *EPTs*і відповідного часу доставки вантажу за допомогою рівняння:

 (4.3)

Після того, як почали робитися пропозиції, *agО* запрошує всі *agДАТМ* зробити свою, використовуючи план «Підготовка пропозиції» (*Prepare\_a\_proposal*). План спершу перевіряє наявність призначення задачі. Якщо не призначена, він готує пропозицію, шукаючи задачу з мінімальним *ELT,* використовуючи рівняння 3.2 і наступне:

 (4.4)

де *EFT* – найближчий час звільнення; *NL* – наступне розташування.

Якщо задача у поточній пропозиції співпадає з обраною задачею у робочому списку, АДАТМ робить пропозицію відповідній операції додаванням *ELTs*до відповідного часу перевезення (рівняння 4.3).

**Агент диспетчеризації ГВМ.** Під час роботи системи, *agО*, що був перевезений до пункту призначення надсилає до ГВМ сигнал «Відіслати запит до ГВМ» (*Send\_request\_to\_AGV*). Цей сигнал оброблюється планом «Записати у робочий список» (*Record\_in\_blackboard*) і запит записується до робочого списку *agДГВМ*. Після отримання сигналу «Інформація про простій» (*IdleInfo*) від *agРГВМ*, план «Перевірка робочого списку» (*Check\_the\_blackboard*) шукає у робочому списку нову задачу. У даному дослідженні гнучкість маршрутизації не враховується, тому *agДГВМ* не включається в переговори. Задача з найменшим часом обробки обирається як нова задача і надсилається до *agРГВМ*. Після вибору нової задачі *agДГВМ* інформує наступний *agО*. *agДГВМ* також інформує про найшвидший можливий час початку виконання надсиланням сигналу «Активувати наступний Агент операції» (*Activate\_next\_operation\_agent*). Цей цикл триває коли отримано новий запит на задачу від АО і сигнал «Інформація про простій» (*IdleInfo*) від *agРГВМ*. Архітектуру *agДГВМ* зображено у Додатку Б5.

**Агенти ресурсів АТМ і ГВМ.** *agРАТМ* і *agРГВМ* відображають фізичні ресурси транспортних і виробничих модулів. Ці модулі мають лише один план і один список. Після отримання нової задачі від *agДАТМ* чи *agДГВМ* через сигнал «Відіслати розклад» (*Send\_schedule*) вони починають обробку замовлення. Після завершення задачі, вони записують завершену задачу у свій список статистики для побудови діаграми Ганта і запитують у *agДАТМ* чи *agДГВМ* нову задачу відсилаючи сигнал «Інформація про простій» (*IdleInfo*). Цей процес виконується планом «Задача доставлена» (*Task\_transported*). Архітектура Агентів ресурсів зображена у Додатку Б5.

* 1. Застосування СНВ для підвищення ефективності роботи методу МАС динамічного керування

Щоб забезпечити зменшення часу переговорів агентів та для функціонування системи диспетчеризації в умовах невизначеності інтелектуальними агентами, що входять до складу МАС. Для вирішення даної задачі пропонується застосувати систему нечіткого виведення. Нечітка логіка раніше була застосована для розробки правил диспетчеризації на основі кількох критеріїв [33, 122]. Далі більш детально визначається механізм і поведінка АДАТМ.

Система нечіткого виведення

У запропонованому підході кожен АДАТМ використовує систему нечіткого виведення (СНВ) для підтримки прийняття рішень. Агент АДАТМ збирає інформацію про його оточення і надає цю інформацію до СНВ, яка оцінює наявні варіанти і допомагає агенту вирішити, яку саме задачу транспортування краще починати виконувати.

СНВ використовує три змінні як вхідні (Відстань, Час очікування і Частоту запитів), і одну в якості вихідної (Пріоритет).

Відстань (X1 ∈ {Далеко, Середня, Близько}) відповідає найкоротшому шляху до місця призначення.

Час очікування (X2 ∈ {Короткий, Середній, Довгий}) – це час, що минув з моменту прибуття АТМ до ГВМ.

Частота запитів (X3 ∈ {Низька, Середня, Висока}) – це відношення між кількістю задач, на які було здійснено запит, і загальною кількістю доступних задач, що відображає перевантаженість ГВМ.

Пріоритет (Y ∈ {Низький, Середньо низький, Середній, Середньо високий, Високий}) є значенням, яке АДАТМ присвоює кожному АО.

Нечіткі змінні і пов'язані з ними нечіткі множини зображені на рис. 4.4.

|  |
| --- |
| *а* |
| *б* |
| *в* |
| *г* |

Рисунок 4.4. – Множини вхідних (*а, б, в*) і вихідної (*г*) нечітких змінних

Нечіткі правила, використовувані у СНВ, наведено у таблиці 4.1. Перший рядок таблиці можна подати як правило "Якщо відстань "далеко" і час очікування "короткий" і частота запитів "висока", то пріоритет "низький"". Інші правила формуються аналогічно. Для виведення використовується метод Mamdani.

Таблиця 4.1. База нечітких правил СНВ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Відстань | Час очікування | Частота запитів | Пріоритет |
| 1 | Далеко | Короткий | Висока | Низький |
| 2 | Далеко | Короткий | Середня | Середньо низький |
| 3 | Далеко | Короткий | Низька | Середній |
| 4 | Далеко | Середній | Висока | Низький |
| 5 | Далеко | Середній | Середня | Низький |
| 6 | Далеко | Середній | Низька | Середній |
| 7 | Далеко | Довгий | Висока | Середньо низький |
| 8 | Далеко | Довгий | Середня | Середній |
| 9 | Далеко | Довгий | Низька | Середньо високий |
| 10 | Середня | Короткий | Висока | Середній |
| 11 | Середня | Короткий | Середня | Середній |
| 12 | Середня | Короткий | Низька | Середньо високий |
| 13 | Середня | Середній | Висока | Середньо низький |
| 14 | Середня | Середній | Середня | Середньо низький |
| 15 | Середня | Середній | Низька | Середній |
| 16 | Середня | Довгий | Висока | Середній |
| 17 | Середня | Довгий | Середня | Середньо високий |
| 18 | Середня | Довгий | Низька | Високий |
| 19 | Близько | Короткий | Висока | Середній |
| 20 | Близько | Короткий | Середня | Середньо високий |
| 21 | Близько | Короткий | Низька | Високий |
| 22 | Близько | Середній | Висока | Середньо низький |
| 23 | Близько | Середній | Середня | Середньо високий |
| 24 | Близько | Середній | Низька | Високий |
| 25 | Близько | Довгий | Висока | Середній |
| 26 | Близько | Довгий | Середня | Високий |
| 27 | Близько | Довгий | Низька | Високий |

Процес прийняття рішення відбувається наступним чином. Як тільки *agДАТМ* стає вільним, він перевіряє робочий список. *agДАТМ* призначає кожному *agО* зі списку пріоритет відповідно до нечітких правил. Він обчислює відстань і запитує час очікування і частоту запитів від *agО*. Далі *agДАТМ* передає цю інформацію до СНВ, яка визначає пріоритет. На виході СНВ маємо вже дефазифікований пріоритет, тобто його значення є дійсним числом, а не лінгвістичною змінною, яка використовується в нечітких правилах. Після того, як кожнен *agО* отримав пріоритет, *agДАТМ* вибирає *agО* з найвищим пріоритетом і запитує задачу від нього. Якщо два або більше *agО* мають однаковий пріоритет, буде обрано той, запит від якого надійшов раніше.

* 1. Порівняльний аналіз результатів моделювання роботи СДК на основі МАС та правил диспетчеризації

Щоб проаналізувати поведінку системи із запропонованим мультиагентним середовищем, порівняємо її продуктивність з кількома широко відомими правилами диспетчеризації, що є основним інструментарієм для диспетчеризації в реальному часі: MFCFS, STD, STT. Критерієм продуктивності було обрано період обробки. Для перевірки запропонованого підходу візьмемо тестову задачу з [13].

На рис. 4.5 зображено варіанти структур транспортної системи ГВС. М1 - М4 – гнучкі виробничі модулі. L/U – модуль завантаження-розвантаження готової продукції на склад.

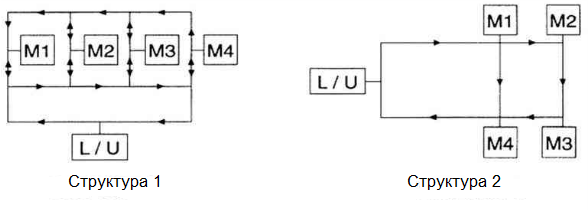


Рисунок 4.5. – Варіанти структур транспортної системи ГВС

Під час розв’язання тестових задач системою із запропонованим МАС, на початку моделювання було створено 4 агента ГВМ і 2 агента АТМ. Всі набори технологічних операцій для кожної задачі було внесено до агента-менеджера.

У таблиці 4.2 наведено матрицю часу переміщень між ГВМ для двох варіантів структур.

Таблиця 4.2. Матриця часу переміщеня між ГВМ

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Структура 1 | | | | | Структура 2 | | | | |
|  | L/U | М1 | М2 | М3 | М4 | L/U | М1 | М2 | М3 | М4 |
| L/U | 0 | 6 | 8 | 10 | 12 | 0 | 4 | 6 | 8 | 6 |
| М1 | 12 | 0 | 6 | 8 | 10 | 6 | 0 | 2 | 4 | 2 |
| М2 | 10 | 6 | 0 | 6 | 8 | 8 | 12 | 0 | 2 | 4 |
| М3 | 8 | 8 | 6 | 0 | 6 | 6 | 10 | 12 | 0 | 2 |
| М4 | 6 | 10 | 8 | 6 | 0 | 4 | 8 | 10 | 12 | 0 |

У таблиці 4.3 наведено набори технологічних операцій для тестових задач. У дужках подано час виконання кожної операції.

Таблиця 4.3. Набори технологічних операцій

|  |  |  |  |
| --- | --- | --- | --- |
| Приклад 1 | | Приклад 2 | |
| НТО 1 | М1(8); М2(16); М4(12) | НТО 1 | М1(10); М4(18) |
| НТО 2 | М1(20); М3(20); М2(18) | НТО 2 | М2(10); М4(18) |
| НТО 3 | М3(12); М4(8); М1(15) | НТО 3 | М1(10); М3(20) |
| НТО 4 | М4(24); М2(18) | НТО 4 | М2(10); М3(15); М4(12); |
| НТО 5 | М3(10); М1(15) | НТО 5 | М1(10); М2(15); М4(12); |
|  |  | НТО 6 | М1(10); М2(15); М3(12); |
| Приклад 3 | | Приклад 4 | |
| НТО 1 | М1(16); М3(15) | НТО 1 | М4(11); М1(10); М2(7) |
| НТО 2 | М2(18); М4(15) | НТО 2 | М3(12); М2(10); М4(8) |
| НТО 3 | М1(20); М2(10) | НТО 3 | М2(7); М3(10); М1(9); М3(8) |
| НТО 4 | М3(15); М4(10) | НТО 4 | М2(7); М4(8); М1(12); М2(6) |
| НТО 5 | М1(18); М2(10); М3(15); М4(17) | НТО 5 | М1(9); М2(7); М4(8); М2(10); М3(8) |
|  |  | НТО 6 | М2(10); М3(15); М4(8); М1(15) |

Набори технологічних операцій відправляються до агента замовлення, де вони розділяються на операції і для кожної створюється агенти операції, що мають дві функції: транспортування та обробки.

Результати, отримані системою з пропонованим МАС та обраними для порівняння правилами наведені в таблиці 4.4.

Таблиця 4.4. Результати виконання тестових завдань

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Приклад | MAS | MFCFS | STD | STT |
| 1-1 | 118 | 121 | 114 | 132 |
| 2-1 | **131** | 150 | 135 | 148 |
| 3-1 | 130 | 126 | 126 | 132 |
| 4-1 | **186** | 198 | 208 | 225 |
| 1-2 | **86** | 98 | 92 | 106 |
| 2-2 | **74** | 106 | 92 | 102 |
| 3-2 | **102** | 104 | 104 | 104 |
| 4-2 | **117** | 143 | 139 | 167 |

З таблиці 4.4 видно, що система з МАС випереджає інші правила диспетчеризації за показником тривалості періоду обробки. У 6 з 8 прикладів використання МАС показує найкращий результат. Однак роботи по удосконаленню запропонованого підходу продовжуються, що дає можливість сподіватися на подальше підвищення продуктивності системи з використанням МАС.

Також для однієї з типових структур було проведено моделювання роботи методу прямої диспетчеризації на основі СНВ. Результати порівнювалися з відповідними результатами мультиагентної системи на основі CNET та з використанням правил диспетчеризації, зокрема First Come First Served (FCFS). На рис. 4.6 зображено графік залежності середнього часу простою АТМ від часу моделювання для трьох наведених методів.

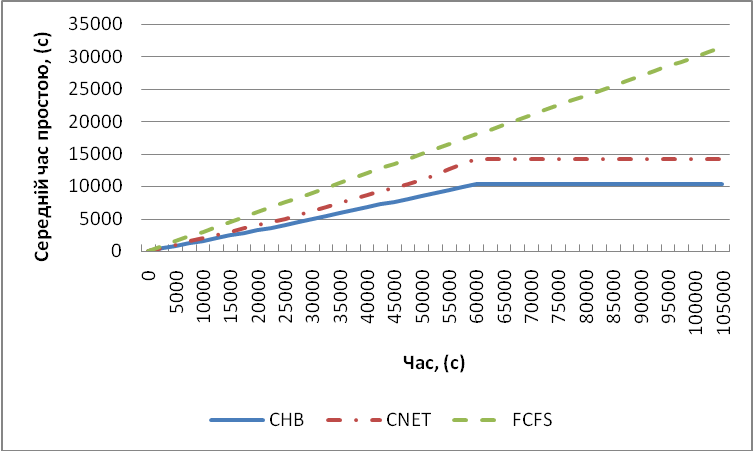


Рисунок 4.6. – Графік залежності середнього часу очікування АТМ

Метод на основі нечіткого виведення переважає за продуктивністю обидва інші методи. При використанні Методу FCFS може виникнути скупчення АТМ біля одного оброблювального ресурсу, що найдовше чекає на виконання транспортної операції. Це було враховано при розробці СНВ. При використанні CNET сам процес переговорів займає багато часу, оскільки АДГВМ повинен чекати відповіді від кожного агента АДАТМ, які можуть надійти не одразу. Якщо не дочекатися всіх пропозицій — можна не отримати найвигіднішу, якщо чекати — можна втратити багато часу.

Мультиагентну систему на основі СНВ було розроблено з урахуванням завантаженості, відстані і часу очікування. Час прийняття рішення значно зменшено в порівнянні з часом проведення переговорів. Замість надання пропозиції і очікування відповіді АДАТМ оцінює параметри виробничого середовища і обирає рішення (задачу для виконання), яке на його думку найбільш підходить для всієї ГВС. Недоліком такої системи диспетчеризації є те, що обравши задачу, АДАТМ більше не змінює свого рішення. Через це можливості, що можуть з'явитися у високо-динамічному середовищі можуть бути втрачені.

Висновки

У даному розділі запропоновано систему динамічної диспетчеризації АТМ і ГВМ у ГВС, яка базується на використанні МАС. Запропонована система детально описана, зокрема чітко викладено процес присвоєння задачі агенту, що має її виконувати. Система динамічної диспетчеризації АТМ і ГВМ у ГВС, яка базується на використанні МАС перевірена на кількох прикладах з вивчених джерел. Результати роботи було порівняно з класичними правилами диспетчеризації. У більшості тестових задач пропонована система показала вищу продуктивність. Отримані результати дозволяють зробити висновки про перспективність застосування МАС для диспетчеризації АТМ у реальному часі і показують, що продуктивність системи може бути покращена в майбутньому. Також перспективним напрямком розвитку системи є підвищення її стійкості за рахунок врахування агентами можливих невизначеностей, що мають місце у ГВС і можуть вплинути на результати роботи.

Висновки

Динамічне керування є однією з ключових складових управління виробничими об'єктами, основними з яких є обробні ресурси та транспортні модулі. Більшість виробничих систем працюють в динамічних середовищах, схильних до впливу різних подій в режимі реального часу, які можуть зробити прогностичне розклад неприпустимим. Таким чином, динамічне керування відіграє значну роль в успішній реалізації систем керування.

Визначено НРКП необхідний для формалізації процесу вибору методу динамічного керування і достатній для адекватного задоволення вимог обслужіваемості з боку розглянутої ГПС. В рамках НРКП проаналізовані види невизначеності в ГПС, чотири підходи до динамічного планування: реактивне, прогнозно-реактивне, Робастное, Робастное превентивне планування, основні політики часу і стратегії вирішення проблем оновлення розкладів, а так само основні методи динамічного керування.

Реалізація узагальненої ГІМАС, що містить АОП відповідно до кожного ознакою з НРКП, є основою для автоматизації процесу вибору, що дозволяє виключити суб'єктивні фактори некомпетентності користувача і реалізувати в автоматизованому вигляді логічну схему синтезу СДК.

[...]

Список використаних джерел

1. A multi-agent architecture for control of AGV systems / P. Farahvash, T.O. Boucher // Robotics and computer-Integrated Manufacturing, 20 (2004).
2. A time window approach to simultaneous scheduling of machines and material handling system in FMS / U. Bilge, G. Ulusoy // Operations Research, 43 (1995)
3. Abumaizar R. J. Rescheduling job shops under random disruptions / R. J. Abumaizar and J. A. Svestka. – International Journal of Production Research, 1997. – 35 (7). – p.p. 2065-2082.
4. Adaptive genetic fuzzy, predictive and multiobjective approach for AGVs dispatching / O. Morandin Jr, V. F. Carida, E. R. R. Kato, and C. C. M. Tuma. – Industrial Electronics Conference, 2011. – pp. 2317–2322.
5. Akturk, M. S. and Gorgulu, E., Match-up scheduling under a machine breakdown, *European Journal of Operational Research*, 112 (1), 81-97 (1999).
6. An approach for agent modeling in manufacturing on JADE / V.R. Komma, P.K. Jain, N.K. Mehta // International journal of advanced manufacturing, 2010.
7. Automated Negotiation and Decision Making in Multiagent Environments / S. Kraus. – Mutli-agents systems and applications, 2001
8. Aydin, M. E. and Öztemel, E., job-shop scheduling using reinforcement learning agents, *Robotics and Autonomous Systems*, 33 (2-3), 169-178 (2000).
9. Aytug, H., Lawley, M. A., McKay, K., Mohan, S., and Uzsoy, R., Execting production schedules in the face of uncertainties: A review and some future directions, *European Journal of Operational Research*, 161 (1), 86-110 (2005).
10. Bean J. C. Match up scheduling with multiple resources release dates and disruptions / J. C.Bean, J. R. Birge, J. Mittenthal, and C. E. Noon. – Journal of Operations Research, 1991. – 39 (3). – p.p. 471-483.
11. Belz, R. and Mertens, P., Combining knowledge-based systems and simulation to solve rescheduling problems, *Decision Support Systems*, 17 (2), 141-157 (1996).
12. Bierwirth C. Production scheduling and rescheduling with genetic algorithms / C. Bierwirth and D. C. Mattfeld. – Evolutionary Computation, 1999. – 7 (1). – p.p. 1-17.
13. Bilge U. A time window approach to simultaneous scheduling of machines and material handling system in FMS / U. Bilge, G. Ulusoy.– Operations Research, 43 (1995)
14. Bongaerts, L., Monostori, L., McFarlane, D., and Kadar, B., Hierarchy in distributed shop floor control, *Computers in Industry*, 43 (2), 123-137 (2000).
15. Brandimarte P. Modelling manufacturing systems: From aggregate planning to real-time control / P. Brandimarte and A. Villa. – Springer-Verlag, 1999.
16. Brennan R. W. Evaluating the performance of reactive control architectures for manufacturing production control / R. W. Brennan and D. H. Norrie. – Computers in Industry, 2001. – 46 (3). – p.p. 235-245 ().
17. Cavalieri, S., Garetti, M., Macchi, M., and Taisch, M., An experimental benchmarking of two multi-agent architectures for production scheduling and control, *Computers in Industry*, 43 (2), 139-152 (2000).
18. Chryssolouris G. Dynamic scheduling of manufacturing job shops using genetic algorithms / G. Chryssolouris and V. Subramaniam. – Journal of Intelligent Manufacturing, 2001. – 12 (3). – p.p. 281-293.
19. Church L. K. Analysis of periodic and event-driven rescheduling policies in dynamic shops / L. K. Church and R. Uzsoy. – International Journal of Computer Integrated Manufacturing, 1992. – 5 (3). – p.p. 153-163.
20. Cowling, P. I. and Johansson, M., Using real-time information for effective dynamic scheduling, *European Journal of Operational Research*, 139 (2), 230-244 (2002).
21. Cowling, P. I., Ouelhadj, D., and Petrovic, S., A Multi-agent architecture for dynamic scheduling of steel hot rolling, in *Proceedings of the Third International ICSC World Manufacturing Congress*, pp. 104-111, Rochester, NY, USA, 2001.
22. Cowling, P. I., Ouelhadj, D., and Petrovic, S., A Multi-agent architecture for dynamic scheduling of steel hot rolling, *Journal of Intelligent Manufacturing,* 14, 457-470 (2003a).
23. Cowling, P. I., Ouelhadj, D., and Petrovic, S., Dynamic scheduling of steel casting and milling using multi-agents, to appear in the *Journal of Production Planning and Control, Special Issue on the Application of Multi Agent Systems to Production Planning and Control,* 2003b.
24. Cowling, P. I., Ouelhadj, D., and Petrovic, S., Multi-agent systems for dynamic scheduling, in *Proceedings of the Nineteenth Workshop of Planning and Scheduling of the UK*, pp. 45-54, Ed. *Garagnani, Max, The Open* University, UK., pp. 45-54, 2000.
25. Daniels, R. L. and Kouvelis, P., Robust scheduling to hedge against processing time uncertainty in single-stage production, *Management Science*, 41 (2), 363-737 (1995).
26. Davenport A. J. Slack-based techniques for robust schedules / A. J. Davenport, C. Gefflot, and J. C. Beck. – Proceedings of the Sixth European Conference on Planning (ECP2001), 2001.
27. Development of an intelligent agent-based AGV controller for a flexible manufacturing system / S.C. Srivastava, A.K. Choudhary, S. Kumar, M.K. Tiwari // International journal of advanced manufacturing technology, 36 (2008).
28. Dorn J. Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning / J. Dorn, R. M. Kerr, and G. Thalhammer. – International Journal of Human Computer Studies, 1995. – 42. – p.p. 687- 704.
29. Dorn, J., Case-based reactive scheduling, in Kerr, R. M. and Szelke, E. (Eds.), *Artificial Intelligence in Reactive Scheduling*, pp. 32-50, Kluwer Academic Publishers, 1995b.
30. Dorn, J., Kerr, R. M., and Thalhammer, G., Reactive scheduling in a fuzzy temporal framework, in Szelke, E. and Kerr, R. M (Eds.), *Knowledge-based Reactive Scheduling*, pp. 39-55, North-Holland, 1994.
31. Duffie, N. A. and Piper, R. S., Non-Hierarchical Control of Manufacturing Systems, *Journal of Manufacturing Systems*, 5 (2), 137-139 (1986).
32. Dutta, A., Reacting to scheduling exceptions in FMS environments, *IIE Transactions*, 22 (4), 33-314 (1990).
33. Farahvash P. A multi-agent architecture for control of AGV systems / P. Farahvash, T.O. Boucher.– Robotics and computer-Integrated Manufacturing, 20 (2004).
34. Fox, M. S., ISIS: A retrospective. Intelligent Scheduling, in Zweben, Monte and Fox, M. S. (Eds.), *Intelligent Scheduling*, pp. 1-28, Morgan Kaufmann Publishers, INC, pp. 1-28, 1994.
35. Garetti, M. and Taisch, M., Using neuronal networks for reactive scheduling, in Kerr, R. M. and Szelke, E. (Eds.), *Artificial Intelligence in Reactive Scheduling*, pp. 146-147, Kluwer Academic Publishers, 1995.
36. Garner, B. J. and Ridley, G. J., Application of neuronal network process in reactive scheduling, in Szelke, E. and Kerr, R. M. (Eds.), *Knowledge-based Reactive Scheduling,* pp. 19-28, North-Holland, 1994.
37. Gasser, L. MACE: A flexible testbed for distributed AI research. / L. Gasser, C. Braganza, N. Hermann. // Distributed Artificial Intelligence. – 1987. – Pp. 119-152
38. Glover, F. and Laguna, M., *Tabu search*. Kluwer Academic Publishers, Boston, 1997.
39. Glover, F., Kelly, J. P., and Laguna, M., Genetic algorithms and tabu search: hybrids for optimisation, *Computers of Operation Research*, 22 (1) (1995), 111-134.
40. Goldsmith, S. Y. and Interrante, L. D., An autonomous manufacturing collective for job shop scheduling, in *The Proceedings of AI & Manufacturing Research Planning Workshop*, pp. 69-74, Albuquere, AAAI Press, 1998.
41. Gou, L., Luh, P. B., and Kyoya, Y., Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation, *Computers in Industry*, 37 (3), 213-231 (1998).
42. Henning, G. P. and Cerda, J., Knowledge-based predictive and reactive scheduling in industrial environments, *Computers and Chemical Engineering*, 24(9), 2315-2338 (2000).
43. Herroelen W. Project scheduling under uncertainty: Survey and research potentials / W. Herroelen and R. Leus. – European Journal of Operational Research, 2005. – 165(2). – p.p. 289-306.
44. Hewitt, C. A universal modular ACTOR formalism for AI // Proc. Third International Joint Conference on Artificial Intelligence (IJCAI-73). – 1973. – Pp. 235-245.
45. Holthaus, O., Scheduling in job shops with machine breakdowns: an experimental study, *Computers & Industrial Engineering*, 36 (1), 137-162 (1999).
46. Jahangirian M. Intelligent dynamic scheduling system: the application of genetic algorithms / M. Jahangirian and G. V. Conroy. – Integrated Manufacturing Systems, 2000. – 11 (4). – p.p. 247-257.
47. Jain, A. K. and Elmaraghy, H. A., Production scheduling/rescheduling in flexible manufacturing, *International Journal of Production Research*, 35 (1), 81-309 (1997).
48. Jensen, M. T., Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures, *Applied Soft Computing*, 1 (1), 35-52 (2001).
49. Jozefowska, J., Mika, M., Roycki, R., Waligora, G., and Wglarz, J. W., Local search meta-heuristics for discretecontinuous scheduling problems, *European Journal of Operational Research*, 107 (2) (1998), 354-370.
50. Kerr, R. M. and Szelke, E., *Artificial intelligence in reactive scheduling,* Kluwer Academic Publishers, 1995.
51. Kim M. Simulation based real-time scheduling in a flexible manufacturing system / M. Kim and Y. Kim. – Journal of Manufacturing Management Systems, 1994. – 13 (2). – p.p. 85-93.
52. Kraus S. Automated Negotiation and Decision Making in Multiagent Environments / S. Kraus. – Mutli-agents systems and applications, 2001.
53. Kutanoglu, E. and Sabuncuoglu, I., An analysis of heuristics in a dynamic job shop with weighted tardiness objectives, *International Journal of Production Research*, 37 (1), 165-187 (1999).
54. Kutanoglu, E. and Sabuncuoglu, I., Routing-based reactive scheduling policies for machine failures in dynamic job shops, *International Journal of Production Research*, 39 (14), 3141-3158 (2001).
55. Le Pape, C., Scheduling as intelligent control of decision-making and constraint propagation, in Zweben, M. and Fox, M. S. (Eds.), *Intelligent Scheduling*, Morgan Kaufmann Publishers, INC, pp. 67-98, 1994.
56. Lee, C. Y. and Uzsoy, R., Minimizing makespan on a single batch processing machine with dynamic job arrivals, *International Journal of Production Research,* 37 (1), 219-236 (1999).
57. Lenat, D. BEINGs: knowledge as interacting experts // Proc. International Joint Conference on Artificial Intelligence. – 1975. – Pp. 126-133.
58. Leon, V. J., Wu, S. D., and Storer, R. H., Robustness measures and robust scheduling for job shops, *IIE Transactions,* 26 (5), 32-41 (1994).
59. Leus, R. and Herroelen, W., The complexity of machine scheduling for stability with a single disrupted job, O*perations Research Letters*, 33 (2), 151-156 (2005).
60. Li, H., Li, Z., Li, L. X., and Hu, B., A production rescheduling expert simulation system, *European Journal of Operational Research*, 124 (2), 283-293 (2000).
61. Lin, G. Y. and Solberg, J. J., An agent based flexible routing manufacturing control simulation system, in *Proceedings of the 1994 Winter Simulation Conference*, pp. 970-977, 1994.
62. Lin, G. Y. and Solberg, J. J., Integrated shop floor control using autonomous agents, *IIE Transactions*, 24 (3), 57-71 (1992).
63. Ljunberg, M. The OASIS air trafic management system // Proc. Second Pacific Rim International Conference on AI.— 1992.
64. Maturana, F., Shen, W., and Norrie, D. H., MetaMorph: an adaptive agent-based architecture for intelligent manufacturing, *International Journal of Production Research*, 37 (10), 2159-2173 (1999).
65. Mehta S. V. Predictable scheduling of a single machine subject to breakdowns / S. V. Mehta and R. Uzsoy. – International Journal of Computer Integrated Manufacturing, 1999. – 12 (1). – p.p. 15-38.
66. Meziane, F., Vadera, S., Kobbacy, K., and Proudlove, N., Intelligent systems in manufacturing: current developments and future prospects, *Integrated Manufacturing Systems*, 11 (4), 218-238 (2000).
67. Miraftabi, R. Agents on the loose: An overview of agent technologies.— University of Joensuu.— 2000.
68. Miyashita K. CABINS: a framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair / K. Miyashita and K. Sycara. – Artificial Intelligence, 1995. – 76 (1). – p.p. 377-426.
69. Muhlemann A. P. Job shop scheduling heuristics and frequency of scheduling / A. P. Muhlemann, G. Lockett, and C. K. Farn. – International Journal of Production Research, 1982. – 20 (2). – p.p. 227-241.
70. Multi-agent negotiation strategies utilizing heuristic for the flow of AGVs / A. Wallace // International journal of production research, 45 (2007).
71. Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence / D. Naso and B. Turchiano. – 2005. – pp. 208–226.
72. Nof, S. Y. and Grant, F. H., Adaptive/predictive scheduling: review and a general framework, *Production Planning & Control*, 2 (4), 298-312 (1991).
73. O’Hare G. Foundations of Distributed Artificial Intelligence / G. O’Hare and N. Jennings. – Wiley, 1996.
74. O’kane, J. F., A knowledge-based system for reactive scheduling decision-making in FMS, *Journal of Intelligent Manufacturing*, 11 (5), 461-474 (2000).
75. O'Donovan, R., Uzsoy, R., and McKay, K. N., Predictable scheduling of a single machine with breakdowns and sensitive jobs, *International Journal of Production Research*, 37 (18), 4217-4233 (1999).
76. Ouelhadj, D., Cowling, P. I., and Petrovic, S. (2003b) Utility and stability measures for agent-based dynamic scheduling of steel continuous casting, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 175-180, Taipei, Taiwan, 2003b, Selected in the finalist best student award.
77. Ouelhadj, D., Cowling, P. I., and Petrovic, S., Contract net protocol for cooperative optimisation and dynamic scheduling of steel production, in Ajith, Ibraham, Katrin, Franke and Mario, Koppen (Eds.), *Intelligent Systems Design and Applications*, pp. 457-470, Springer-Verlag, 2003a.
78. Ouelhadj, D., Hanachi, C., and Bouzouia, B., Farhi, A. and Moualek, A., A Multi-contract net protocol for dynamic scheduling in flexible manufacturing systems, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1114-1120, USA, 1999.
79. Ouelhadj, D., Hanachi, C., and Bouzouia, B., Multi-agent architecture for distributed monitoring in flexible manufacturing systems (FMS). In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1120-1126, San Francisco, USA, 2000.
80. Ouelhadj, D., Hanachi, C., and Bouzouia, B., Multi-agent system for dynamic scheduling and control in manufacturing cells, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.1256-1262, Belgium, 1998.
81. Ovacik, I. M. and Uzsoy, R., Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent set-up times, *International Journal of Production Research*, 32 (6), 1243-1263 (1994).
82. Park, J., Kang, M., and Lee, K., Intelligent operations scheduling system in a job shop, *International Journal of Advanced Manufacturing Technology*, 11, 111-119 (1996).
83. Parunak H. V. Applications of distributed artificial Intelligence in industry, in O'Hare G. M. P. and Jennings N. R (Eds.), Foundation of Distributed Artificial Intelligence, Wiley Inter-science, 1996. – Chapter 4.
84. Parunak, H. V. D. Foundations of Distributed Artificial Intelligence / H. V. D. Parunak // Foundations of Distributed Artificial Intelligence.— Wiley Inter-Science, 1994.
85. Parunak, H. V., Agents in Overalls: experiences and issues in the development and deployment of industrial agentbased systems, *International Journal of Cooperative Information Systems*, 9(3), 209-227 (2000).
86. Parunak, H. V., Baker, A. D., and Clark, S. J., The AARIA agent architecture: an example of requirements-driven agent based system design, in *Proceedings of the 1st International Conference on Autonomous Agents*, pp.482-483, California, USA, 1997.
87. Parunak, H. V., Manufacturing experience with the contract net, in Huhns, M. (Eds.), *Distributed Artificial Intelligence*, pp. 285-310, Pitman, London, 1987.
88. Pendharkar, P. C., A computational study on design and performance issues of multi-agent intelligent systems for dynamic scheduling environments, *Expert Systems with Applications*, 16 (2), 121-133 (1999).
89. Petrovic, D., and Duenas, A., (2006), A Fuzzy Logic Based Production Scheduling/Rescheduling in the Presence of Uncertain Disruptions, to appear in *Fuzzy Sets and Systems*.
90. Pham D. T. Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks / D. T. Pham and D. Karaboga. – Springer, 2000.
91. Pinedo M. Scheduling theory, algorithms and systems / M. Pinedo. – First edition, Prentice Hall, 1995.
92. Rajendran C. A comparative study of dispatching rules in dynamic flow shops and job shops / C. Rajendran and O. Holthaus. – European Journal of Operational Research, 1999. – 116 (1), 156-170.
93. Ramasesh, R., Dynamic job shop scheduling: a survey of simulation research, *OMEGA International Journal of Management Science*, 18 (1), 43-57 (1990).
94. Ramos, C., An architecture and a negotiation protocol for the dynamic scheduling of manufacturing systems, in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 8-13, 1994.
95. Reeves, C. R., *Modern heuristic techniques for combinatorial problems,* John Wiley & Sons, McGraw-Hill International (UK) Limited, 1995.
96. Rossi, A. and Dini, G., Dynamic scheduling of FMS using a real-time genetic algorithm, *International Journal of Production Research*, 38 (1), 1-20 (2000).
97. Ruiz D. On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants / D. Ruiz, J. Canton, and N. J. Mara, A. Espuna and L. Puigjaner. – Computers and Chemical Engineering, 2001. – 25 (4). – p.p 829-837.
98. Sabuncuoglu I. Analysis of reactive scheduling problems in a job shop environment / I. Sabuncuoglu and M. Bayiz. – European Journal of Operational Research, 2000. – 126 (3). – p.p. 567-586.
99. Sabuncuoglu, I. and Karabuk, S., Rescheduling frequency in an FMS with uncertain processing times and unreliable machines, *Journal of Manufacturing Systems*, 18 (4), 268-283 (1999).
100. Sabuncuoglu, I., A study of scheduling rules of flexible manufacturing systems: a simulation approach, *International Journal of Operational Research*, 36(2), 527-546 (1998).
101. Sarin, S. C. and Salgame, R. R., Development of a knowledge-based system for dynamic scheduling, *International Journal of Production Research*, 28(8), 1499-1513 (1990).
102. Schmidt G. How to apply fuzzy logic to reactive scheduling, in Szelke E. and Kerr R. M. (Eds.), Knowledge-based Reactive Scheduling, North-Holland, 1994. – p.p. 57-67.
103. Shafaei, R. and Brunn, P., Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach, *International Journal of Production Research*, 37 (17), 3913-3925 (1999a).
104. Shaw, J. M., Dynamic scheduling in cellular manufacturing systems: a framework for Network decision making, *Journal of Manufacturing Systems*, 7 (2), 83-94 (1988).
105. Shen, W. and Norrie, D. H., Agent based systems for intelligent manufacturing: a state of the art survey, *International Journal of Knowledge and Information Systems*, 1 (2), 129-156 (1999).
106. Shen, W., Norrie, D. H., and Barthes, J. P. A., *Multi-agent systems for concurrent intelligent design and manufacturing*. Taylor & Francis, London, 2001.
107. Shukla, C. S. and Chen, F. F., The state of the art in intelligent real-time FMS control: a comprehensive survey, *Journal of Intelligent Manufacturing*, 7, 441-455 (1996).
108. Smith F. S. OPIS: A methodology and architecture for reactive scheduling, in Zweben, M. and Fox, M. S. / F. S. Smith. – Intelligent Scheduling, Morgan Kaufmann Publishers, INC, 1994.
109. Smith, F. S., Reactive scheduling systems, in Brown, D. and Scherer, W. T. (Eds.), *Intelligent Scheduling Systems*, pp. 155-192, Kluwer Academic Publisher, 1995.
110. Smith, R., The contract net protocol: high level communication and control in distributed problem solver, *IEEE Transactions on Computers*, 29 (12), 1104-1113 (1980).
111. Sousa, P. and Ramos, C., A distributed architecture and negotiation protocol for scheduling in manufacturing systems, *Computers in Industry*, 38 (2), 103-113 (1999).
112. Stoop P. P. M. The complexity of scheduling in practice, International Journal of Operations and Production management / P. P. M. Stoop and V. C. S. Weirs, 1996. – 16 (10). – p.p. 37-53.
113. Sun, J. and Xue, D., A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources, *Computers in Industry*, 46 (2), 189-207 (2001).
114. Suresh V. Dynamic scheduling a survey of research / V. Suresh and D. Chaudhuri. – International Journal of Production Economics, 1993. – 32 (1). – p.p. 53-63.
115. Szelke, E. and Kerr R. M., *Knowledge-based reactive scheduling,* North-Holland, 1994.
116. Tharumarajah, A. and Bemelman, R., Approaches and issues in scheduling a distributed shop-floor environment, *Computers in Industry*, 34 (1), 95-109 (1997).
117. The Hearsay-I speech understanding system: An example of the recognition process / D. Reddy, L. Erman, R. Fennell, R. Neely // Transactions on Computers. – 1976. – April. – no. 4. – Pp. 422–431.
118. Using ARCHON to develop real-world DAI applications for electricity transportation management and particle acceleration control / N. R. Jennings, J. M. Corera, I. Laresgoiti et al. // EEE Expert Special Issue on Real World Applications of DAI systems.— 1996.
119. Vieira G. E. Rescheduling manufacturing systems: a framework of strategies, policies and methods / G. E. Vieira, J. W. Hermann, and E. Lin. – Journal of Scheduling, 2003. – 6 (1). – p.p. 36-92.
120. Vieira, G. E., Hermann, J. W., and Lin, E., Predicting the performance of rescheduling strategies for parallel machine systems, *Journal of Manufacturing Systems*, 19 (4), 256-266 (2000b).
121. Vieira, G. E., Herrmann, J. W., and Lin, E., Analytical models to predict the performance of a single machine system under periodic and event-driven rescheduling strategies, *International Journal of Production Research*, 38 (8), 1899- 1915 (2000a).
122. Wallace A. Multi-agent negotiation strategies utilizing heuristic for the flow of AGVs / A. Wallace.– International journal of production research, 45 (2007).
123. Wooldridge, M. J. Intelligent agents: Theory and practise / M. J. Wooldridge, N. R. Jennings // The Knowledge Engineering Review.— 1995.
124. Wu S. D. A rescheduling procedure for manufacturing systems under random disruptions / S. D. Wu, R. H. Storer and P. C. Chang. – Proceedings Joint USA/German Conference on New Directions for Operations Research in Manufacturing, 1991. – p.p. 292-306.
125. Wu, S. D., Storer, R. H., and Chang, P. C., One machine rescheduling heuristics with efficiency and stability as criteria, *Computers Operations Research*, 20 (1), 1-14 (1993).
126. Yamamoto M. Scheduling/rescheduling in the manufacturing operating system environment / M. Yamamoto and S. Y. Nof. – International Journal of Production Research, 1985. – 23 (4). – p.p. 705-722.
127. Youssef, H., Sait, S.M., and Adiche, H., Evolutionary algorithms, simulated annealing and tabu search: a comparative study, *Engineering Applications of Artificial Intelligence*, 14 (2) (2001), 167-181.
128. Zhou, H., Feng, Y., and Han, L., The hybrid heuristic genetic algorithm for job shop scheduling, *Computers and Industrial Engineering*, 40 (3) (2001), 191-200.
129. Zweben M. Intelligent scheduling / M. Zweben and M. S. Fox. – Morgan Kaufmann Publishers, INC, 1994.
130. Гнучкі комп’ютеризовані системи: проектування, моделювання і управління / Л.С. Ямпольський, П.П. Мельничук, Б.Б. Самоткін, М.М. Поліщук, М.М. Ткач, К.Б. Остапченко, О.І. Лісовиченко. – Житомир: ЖДТУ, 2005.–680 с.
131. Гнучкі комп’ютеризовані системи: проектування, моделювання і управління / Л.С. Ямпольський, П.П. Мельничук, Б.Б. Самоткін, М.М. Поліщук, М.М. Ткач, К.Б. Остапченко, О.І. Лісовиченко. – Житомир: ЖДТУ, 2005.–680 с.
132. Дьяков С.О. Динамічне планування у виробничих системах в умовах невизначеності / Дьяков С.О., Ямпольський Л.С. – Технологічні комплекси, 2014. – 2 (10). – С. 22-26
133. Дьяков С.О. Узагальнена концептуальна модель системи динамічного керування у гнучких виробничих системах / Дьяков С.О., Ямпольський Л.С. – Вісник ЖДТУ. Серія:Технічні науки, 2015. – 1 (72).
134. Мультиагентне середовище моделювання задач диспетчеризації автономних транспортних модулів / Л.С. Ямпольський, С.О. Дьяков. – АСАУ, 2013 № 2(23). – с. 9-17.
135. Проектирование металорежущих станков и станочных систем : справочник-учебник в 3-х т. Т. 3: Проектирование станочных систем / под общ ред. А. С. Проникова. – М. : МГТУ им. Н. Э. Баумана; Станкин, 2000. – 584 с.
136. Ямпольский Л.С. Системи штучного інтелекту в плануванні, моделюванні та управлінні: підруч. для студ. вищ. навч. закл. / Л.С. Ямпольський; Б.П. Ткач; О.І. Лісовиченко. – К.: ДП «Вид Дім «Персонал», 2011, – 544 с.
137. Ямпольський Л.С. Агентно-орінтована ідентифікація нейронних сіток / Л.С. Ямпольський. – Адаптивні системи автоматичного управління, 2015. – 2 (27).
138. Ямпольський Л.С. Гнучкі комп'ютеризовані системи: проектування, моделювання і управління: підручник / Л.С. Ямпольський, П.П. Мельничук, Б.Б. Самотокін, М.М. Поліщук, М.М. Ткач, К.Б. Остапченко, О.І. Лісовиченко. – Житомир: ЖДТУ, 2005.
139. Ямпольський Л.С. Нейротехнології та нейросистеми / Л.С. Ямпольський. – К.: Монографія. – "Дорадо-Друк", 2015. – 508 с.
140. Ямпольський Л.С. Нечітка ітераційна метаідентифікація штучних нейросіток в мультиагентному середовищі / Л.С. Ямпольський. – Вісник кіровоградського національного те-хнічного університету – Кіровоград: КНТУ. – №26 – 2013. – С. 207 – 218.

Додаток А

Таблиця А1. Оцінка поєднання ознак СДК: підходи, невизначеності

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Підходи** | Реактивний | Прогнозтично-реактивний | Робастний прогнозтично-реактивний | Робастний превентивний |
| **Невизначеності** |
| Несправність виробничого модуля | 0,6 | 0,7 | 0,8 | 0,4 |
| Несправність інструменту | 0,7 | 0,8 | 0,3 | 0,6 |
| Затримка або дефектність матеріалу | 0,6 | 0,7 | 0,4 | 0,8 |
| ... | ... | ... | ... | ... |
| Зміна пріоритету задач | 0,7 | 0,6 | 0,8 | 0,3 |
| Зміна терміну виконання операції | 0,5 | 0,8 | 0,7 | 0,6 |

Таблиця А2. Оцінка поєднання ознак СДК: архітектури, підходи

|  |  |  |
| --- | --- | --- |
| **Архітектури** | Централізована | Розподілена |
| **Підходи** |
| Реактивний | 0,7 | 0,4 |
| Прогнозтично-реактивний | 0,5 | 0,8 |
| Робастний прогнозтично-реактивний | 0,3 | 0,8 |
| Робастний превентивний | 0,4 | 0,7 |

Таблиця А1. Оцінка поєднання ознак СДК: методи, підходи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методи** | Правила диспетчеризації | Евристики | ... | Мультиагентні системи |
| **Підходи** |
| Реактивний | 0,9 | 0,3 | ... | 0,4 |
| Прогнозтично-реактивний | 0,7 | 0,5 | ... | 0,9 |
| Робастний прогнозтично-реактивний | 0,5 | 0,8 | ... | 0,8 |
| Робастний превентивний | 0,4 | 0,6 | ... | 0,6 |

Таблиця А1. Оцінка поєднання ознак СДК: стратегії, підходи

|  |  |  |
| --- | --- | --- |
| **Стратегії** | Повне перепланування | Корекція плану |
| **Підходи** |
| Реактивний | 0,2 | 0,9 |
| Прогнозтично-реактивний | 0,4 | 0,6 |
| Робастний прогнозтично-реактивний | 0,2 | 0,8 |
| Робастний превентивний | 0,3 | 0,7 |

Таблиця А1. Оцінка поєднання ознак СДК: політики часу, підходи

|  |  |  |  |
| --- | --- | --- | --- |
| **Політики часу** | Періодична | Подієва | Гібридна |
| **Підходи** |
| Реактивний | 0,3 | 0,8 | 0,5 |
| Прогнозтично-реактивний | 0,5 | 0,7 | 0,9 |
| Робастний прогнозтично-реактивний | 0,4 | 0,6 | 0,8 |
| Робастний превентивний | 0,8 | 0,5 | 0,7 |

Додаток Б

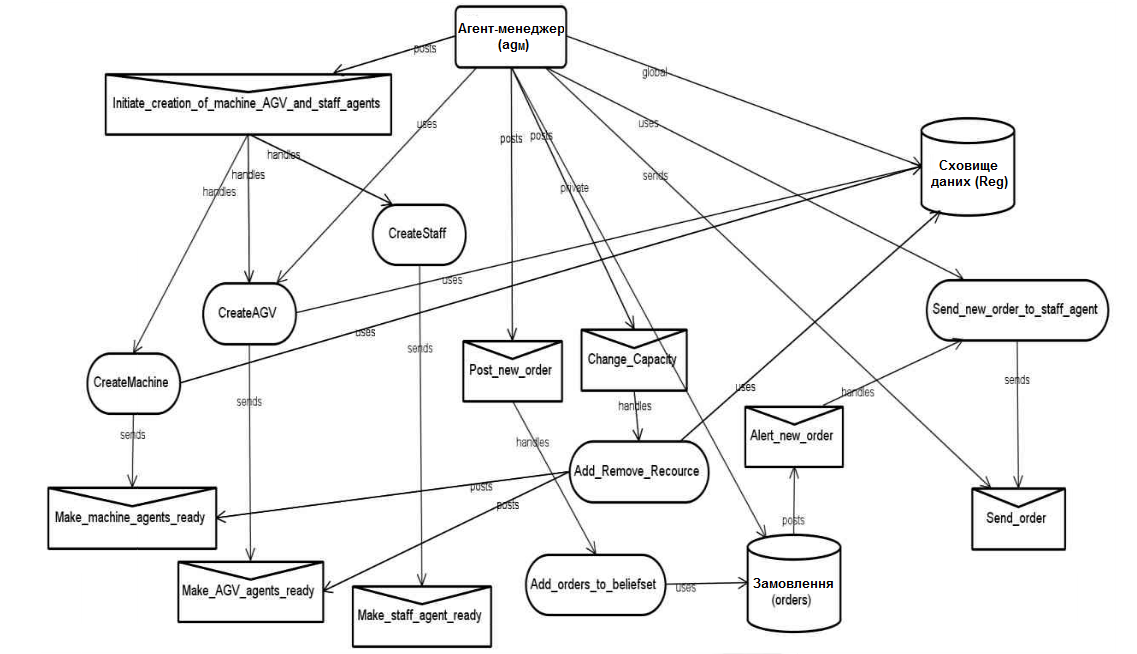


Рисунок Б1. Архітектура агента-менеджера *agМ*

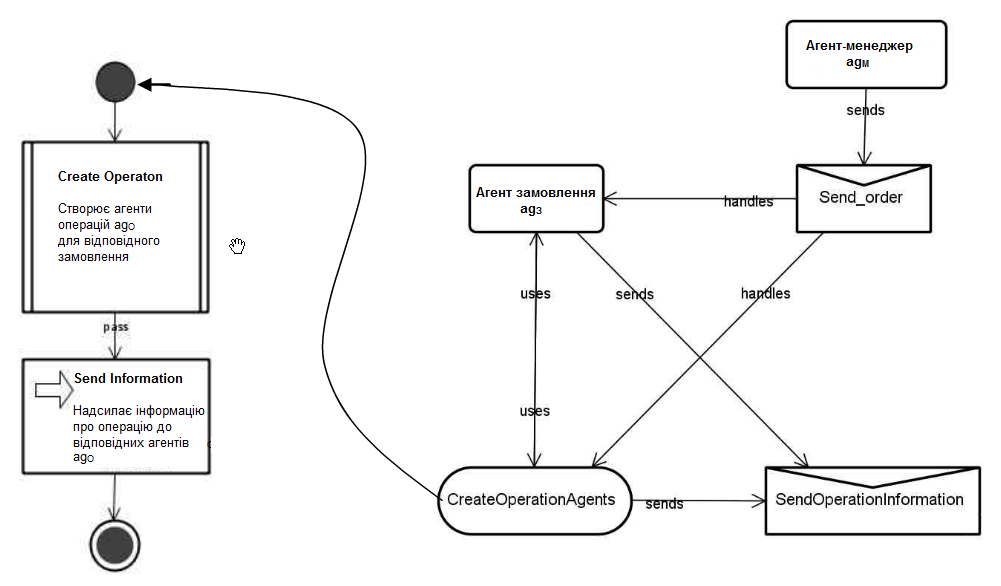


Рисунок Б2. Архітектура агента замовлення *agЗ*

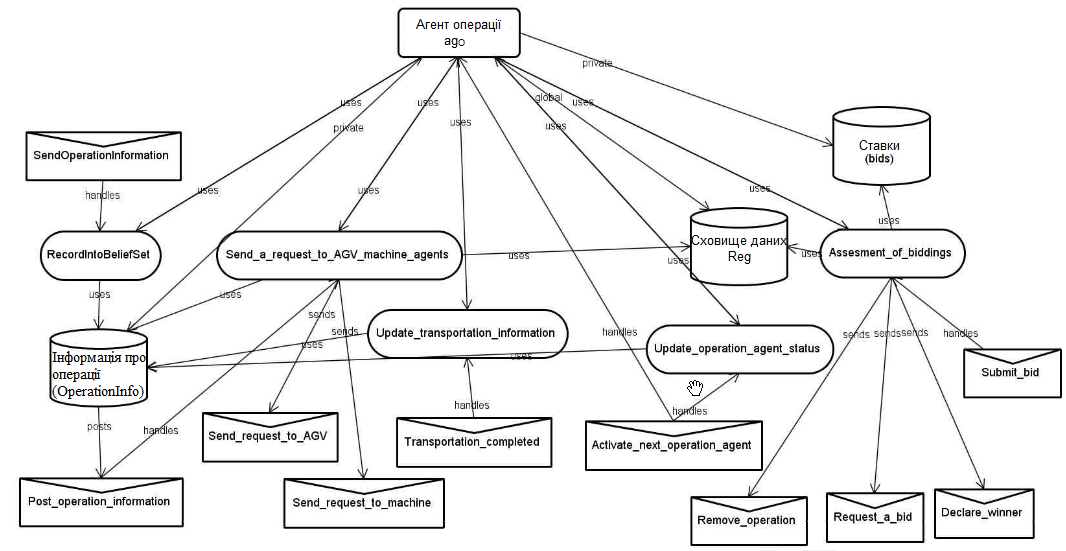


Рисунок Б3. Архітектура агента операції *agО*

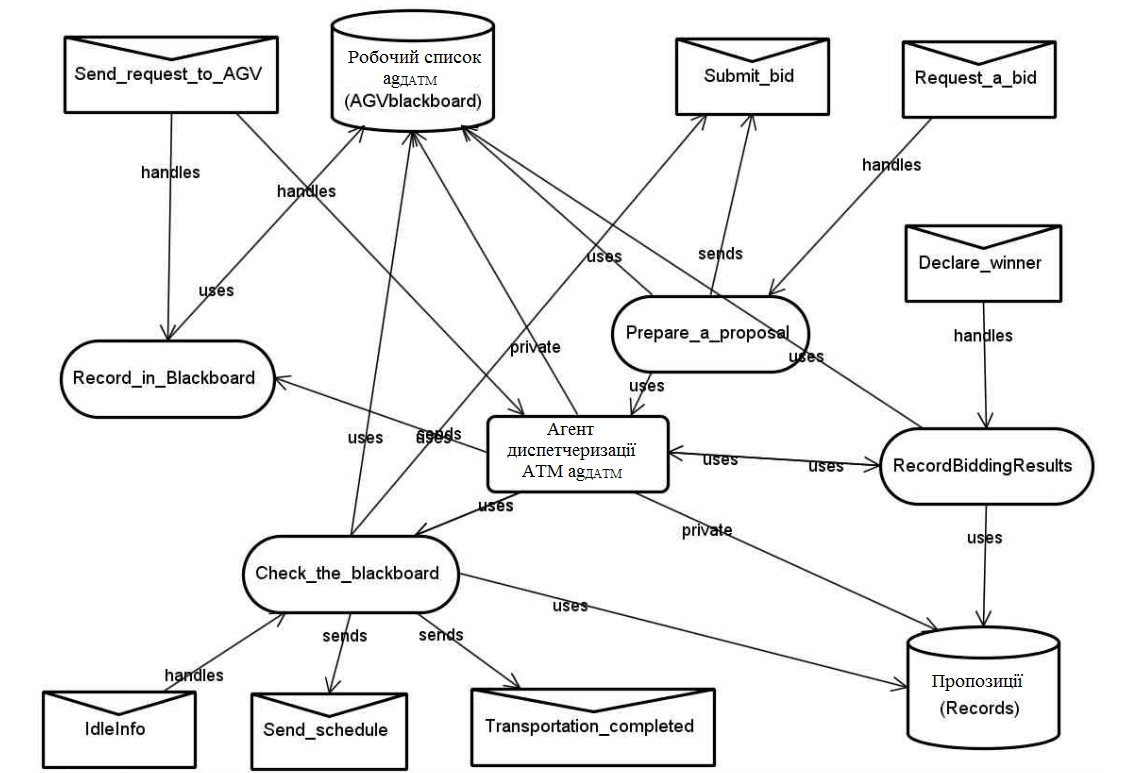


Рисунок Б4. Архітектура агента диспетчеризації АТМ *agДАТМ*

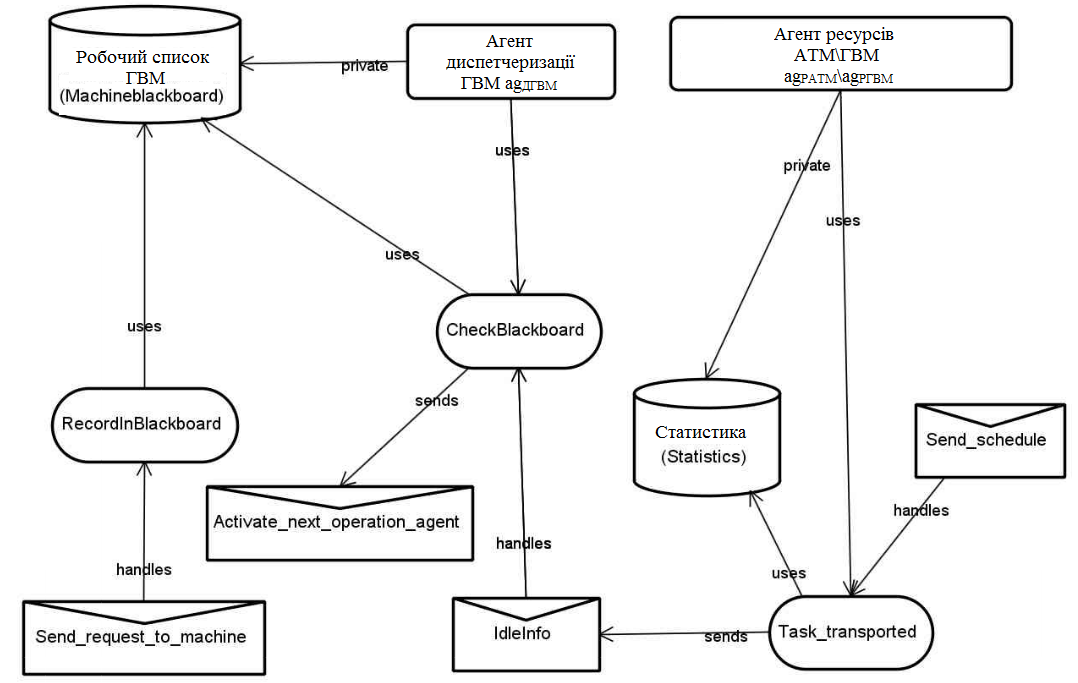


Рисунок Б5. Архітектура агента диспетчеризації ГВМ *agДГВМ* та агентів ресурсів