# Огляд динамічного планування у виробничих систем

**1. ВСТУП**

Планування визначається як процес розподілу ресурсів по операціях у часі. Це процес прийняття рішень з метою оптимізації однієї або кількох цілей (Pinedo, 1995). Цілі можуть бути наступні: мінімізація часу виконання задач, середнього часу потоку, затримки операцій, вартості обробки і т.д.

Планування відіграє важливу роль в багатьох виробничих системах. Проблеми планування, які пов’язані з пошуком оптимальних (або майже оптимальних) прогностичних графіків, підлягають ряду обмежень, в основному NP-складності. Досі дослідження в основному були зосереджені на пошуку оптимального (або майже оптимального) рішення для статичних моделей відносно різних величин, наприклад найкоротший загальний час обробки, мінімальна вартість виробництва і т.д. Ці підходи в основному використовували неявне припущення щодо статичності середовища без будь-яких збоїв. Широкий огляди літератури з статичного детермінованого планування можна знайти в (Weirs, 1997; Jain and Meeran, 1999; Pinedo, 1995, 2002).

Інтелектуальне планування з багатьох причин є невід'ємною частиною системи планування виробництва (Shafaei and Brunn, 1999b; Vieira et al., 2003). Прогностичний графік виступає у якості загального плану, на якому базуються багато видів діяльності виробничої системи. Приклади такої діяьності включають короткострокове планування роботи, впорядкування та підготовку деталей, планування оснастки, наладку діяльності і т.д. Це може дозволити поліпшити координацію за допомогою правильної синхронізації видів діяльності виробничої системи для підвищення продуктивності та мінімізації експлуатаційних витрат. Прогностичний розклад може визначити конфлікти ресурсів, контролювати закінчення виконання операцій, і забезпечити вчасне надходження необхідних деталей. Це дає виробничому персоналу чітке розуміння того, що потрібно робити і таким чином менеджери можуть визначити їх продуктивність. На жаль, більшість виробничих систем працюють в динамічних середовищах, де, як правило, неминучі непередбачувані події в режимі реального часу можуть призвести до зміни запланованих графіків і оптимальні або близькі до оптимальних розклади на базі оціночних даних можуть бути вже застарілими на той час, коли вони будуть застосовані у виробництві. Прикладами таких подій у реальному часі можуть бути: відмова машин, необхідність виконання невідкладних задач, зміна терміну виконання і т.д. MacCarthy and Liu (1993) відзначають природу різниці між теорією та практикою планування, провал класичної теорії розкладів щодо необхідності реагувати на потреби практичних середовищ і останні тенденції в дослідженнях планування, пов’язані з намаганням зробити його більш актуальним і застосовним. Shukla and Chen (1996) у комплексному огляді інтелектуального управління в режимі реального часу в гнучких виробничих системах вказали, що порівняння теорії та практики планування показує дуже малу відповідність між ними. Cowling і Johanson (2002) відзначають важливу різницю між теорією і практикою планування та заявляють, що моделі та алгоритми планування не можуть користуватися інформацією в режимі реального часу.

До недавнього часу проблемою планування за наявності подій в реальному часі, що визначається як *динамічне планування,* значною мірою нехтували і в цій області було проведено не багато досліджень. У цій статті ми зосередилися на ряді питань, які з’явилися в останні роки у динамічного плануванні у виробничих систем. Ми, перш за все, розглядаємо питання обробки виникнення подій у реальному часі під час виконання заданого графіка у виробничій системі.

Робота організована таким чином. Розділ 2 визначає категорії подій в реальному часі. Далі, Розділ 3 визначає динамічні підходи планування. Розділ 4 описує політики і стратегії перепланування. Розділ 5 наводить огляд попередніх досліджень щодо методів, що використовуються для вирішення проблеми динамічного планування, наприклад: правила диспетчеризації, евристики, мета-евристики, системи засновані на знаннях, нечітка логіка, нейронні мережі, гібридні методи і мульти агентні системи. Розділ 6 представляє порівняльне дослідження різних методів. Нарешті, реферат та висновки завершують статтю.

**2. ПОДІЇ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ**

Виробничі середовища динамічні за своєю природою і схильні до виникнення різних невизначеностей, що являють собою до події в реальному часі, які можуть змінити стан системи і впливають на її продуктивність. Література з динамічного планування 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 (Stoop and Weirs, 1996; Suresh and Chaudri, 1993; Cowling and Johanson, 2002; Vieira et al., 2003):

• **Пов'язані з** **ресурсами:** несправність машини, помилка оператора, відсутність або несправність інструмента, ліміти завантаження, затримки у доставці матеріалів, дефектність матеріалу (матеріал з неправильною специфікацією) і т.д.

• **Пов'язані з** **операціями:** термінові операції, відміна операцій, зміни терміну виконання, невчасне надходження операцій, зміна пріоритету операцій, зміна тривалості виконання операцій і т.д.

**3. ПІДХОДИ ДО ДИНАМІЧНОГО ПЛАНУВАННЯ**

Динамічне планування можна поділити на чотири категорії (Mehta and Uzsoy, 1999; Vieira et al., 2000a, 2003; Aytug et al., 2005; Leus and Herroelen, 2005): реактивне планування, прогностично-реактивне планування, робастне прогностично-реактивне планування і робастне превентивне планування.

***3.1.*** ***Реактивна 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 в підрозділі 5.1.

***3.2.*** ***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.

***3.3.*** ***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 (Wu et al., 1991, 1993; Leon et al., 1994).

Навіть незважаючи на те, що необхідності створення робастного графіка була визнана більше двох десятиліть тому Graves 1981 (Daniels and Kouvelis, 1995), 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 (Wu et al., 1991, 1993; Cowling and Johansson, 2002; Leus and Herroelen, 2005). Wu et al. (1991, 1993) визначив 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. Abumaizar and Svestka (1997) 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. Jensen (2001) досліджував різні міри робастності для поліпшення показників затримок і загального часу обробки при поломках машин. Leon et al. (1994) розробили 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 Daniels and Kouvelis (1995) 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. Cowling and Johanson (2002) і Ouelhadj et al. (2003b) 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.

***3.4. Робастне проактивне планування***

Робастні проактивні підходи планування зосереджені на створенні прогностичних графіків, які задовольняють вимогам до продуктивності в динамічному середовищі(Mehta and Uzsoy, 1999; Davenport et al., 2001; Vieira et al., 2003). Основні труднощі цього підходу полягають у визначенні прогностичних мір. Mehta and Uzsoy (1999) 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. O’Donovan et al. (1999) розширив підхід прогностичного планування Mehta and Uzsoy де показником продуктивності розкладу стало запізнення виконання операцій. Davenport et al. (2001) 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.

**4. Перепланування за наявності подій в реальному часі**

Перепланування за наявності подій в реальному часі потребує вирішення двох проблеми: коли і як реагувати на події в реальному часі. Перша проблема стосується питання коли здійснювати перепланування, а друга проблема стосується визначення стратегій перепланування для реагування на події в реальному часі.

***4.1. Коли переплановувати***

Щодо першої проблеми, коли здійснювати перепланування, в літературі були запропоновані три політики поведінки (Sabuncuoglu and Bayiz, 2000; Vieira, et al., 2003): періодична, подієва і гібридна. Періодичні та гібридні політики часто визначаються як рухомий часовий горизонт (Church and Uzsoy, 1992; Ovacik and Uzsoy, 1994; Sabuncuoglu and Karabuk, 1999; Vieira et al. 2000a; Aytug et al., 2005).

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 Muhlemann et al. (1982). 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. Ovacik і Uzsoy (1994) 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. Sabuncuoglu and Karabuk (1999) вивчали частоту 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. Kim and Kim (1994) 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. Yamamoto and Nof (1985) 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. Vieira, et al. (2000a) описали 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. Vieira et al. (2000b) 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. Church and Ozsoy (1992) 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.

***4.2. Стратегії перепланування***

Що стосується другого питання, а саме які стратегії необхідно використовувати для перепланування, у літературі вирізняються дві основні стратегії перепланування (Sabuncuoglu and Bayiz, 2000; Cowling and Johanson, 2002; Vieira et al., 2003): 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.

Sun and Xue (2001), і Dorn et al. (1995a) зазначили, 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. Abumaizar and Svestka (1997) заявляли, що на практиці перепланування проводиться за допомогою корекції плану. У той же час повне перепланування у певній мірі також використовується. Sabuncuoglu and Bayiz (2000) показали 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. Wu et al. (1991, 1993), Daniels and Kouvelis (1995), Abumaizar and Svestka (1997), Jensen (2001) використовували виміри робастності (ефективність і стабільність), для вирішення, яку стратегію перепланування краще застосовувати. Cowling and Johansson (2002) і Ouelhadj et al,. (2003b) використовували міри корисності та стабільності для оцінки продуктивності різних стратегій корекції плану і повного перепланування, що допомагало обрати кращу стратегію перепланування. Приклади застосування моделювання представлені в п. 4.4.1.

**5.** **Методи динамічного планування**

Динамічне планування здійнюється за допомогою використання наступних методів (Suresh and Chaudhuri, 1993; Shukla and Chen, 1996; Stoop and Weirs, 1996; Brandimarte and Villa, 1999): правила диспетчеризації, евристики, мета-евристики, штучний інтелект та мультиагентні системи.

***5.1.*** ***Правила диспетчеризації***

Правила диспетчеризації зіграли значну роль в динамічних умовах. Протягом багатьох років, у літературі була запропонована багато простих і складних правил диспетчеризації. 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.

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. Багато авторів використовували моделювання для оцінки ефективності правил диспетчеризації. Ramasesh (1990), Rajendran і Holthaus (1999) 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 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. Holthaus (1999) 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. Sabuncuoglu (1998) наводить 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. Shafaei і Brun (1999a) у своїй роботі 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. Kutanoglu і Sabuncuoglu (1999) представили всебічний порівняльний аналіз більше двадцяти правил диспетчеризації в динамічному виробничому середовищі зі зваженим критерієм запізнення для динамічного надходження завдань. Kim і Kim (1994) запропонували 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. Jeong and Kim (1998) використовували 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. Kutanoglu і Sabuncuoglu (2001) 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.

***5.2. Евристики***

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 графіку;

співпадаюча і часткова корекція графіку.

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

Yamamoto and Nof (1985) досліджували продуктивність евристики зсуву вправо у порівнянні з правилами диспетчеризації і повного перепланування за допомогою гілок і меж. Результати експерименту показали, що зміщення вправо перевершує за продуктивністю правила диспетчеризації і повне перепланування. Mehta і Uzsoy (1999) і O'Donovan et al. (1999) використовували евристику зсуву вправо для вставки часу простою для визначення прогностични графіків. Abumaizar and Svestka (1997) порівняли продуктивність 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.

Bean et al. (1991) запропонував 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, Akturk and Gorgulu (1999) 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. Nof and Grant (1991) 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. Kutanoglu і Sabuncuoglu (2001) запропонували кілька евристик корекції графіку за наявності поломок машин. Ці евристики корекції графіку ґрунтуються на перенаправленні задач до альтернативних машин. Lee and Uzsoy (1999) розглянули задачу мінімізації часу. Вони 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. Jain and Elmaraghy (1997) 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.

***5.3. Мета-евристики: пошук із заборонами, імітації відпалу та генетичні алгоритми***

В останні роки, мета-евристики (пошук із заборонами, імітація відпалу та генетичні алгоритми) успішно використовувалися для вирішення завдань складання виробничих розкладів. Мета-евристики – евристики високого рівня, які керують евристиками локального пошуку для виходу з локальних оптимумів (Reeves, 1995; Glover and Laguna, 1997; Pham and Karaboga, 2000). 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 пошук із заборонами, імітації відпалу і генетичні алгоритми покращують алгоритми локального пошуку для уникнення локальних оптимумів або прийняттям гірших 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. Dorn et al. (1995a) і Zweben et al. (1994) розглянули важливість використання мета-евристик для корекції розкладу замість використання локального пошуку або простих евристик, оскільки вони можуть потрапити у локальний оптимум. Mehta and Uzsoy, (1999) використовували пошук із заборонами для пошуку прогностичних графіків. Dorn et al. (1995a) використовували пошук з заборонами для корекції розкладу, викликаного невизначеністю часу обробки. Zweben et al. (1994) 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.

Chryssolouris і Subramaniam (2001) використовували 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. Rossi і Dini (2000) 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. Leon et al. (1994) і Jensen (2001) використовували генетичні алгоритми для генерації робастних графіків і для оцінки продуктивності різних мір робастності. Wu et al. (1991, 1993) 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, Bierwirth і Mattfeld (1999) 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.

***5.4. 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 коригуючих дій. ISIS (Fox, 1994; Smith, 1995) розроблений в Carnegie Mellon в 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 (Сміт, 1994) 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 (Park et al., 1996) є ще однією інтерактивною системою планування, заснованою на знаннях на основі ймовірністного та інтерактивного вирішення проблем. SONIA (Le Pape, 1994) 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 коригуючих дій для обробки події в реальному часі Belz and Mertens (1996). Деякі системи, засновані на знаннях, були розроблені, щоб допомогти користувачеві реагувати інтерактивно на події в режимі реального часу (Dutta, 1990; Sarin and Salgame, 1990; Henning and Cerda, 2000; O'Kane, 2000).

Інші Методи штучного інтелекту, які були використані для вирішення проблеми динамічного планування: ситуаційне керування, нейронні мережі, мережі Петрі і нечітка логіка. Широкі обговорення цих методів можна знайти в (Suresh and Chaudhuri, 1993; Szelke and Kerr, 1994; Zweben and Fox, 1994; Kerr and Szelke, 1995; Meziane et al., 2000).

Для отримання кращих динамічних систем планування, деякі дослідники розробили гібридні системи, які поєднують різні методи штучного інтелекту. Miyashita and Sycara (1995) розробили 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. Jahangirian and Conroy (2000), і Li et al. (2000) розробили 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 Dorn (1995b) використовується ситуаційне керування і нечітка логіка для реактивного планування в сталеливарній промисловості. Schmidt (1994) використовував 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. Dorn (1994) використовував нечітку логіку для динамічного планування безперервного розливання сталі. У (Petrovic and Duenas, 2006) 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. Garetti and Taisch (1995) and Garner and Ridley (1994) використовували 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. Ruiz et al. (2001) 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.

***5.5. Мультигентне динамічне планування***

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 (Parunak, 1996; Gou et al., 1998; Shen and Norrie, 1999; Bongaerts et al., 2000; Shen et al., 2001). 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 (Parunak, 1996; Tharumarajah and Bemelman, 1997; Shen and Norrie, 1999; Bongaerts et al., 2000; Brennan and Norrie, 2001). 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.

Figure 1. Centralised architecture

Figure 2. Centralised hierarchical architecture

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 (Parunak 1996; Brennan and Norrie, 2001; Shen et al., 2001). 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.

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 (Parunak, 1996, 2000; Shen et al., 2001; Brennan and Norrie, 2001). Мультиагентна система являє собою мережу інтелектуальних агентів, які працюють разом задля вирішення проблем, які поза їх індивідуальними можливостями O’Hare and Kennings (1996)*.* Використання мультиагентних систем для вирішення проблеми динамічного планування мотивується наступними ключовими моментами (Parunak, 1996, 2000; Shen and Norrie, 1999; Cowling et al., 2000; Brennan and Norrie, 2001; Shen et al., 2001). 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.

***5.5.1. 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.

***5.5.5.1.1. Автономні архітектури***

В автономних архітектурах (рис. 3), 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.

Figure 3. Autonomous architecture

Yams (Yet Another Manufacturing System) (Parunak, 1987) 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). Shaw (1988) розробив систему динамічного планування у модульній виробничій системі. У своїй роботі агент виробничого модуля може одночасно працювати з іншими модулями через CNP (Smith, 1980). 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. Goldsmith et al. (1998), Ouelhadj et al. (1998, 1999, 2000) запропонували 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 Sousa and Ramos (1999) 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. Cowling et al. (2001, 2003а, 2003b) і Ouelhadj et al. (2003a, 2003b) 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.

Нещодавно levelled укладання контактів були запропоновані в якості розширення CNP для підвищення економічної ефективності контрактів між агентами за наявності неповної інформації про майбутні події. Sandholm (2000) описав протокол levelled укладання контактів для автоматизованого укладання контрактів у розподіленому виробництві. Розширений протокол дозволяє агентам, що керуються власними інтересами ефективно пристосуватися до майбутніх подій, даючи можливість кожному агенту розірвати договір, просто заплативши штраф іншій стороні.

Деякі мультиагентні системи планування використовують механізми торгів і ставок для переговорів між агентами. Агенти обмінюють ресурси на гроші за цінами, визначеними за допомогою ставок. Lin and Solberg (1992, 1994) запропонував 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) (Parunak et al., 1997) 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. Aydin and Oztemel (2000) 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. Pendharkar (1999) 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.

***4.5.1.1.2. 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 (Brennan and Norrie, 1998; Shen and Norrie, 1999; Bongaerts et al., 2000; Shen et al., 2001; Tharumarajah, 2001). 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 (Gou et al., 1998; Shen and Norrie, 1999; Bongaerts et al., 2000; Shen et al., 2001). 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.

Figure 4. Mediator architecture

Медіаторна рхітектура 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. Brennan and Norrie (2001), Bongaerts et al. (2000), and Cavalieri et al. (2000) 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 Ramos (1994) 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. Maturana and Norrie (1999) 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, Shen et al. (2000) 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. Sun and Xue (2001) 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.

**6. 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: Suresh and Chaudri (1993), Shukla and Chen (1996), Stoop and Weirs (1996), and Brandimarte and Villa (1999).

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 (Glover et al., 1995; Jozefowska et al., 1998; Youssef et al., 2001; Zhou et al., 2001). 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. Parunak (1996, 2000) показали, що мультиагентні системи добре підходять для застосувань на об’єктах, що є модульними, децентралізованими, часто змінюваними, погано структурованими, і складними. Duffie and Piper (1986), Tahrumarajah and Bemelman (1997), Brennan and Norrie (2001) 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. За Sandholm (2000), 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. Shen and Norrie (1999) навели 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 Brennan and Norrie (2001), Shen and Norrie (1999), Bongaerts et al. (2000), Шен та ін. (2001), і Tharumarajah (2001) 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.

**7. Висновки**

Переважна більшість літератури, що стосується планування виробництва в основному зосереджена на пошуку оптимальних або близьких до оптимальним прогнозних графіків для простих моделей планування відносно різних критеріїв, за умови, що всі проблемні характеристики відомі. Такі прогнозні графіки часто створюються заздалегідь для керування виробничими операціями і підтримки інших заходів у сфері планування. На жаль, більшість виробничих систем працюють в динамічних середовищах, що підпадають під вплив різних подій в режимі реального часу, які можуть зробити прогностичний оптимальний розклад неприйнятним. Таким чином, динамічне планування відіграє значну роль в успішній реалізації систем планування.

Ми визначили дві категорії інформації в реальному часі, що зазвичай розглядаються в літературі: події в реальному часі події, пов'язані з ресурсами і події в режимі реального часу, пов'язані з завданнями. Динамічного планування було розділено на чотири категорії : он-лайн планування (повністю реактивні підходи) , прогностично-реактивного планування, робастне прогностично-реактивне планування і робастне превентивне планування . У повністю реактивному плануванні розкладу легко генерується за допомогою правил диспетчеризації. Проте якість рішення залишає бажати кращого через короткозору природу цих правил , оскільки при їх використанні неможливо забезпечити будь-який план для інших видів діяльності і важко передбачити продуктивність системи де рішення приймаються на локальному рівні в реальному часі зазвичай без використання глобальної інформації. Прогностично-реактивне планування є найбільш поширеним підходом динамічного планування . Прогностично-реактивні підходи здійснюють пошук у більшому просторі рішень , генерують графіки високої якості і можуть забезпечити більш високу продуктивність системи та мінімізацію експлуатаційних витрат у порівнянні з он-лайн плануванням та інтелектуальним плануванням. Прості настройки розкладу вимагають небагато зусиль і прості в реалізації . Тим не менш, вони можуть привести до погіршення продуктивності системи. Генерація робастних графіків призводить до поліпшення продуктивності системи, незважаючи на те, що міри робастності непросто визначити .

Ми обговорили дві основних альтернативи для вирішення проблеми оновлення графіків найбільш ефективним способом за наявності подій в реальному часі корекція графіка і повне перепланування. Корекція графіка відноситься до локального регулювання поточного графіка. Повне перепланування створює графік з нуля. Повне перепланування може в принципі бути більш здатним до підтримки оптимальних рішень, але такі рішення рідко досяжні на практиці, а процесорний час буде непомірно високими. Крім того, часте перепланування графіка може призводити до нестабільності і відсутності неперервності у детальних виробничих графіках. Корекція розкладу широко розповсюджена через потенційну економію процесорного часу і збереження стабільністі системи.

Було представлено кілька динамічних методів планування в тому числі: правила диспетчеризації, евристики, метаевристики, системи засновані на знаннях, нечітка логіка, нейронні мережі, мережі Петрі, гібридні методи і мультиагентні системи . Порівняльне дослідження доводить, що мультиагентні систем є дуже перспективним напрямком сучасних і майбутніх досліджень у сфері динамічного планування. Хоча були проведені деякі дослідження агентно-орієнтованих систем планування, необхідна подальша робота в цьому напрямі. Крім того, в розробці практичних комплексних динамічних систем планування, необхідно комбінувати різні методи , такі як операційні дослідження, штучний інтелект, а також необхідно наділити систему планування необхідною гнучкістю та надійністю.

Останнє але не найменш важливе, надійність є одним з ключових чинників для збереження стабільності виробничих систем за наявності невизначеностей. Маленька дослідна робота була пророблена з генерації надійних графіків; необхідні додаткові дослідження задля розробки більш загальних ефективних критеріїв надійності та стратегій перепланування.

**8. Список джерел**

1. Abumaizar, R. J. and Svestka, J. A., Rescheduling job shops under random disruptions, *International Journal of Production Research*, 35 (7), 2065-2082 (1997).
2. Akturk, M. S. and Gorgulu, E., Match-up scheduling under a machine breakdown, *European Journal of Operational Research*, 112 (1), 81-97 (1999).
3. Aydin, M. E. and Öztemel, E., job-shop scheduling using reinforcement learning agents, *Robotics and Autonomous Systems*, 33 (2-3), 169-178 (2000).
4. 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).
5. Bean, J. C., Birge, J. R., Mittenthal, J., and Noon, C. E., Match up scheduling with multiple resources release dates and disruptions, *Journal of Operations Research*, 39 (3), 471-483 (1991).
6. Belz, R. and Mertens, P., Combining knowledge-based systems and simulation to solve rescheduling problems, *Decision Support Systems*, 17 (2), 141-157 (1996).
7. Bierwirth, C. and Mattfeld, D. C., Production scheduling and rescheduling with genetic algorithms, *Evolutionary Computation*, 7 (1), 1-17, (1999).
8. Bongaerts, L., Monostori, L., McFarlane, D., and Kadar, B., Hierarchy in distributed shop floor control, *Computers in Industry*, 43 (2), 123-137 (2000).
9. Brandimarte, P. and Villa, A., *Modelling manufacturing systems: From aggregate planning to real-time control,* Springer-Verlag, 1999.
10. Brennan, R. W. and Norrie, D. H., Evaluating the performance of reactive control architectures for manufacturing production control, *Computers in Industry*, 46 (3), 235-245 (2001).
11. 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).
12. Chryssolouris, G. and Subramaniam, V., Dynamic scheduling of manufacturing job shops using genetic algorithms, *Journal of Intelligent Manufacturing*, 12 (3), 281-293 (2001).
13. Church, L. K. and Uzsoy, R. Analysis of periodic and event-driven rescheduling policies in dynamic shops, *International Journal of Computer Integrated Manufacturing*, 5 (3), 153-163 (1992).
14. 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.
15. 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.
16. Cowling, P. I. and Johansson, M., Using real-time information for effective dynamic scheduling, *European Journal of Operational Research*, 139 (2), 230-244 (2002).
17. 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).
18. 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.
19. 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).
20. Davenport, A. J., Gefflot, C., and Beck, J. C., Slack-based techniques for robust schedules, in Proceedings of the Sixth European Conference on Planning (ECP2001), 2001.
21. Dewan, P. and Joshi, S., Dynamic single-machine scheduling under distributed decision-making, *International Journal of Production Research*, 38 (16), 3759-3777 (2000).
22. 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.
23. Dorn, J., Kerr, R. M., and Thalhammer, G., Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning, *International Journal of Human Computer Studies*, 42, 687- 704 (1995a).
24. 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.
25. Duffie, N. A. and Piper, R. S., Non-Hierarchical Control of Manufacturing Systems, *Journal of Manufacturing Systems*, 5 (2), 137-139 (1986).
26. Dutta, A., Reacting to scheduling exceptions in FMS environments, *IIE Transactions*, 22 (4), 33-314 (1990).
27. 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.
28. 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.
29. 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.
30. 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.
31. Glover, F. and Laguna, M., *Tabu search*. Kluwer Academic Publishers, Boston, 1997.
32. 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.
33. Gou, L., Luh, P. B., and Kyoya, Y., Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation, *Computers in Industry*, 37 (3), 213-231 (1998).
34. Henning, G. P. and Cerda, J., Knowledge-based predictive and reactive scheduling in industrial environments, *Computers and Chemical Engineering*, 24(9), 2315-2338 (2000).
35. Herroelen, W. and Leus, R., Project scheduling under uncertainty: Survey and research potentials, *European Journal of Operational Research,* 165 (2), 289-306 (2005).
36. Holthaus, O., Scheduling in job shops with machine breakdowns: an experimental study, *Computers & Industrial Engineering*, 36 (1), 137-162 (1999).
37. Jahangirian, M. and Conroy, G. V., Intelligent dynamic scheduling system: the application of genetic algorithms, *Integrated Manufacturing Systems*, 11 (4), 247-257 (2000).
38. Jain, A. K. and Elmaraghy, H. A., Production scheduling/rescheduling in flexible manufacturing, *International Journal of Production Research*, 35 (1), 81-309 (1997).
39. Jain, A. S. and Meeran, S., Deterministic job-shop scheduling: Past, present and future, *European Journal of Operational Research*, 113 (2), 390-434 (1999).
40. 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).
41. 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.
42. Kerr, R. M. and Szelke, E., *Artificial intelligence in reactive scheduling,* Kluwer Academic Publishers, 1995.
43. Kim, M. and Kim, Y., Simulation based real-time scheduling in a flexible manufacturing system, *Journal of Manufacturing Management Systems*, 13 (2), 85-93 (1994).
44. Kouis, K., Pierreval, H., and Mebarki, N., Using multi-agent architecture in flexible manufacturing systems for dynamic scheduling, *Journal of Intelligent Manufacturing*, 8 (1), 41-47 (1997).
45. 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).
46. 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).
47. 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.
48. 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).
49. 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).
50. 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).
51. 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).
52. Lin, G. Y. and Solberg, J. J., Integrated shop floor control using autonomous agents, *IIE Transactions*, 24 (3), 57-71 (1992).
53. 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.
54. MacCarthy, B. L. and Liu, J., Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling, *International Journal of Production Research*, 31 (1), 59-79 (1993).
55. 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).
56. Mehta, S. V. and Uzsoy, R., Predictable scheduling of a single machine subject to breakdowns, *International Journal of Computer Integrated Manufacturing*, 12 (1), 15-38 (1999).
57. 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).
58. Miyashita, K. and Sycara, K., CABINS: a framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair, *Artificial Intelligence*, 76 (1), 377-426 (1995).
59. Muhlemann, A. P., Lockett, G., and Farn, C. K., Job shop scheduling heuristics and frequency of scheduling, *International Journal of Production Research*, 20 (2), 227-241 (1982).
60. Nof, S. Y. and Grant, F. H., Adaptive/predictive scheduling: review and a general framework, *Production Planning & Control*, 2 (4), 298-312 (1991).
61. 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).
62. O’Hare, G. and Jennings, N., *Foundations of Distributed Artificial Intelligence*, Wiley, New York, 1996.
63. O’kane, J. F., A knowledge-based system for reactive scheduling decision-making in FMS, *Journal of Intelligent Manufacturing*, 11 (5), 461-474 (2000).
64. 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.
65. 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.
66. 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.
67. 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.
68. 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.
69. 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).
70. 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).
71. Parunak, H. V., Manufacturing experience with the contract net, in Huhns, M. (Eds.), *Distributed Artificial Intelligence*, pp. 285-310, Pitman, London, 1987.
72. 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*, Chapter 4, Wiley Inter-science, New York, 1996.
73. 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.
74. 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).
75. 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).
76. 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*. Pham, D. T. and Karaboga, D., *Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks,* Springer, London, 2000.
77. Pinedo, M., *Scheduling theory, algorithms and systems,* First edition, Prentice Hall, 1995.
78. Pinedo, M., *Scheduling theory, algorithms and systems,* Second edition, Prentice Hall, 2002.
79. Rajendran, C. and Holthaus, O., A comparative study of dispatching rules in dynamic flow shops and job shops, *European Journal of Operational Research*, 116 (1), 156-170 (1999).
80. Ramasesh, R., Dynamic job shop scheduling: a survey of simulation research, *OMEGA International Journal of Management Science*, 18 (1), 43-57 (1990).
81. 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.
82. Reeves, C. R., *Modern heuristic techniques for combinatorial problems,* John Wiley & Sons, McGraw-Hill International (UK) Limited, 1995.
83. 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).
84. Ruiz, D., Canton, J., and Mara, N. J., Espuna, A. and Puigjaner, L., On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants, *Computers and Chemical Engineering*, 25 (4), 829-837 (2001).
85. Sabuncuoglu, I., A study of scheduling rules of flexible manufacturing systems: a simulation approach, *International Journal of Operational Research*, 36(2), 527-546 (1998).
86. 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).
87. Sabuncuoglu, I. and Bayiz, M., Analysis of reactive scheduling problems in a job shop environment, *European Journal of Operational Research*, 126 (3), 567-586 (2000).
88. 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).
89. Schmidt, G., How to apply fuzzy logic to reactive scheduling, in Szelke, E. and Kerr, R. M. (Eds.), *Knowledge-based Reactive Scheduling,* pp. 57-67, North-Holland, 1994.
90. 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).
91. Shafaei, R. and Brunn, P., Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment, *International Journal of Production Research*, 37 (18), 4105-4117 (1999b).
92. Shaw, J. M., Dynamic scheduling in cellular manufacturing systems: a framework for Network decision making, *Journal of Manufacturing Systems*, 7 (2), 83-94 (1988).
93. 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).
94. Shen, W., Maturana, F., and Norrie, D. H., Metaphor II: an agent-based architecture for distributed intelligent design and manufacturing, *Journal of Intelligent Manufacturing*, 11 (3), 237-251 (2000).
95. Shen, W., Norrie, D. H., and Barthes, J. P. A., *Multi-agent systems for concurrent intelligent design and manufacturing*. Taylor & Francis, London, 2001.
96. 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).
97. Smith, R., The contract net protocol: high level communication and control in distributed problem solver, *IEEE Transactions on Computers*, 29 (12), 1104-1113 (1980).
98. Smith, F. S., OPIS: A methodology and architecture for reactive scheduling, in Zweben, M. and Fox, M. S., *Intelligent Scheduling*, pp. 29-66, Morgan Kaufmann Publishers, INC, 1994.
99. Smith, F. S., Reactive scheduling systems, in Brown, D. and Scherer, W. T. (Eds.), *Intelligent Scheduling Systems*, pp. 155-192, Kluwer Academic Publisher, 1995.
100. Sousa, P. and Ramos, C., A distributed architecture and negotiation protocol for scheduling in manufacturing systems, *Computers in Industry*, 38 (2), 103-113 (1999).
101. Stoop, P. P. M. and Weirs, V. C. S., The complexity of scheduling in practice, *International Journal of Operations and Production management*, 16 (10), 37-53 (1996).
102. 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).
103. Suresh, V. and Chaudhuri, D., Dynamic scheduling a survey of research, *International Journal of Production Economics*, 32 (1), 53-63 (1993).
104. Szelke, E. and Kerr R. M., *Knowledge-based reactive scheduling,* North-Holland, 1994.
105. Tharumarajah, A. and Bemelman, R., Approaches and issues in scheduling a distributed shop-floor environment, *Computers in Industry*, 34 (1), 95-109 (1997).
106. Tharumarajah, A., Survey of resource allocation methods for distributed manufacturing systems, *Production Planning & Control*, 12 (1), 58-68 (2001).
107. 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).
108. 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).
109. Vieira, G. E., Hermann, J. W., and Lin, E., Rescheduling manufacturing systems: a framework of strategies, policies and methods, *Journal of Scheduling*, 6 (1), 36-92 (2003).
110. Weirs, V. C. S., A review of the applicability of OR and AI scheduling techniques in practice, *Omega International Journal of Management Science*, 25 (2), 145-153 (1997).
111. Wu, S. D., Storer, R. H. and Chang, P. C., A rescheduling procedure for manufacturing systems under random disruptions, in *Proceedings Joint USA/German Conference on New Directions for Operations Research in Manufacturing,* pp. 292-306, 1991.
112. 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).
113. Yamamoto, M. and Nof, S. Y., Scheduling/rescheduling in the manufacturing operating system environment, *International Journal of Production Research*, 23 (4), 705-722 (1985).
114. 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.
115. 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.
116. Zweben, M. and Fox, M. S., *Intelligent scheduling,* Morgan Kaufmann Publishers, INC, 1994.
117. Zweben, M., Daun, B., and Deale, M., Scheduling and rescheduling with iterative repair, in Zweben, M. and Fox, M. S. (Eds.), *Intelligent Scheduling*, pp. 241-254, Morgan Kaufmann Publishers, INC, 1994.