

УДК 621 (075.8)

В.М. КОРЕНЬКОВ, І.І. ТКАЧ

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

## ЕВРИСТИЧНИЙ АЛГОРИТМ СИНТЕЗУ МАРШРУТНИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

*В роботі розглянуто задачу синтезу маршрутних технологічних процесів механообробки на основі мультиагентного алгоритму. Розроблено математичну модель, яка дозволяє обирати найбільш імовірні послідовності технологічних операцій на множині усіх технічно допустимих рішень. Частина параметрів моделі є змінними у часі, що дозволяє оперативно реагувати на зміну виробничої обстановки. Модель базується на використанні евристик, що дозволяє гнучко підлаштовувати її до вимог виробництва.*

*Ключові слова: синтез технологічних процесів, алгоритм Ant Colony Optimization, CAPP-системи, Times Dependent Travelling Salesman Problem.*

В.Н. КОРЕНЬКОВ, И.И. ТКАЧ

Национальный технический университет Украины  
«Киевский политехнический институт имени Игоря Сикорского»

## ЭВРИСТИЧЕСКИЙ АЛГОРИТМ СИНТЕЗА МАРШРУТНЫХ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

*В работе рассмотрена задача синтеза маршрутных технологических процессов механообработки на основе мультиагентного алгоритма. Разработана математическая модель, которая позволяет выбирать наиболее вероятные последовательности технологических операций на множестве всех технически допустимых решений. Часть параметров модели являются переменными во времени, что позволяет оперативно реагировать на изменение производственной обстановки. Модель базируется на использовании эвристик, что позволяет гибко подстраивать ее к требованиям производства.*

*Ключевые слова: синтез технологических процессов, алгоритм Ant Colony Optimization, CAPP-системы, Times Dependent Travelling Salesman Problem.*

V. KORENKOV, I. TKACH

National Technical University of Ukraine  
«Igor Sikorsky Kyiv Polytechnic Institute»

## HEURISTIC ALGORITHM OF PRODUCT PROCESSES SYNTHESIS

*The paper is about the problem of product process synthesis of machining based on the multiagent algorithm. A developed mathematical model allows us to choose the most probable sequences of technological operations on the set of all technically possible solutions. Some of the model's parameters are variable in time that allows reacting promptly to changes in the production environment. The model is based on the use of heuristics that makes it flexible and adaptive to the production requirements.*

*Key words: process planning, Ant Colony Optimization algorithm, CAPP-system, Times Dependent Traveler Salesman Problem.*

### Постановка проблеми

Сучасна концепція PLM (англ. Product Lifecycle Management) передбачає наскрізну інтеграцію всіх автоматизованих систем підтримки життєвого циклу виробів. На сьогодні, досить багато зусиль прикладається задля автоматизації однієї з ключових підсистем – проектування технологічних процесів CAPP (англ. Computer-Aided Process Planning). Основні перешкоди тому наступні.

Історично, обмін інформацією між конструктором і технологом був людино-орієнтованим і подібного роду системи розвивались як діалогові [1]. Перша проблема полягає у тому, що відомі методи синтезу технологічних процесів (ТП) вимагають окрім геометричної ще і інформації щодо функціонального призначення певного типового елемента конструкції, яка на даний час відсутня в переважній більшості CAD-систем. В результаті, інженер-технолог змушений виконувати свого роду «reverse engineering» даних, тобто відновлювати ту початкову інформацію, яку конструктор в невідомому виді закладав при моделюванні об'єктів: розпізнавання типових геометричних примітивів з завершених моделей; розпізнавання функціонального призначення кожного з примітивів; встановлення зв'язку між

геометричним/функціональним об'єктом та типовим технологічним рішенням тощо.

Друга проблема полягає у тому, що не може бути єдиного підходу до розробки ТП, придатного для використання на різних підприємствах. Цей факт підтверджується безуспішністю впровадження формально-логічних моделей синтезу ТП впродовж кількох десятиліть (на даний час їм на заміну прийшли евристичні методи з гнучкою системою налаштувань).

Третя проблема – неможливо розробити ТП без оперативної інформації з виробництва. Тобто алгоритми синтезу повинні самостійно адаптуватись до навколишніх змін, а не діяти лише на основі «статичної» інформації. Ну і звісно, накопичення та використання знань; вирішення оптимізаційних задач – все це на даний час присутнє лише частково і у декількох розробках PLM-систем, які, як правило, недоступні для малих та середніх підприємств.

#### **Аналіз останніх досліджень і публікацій**

Для початку слід визначитись, на якому рівні автоматизації знаходяться сучасні CAD/CAPP-системи.

Як відомо, технологічна та конструкторська (геометрична) інформаційні моделі не тотожні. Для встановлення зв'язку між ними на сьогодні використовують два принципово протилежні підходи: розпізнавання геометрії та функціональне моделювання.

Перший підхід реалізовано у переважній більшості CAD-систем у вигляді Features-Recognize функцій, однак вони обмежені лише набором типових елементів (наприклад, площина, отвір, паз, ...), що недостатньо для синтезу ТП більшості деталей. Другий підхід полягає у видозміні процесу моделювання – конструктор може оперувати лише компонентами, що мають однозначну відповідність певним процесам обробки (наприклад, CATIA Functional Modeling). Слід відзначити, що в обох випадках необхідне втручання людини для однозначної ідентифікація геометрії та співставленні типового технологічного рішення. На досить високому рівні розвинені PIM модулі (англ. Product Information Modeling), які дозволяють автоматизувати аналіз розмірної інформації (до того, використовували ряд опосередкованих методів, наприклад кодування точності розмірів кольором поверхонь).

Зворотній зв'язок з виробництвом на досить високому рівні реалізовано у вигляді SCADA-систем: наразі, майже все обладнання використовує ЧПК і дані моніторингу через мережу потрапляють на сервери систем управління. SCADA та MES-системи представляють собою одні з найбільш затребуваних та впроваджуваних систем.

Стосовно технологічних рішень, основним інструментом на сьогодні є механізм шаблонів в NC/CAPP-системах. Як правило у технологів накопичується бібліотека шаблонів (або така передається від замовника), у якій зафіксовано майже всі типові для даного виробництва параметри, окрім оброблюваної геометрії. При використанні технолог повинен у дерево проекту у правильній послідовності додавати бібліотечні шаблони, змінюючи при цьому лише посилання на геометричні моделі оброблюваних елементів.

Саме остання задача найбільш відповідальна, а з іншого боку – найменш формалізована та майже неавтоматизована. Зважаючи на доступність даних моніторингу обладнання та наявного інструменту представлення типових технологічних рішень, в даній роботі буде представлено спробу комплексної автоматизації процес синтезу маршруту обробки деталей.

#### **Формулювання мети дослідження**

Перед описанням математичної моделі, слід визначитись з основними інформаційними одиницями, якими буде оперувати дана модель.

Матеріальний об'єкт в процесі виробництва послідовно змінює свій стан (форму, хімічні та фізичні властивості тощо). Подібні стани фіксовані, тобто однозначно вирізняються в часі і просторі, що дозволяє їх розглядати як об'єкти планування і управління (це твердження справедливе для дискретного виробництва, до якого, власне, відноситься механообробка). Кожен стан однозначно характеризується геометричною складовою  $G$ , послідовністю дій  $P$  та використовуваними ресурсами  $R$ , і на логічному рівні може бути представлений кортежем  $S = \langle G, P, R \rangle$ . На даний час, такий підхід взятій за основу всіх сучасних PLM систем (3DEXPERIENCE - Dassault Systèmes, Siemens PLM Software, PTC Creo PLM та ін.) у вигляді моделі даних PPR (Product - Process - Resource).

Математична абстракція  $S$  представляє собою мінімальну інформаційну одиницю і тотожна такому поняттю як «технологія обробки геометричного елемента». У такий спосіб, кожному геометричному елементу деталі можна поставити у відповідність кінцеве число кортежів  $S$ , а вся їх множина, представлена декартовим добутком  $S \times S$  – називатиметься простором рішень (множиною всіх технічно допустимих варіантів обробки даної деталі). У цій множині:

- представлені тільки унікальні екземпляри в цілому, хоча деякі складові  $S$  можуть повторятись (наприклад:  $S_1 \langle G_1, P_1, R_1 \rangle$ ;  $S_2 \langle G_1, P_2, R_1 \rangle$ ;  $S_3 \langle G_1, P_3, R_1 \rangle$  - можна трактувати, як декілька варіантів обробки однієї і тієї ж поверхні за допомогою одних і тих же ресурсів);
- ключовою є геометрична інформація, інші види інформації (процеси і ресурси) - залежні від неї і вторинні; між собою не мають пріоритетів;

- частина кортежів  $S$  «константна», частина – залежна від часу, тобто  $S(t)$  в певні проміжки часу повинні оновлюватися для підтримання актуальності моделі.

Пошук траєкторії – упорядкованого ланцюжка  $S_i \rightarrow S_j \rightarrow S_k \rightarrow \dots$  на просторі рішень, представляє собою процес синтезу структури технологічного процесу [2,3], а формалізація та алгоритмізація процесу пошуку – мету даної роботи.

Слід відзначити, що простір рішень в загальному випадку є  $n$ -мірним, де  $n$  - кількість сутностей, які впливають на прийняття рішення (наприклад, комбінація даних про «геометричний елемент; обладнання; інструмент; оснастку» складається з 4-х сутностей і відповідно буде описуватись 4-вимірним простором для пошуку рішення).

З огляду на практичну реалізацію даної математичної моделі у вигляді програмного коду, кортеж  $S$  представляється об'єктною моделлю, а простір рішень – орієнтованим графом, кожна вершина якого є об'єктом (описує стан), а зв'язок отождожене упорядкованість у просторі та часі послідовності станів.

Дану задачу можна класифікувати як задачу комівояжера, що залежить від часу  $t$ , в зарубіжній літературі – задача TDTSP (Times Dependent Travelling Salesman Problem). У термінах теорії графів її можна сформулювати наступним чином: в зваженому графі  $G = (S, E)$  знайти Гамільтонів цикл найменшої ваги.

Якщо є упорядкована пара:  $S = \{s_1, s_2, \dots, s_n\}$  непустиї множини вершин та  $E = \{s_i, s_j \in S, i \neq j\}$  множини ребер, а також задана матриця ваги  $M(t) = \{\mu_{ij}(t) | i \neq j\}$ , значення якої змінюються в часі, то цільова функція  $f$  визначається як сумарна довжина маршруту, що покриває всі вершини з  $S$ , а задача TDTSP зводиться до знаходження циклу  $A(t)$ , що проходить по всіх вершинах рівно один раз, причому:

$$f(t) = \sum_{i \in A} \mu_{ij}(t) \rightarrow \min$$

Дана задача оптимізації є NP-повною, тобто її неможливо розв'язати за поліноміальний час [3].

Існує декілька різновидів методів вирішення таких задач. Методи повного перебору всіх варіантів і спрямованого перебору (наприклад, гілок і меж) теоретично дозволяють знайти точне оптимальне рішення, але не використовуються, так як їх алгоритмічна складність оцінюється як  $O(n!)$ . Наприклад, якщо для виготовлення відносно простої деталі потрібно обробити  $\sim 10$  конструктивних елементів, для кожного з яких існує по 2..4 методи обробки, то така задача вже буде в принципі необчислювана за прийнятний час.

Внаслідок цього, використовуються різні евристичні та наближені методи. Вони не гарантують отримання оптимального рішення, але добре масштабуються, тобто дають можливість вирішувати завдання незалежно від їх розмірності, а також ефективні по швидкості [4]. Досить добре себе зарекомендували методи, засновані на механізмах, що використовуються в природі (наприклад, генетичні алгоритми, нейронні мережі, ройовий інтелект та ін.).

У даній роботі використовується модифікація мета-евристичного алгоритму Ant Colony Optimization (ACO), призначеного для вирішення задач комбінаторної оптимізації на графовій моделі [5-10].

#### Викладення основного матеріалу дослідження

Перш ніж перейти до опису алгоритму, слід уточнити, що саме розуміється під поняттям ваги  $\mu_{ij}(t)$  графа. Адже його вершинами є об'єкти  $S$ , що уособлюють досить велику кількість сутностей з абсолютно різними непорівнянними параметрами.

В даній роботі ключовою особливістю є те, що всі параметри (незалежно від їх розмірності і суті) приводяться до одного значення – імовірності вибору певного рішення з допустимого списку, на кожному кроці процесу синтезу. У такий спосіб розраховується імовірність  $p_i$  для кожного процесного чи ресурсного параметра, а загалом для кожної пари  $\langle S_i, S_j \rangle$ , за законом множення імовірностей, вага:

$$\mu_{ij} = (1 - p_1) \cdot (1 - p_2) \cdots (1 - p_n)$$

Тут різниця  $(1 - p_i)$  використовується з тієї причини, що в даній роботі розглядається задача мінімізації і алгоритм, який її реалізує, призначено саме для пошуку мінімальних маршрутів на графах, у той час як при прийнятті рішень слід орієнтуватись на максимальні значення  $p_i$ . Тому для пошуку найбільш імовірних маршрутів  $S_i \rightarrow S_j \rightarrow S_k \rightarrow \dots$  на графах слід привести значення вагових коефіцієнтів до мінімальних.

Отже, у відповідності до алгоритму ACO імовірність переходу з  $i$ -ї вершини в  $j$ -ту вершину визначається наступним чином [11]:

$$P_{ij}^k = \begin{cases} \frac{(\mu_{ij})^\alpha \cdot (\tau_{ij})^\beta}{\sum_{u \in N_i^k} (\mu_{iu})^\alpha \cdot (\tau_{iu})^\beta}, & j \in N_i^k, \\ 0, & j \notin N_i^k \end{cases}$$

де:

- $\mu_{ij}$  представляє апіорну (обчислену до виконання  $k$ -ї ітерації алгоритму) ефективність переходу з вершини  $i$  в  $j$  на основі деякої евристики;
- $\tau_{ij}$  представляє апостеріорну (обчислену на  $k-1$  ітерації алгоритму) ефективність переходу з вершини  $i$  в  $j$ ;
- $N_i^k$  множина, що визначає набір допустимих вершин для переходу  $k$ -го агента, пов'язаних з  $i$ -ю вершиною графа.

При обчисленні ймовірності переходу  $P_{ij}^k$ , баланс впливу апостеріорної ефективності  $\tau_{ij}$  (відображає історію успішних дій) і евристичної інформації  $\mu_{ij}$  досягається шляхом підбору коефіцієнтів  $\alpha$  і  $\beta$  (у випадку  $\beta = 0$ , результатом попереднього пошуку ( $k-1$ -м агентом нехтують і алгоритм вироджується в «жадібний»).

Ідея мультиагентного алгоритму полягає у моделюванні поведінки агентів, пов'язаних з їх здатністю адаптуватися до умов, що змінюються, знаходячи новий найкоротший шлях від початкової до кінцевої точки маршруту.

При своєму русі по графу,  $k$ -й агент на основі значення  $P_{ij}^k$  приймає рішення про перехід від  $i$ -ї до  $j$ -ї вершини і залишає відповідну відмітку у вигляді значення  $\tau_{ij}$ , тобто помічає пройдений шлях. Сумарне значення відміток відповідає інтенсивності руху агентів на певній ділянці, і ця інформація використовується іншими агентами для вибору шляху – у такий спосіб відбувається пошук найбільш інтенсивних маршрутів на графі (тобто найімовірніших послідовностей обробки).

В даній роботі значення інтенсивності визначається за формулою:

$$\tau_{ij}(t+1) = f \cdot \tau_{ij}(t) + \begin{cases} G/L_k, & (i,j) \in T_k, \\ 0, & (i,j) \notin T_k, \end{cases}$$

де:  $f$  – параметр, що контролює швидкість зростання сумарного коефіцієнта інтенсивності;  $T_k$  – маршрут, пройдений  $k$ -м агентом на ітерації  $t$ ;  $L_k$  – довжина цього маршруту;  $G$  – параметр регулювання алгоритму, значення якого вибирають одного порядку з довжиною оптимального маршруту.

В даній роботі було виконано певну модифікацію базового алгоритму з наступних причин.

У класичній постановці задачі комівояжера, з метою гарантії існування рішення, вважається, що вихідний граф є повним, тобто, між довільною парою його вершин існує ребро (повний граф з  $n$  вершинами має  $n(n-1)/2$  ребер).

Однак при моделюванні технології механічної обробки слід врахувати обмеження (наприклад, геометричні), що накладаються на послідовність виконання переходів. В такому випадку, вирішувана задача стає асиметричною і, відповідно, вихідний граф може бути в загальному випадку неповним та орієнтованим (наявність хоча б одного гамільтонового циклу гарантується тим, що конструкція деталі початково вважається безпомилковою і, отже, повністю оброблюваною). Обмеження на переходи може бути накладено або шляхом видалення відповідного ребра графа, або введенням додаткового параметру у ваговий коефіцієнт з імовірністю переходу 0%.

У класичній постановці алгоритму АСО, заздалегідь відома принаймні початкова точка маршруту і, відповідно, алгоритм виконується лише один раз. При виборі маршруту обробки, початкова вершина невідома (тобто ми не знаємо наперед, з якою поверхні почати обробку деталі). З цієї причини, ми змушені виконувати алгоритм  $N$  раз (тобто для кожної поверхні у якості початкової), що з одного боку значно сповільнює розрахунки, а з іншого – гарантує пошук траєкторії по всьому простору рішень.

#### Висновки

У даній роботі приведено опис евристичного алгоритму синтезу структури технологічних процесів механічної обробки деталей довільних конструкцій, призначений для використання в сучасних САРР-системах, який дозволяє автоматизувати процес розробки маршрутної технології.

Особливістю даного алгоритму є те, що він оперує лише поняттям ймовірності переходу по графу альтернатив, що саме по собі дає можливість звести до єдиної метрики і тим самим замінити вихідну багатопараметричну задачу однопараметричною на кожному кроці пошуку. Даний підхід суттєво спрощує пошук екстремумів і дозволяє для їх визначення використовувати надійні і ефективні методи однопараметричної оптимізації.

Програмна реалізація алгоритму розміщена на web-сторінці [12]

## Список використаної літератури

1. А.Волков, И.Пасынков, А.Саранчин, С.Чечиков. Pro/TechDoc — средство разработки технологических процессов в системе Pro/ENGINEER. Режим доступа: <http://sapr.ru/article/15611>, вільний (дата звернення: 10.01.2018). — Заголовок з екрана
2. Шарин Ю.С., Якимович Б.А., Толмачев В.Г., Коршунов А.И. Теория сложности. - Ижевск : Изд-во ИжГТУ, 1999. - 132 с.
3. Основы конструкторско-технологической информатики: учеб. пособие / О.В. Захаров, В.В. Горшков. Саратов: Сарат. гос. техн. ун-т, 2009. 190 с. ISBN 978-5-7433-2074-5
4. Matrenin P. Adaptive particle swarm algorithm for NP-hard problems. Режим доступа: [https://www.researchgate.net/publication/281065333\\_Adaptivnyj\\_algoritm\\_roa\\_castic\\_dla\\_resenia\\_NP-trudnyh\\_zadac](https://www.researchgate.net/publication/281065333_Adaptivnyj_algoritm_roa_castic_dla_resenia_NP-trudnyh_zadac), вільний (дата звернення: 10.01.2018). — Заголовок з екрана
5. Матренин П.В. Разработка и исследование адаптивных методов роевого интеллекта в задачах календарного планирования // Автоматика и программная инженерия. 2013, №1(3), стр.109-114
6. Dorigo, M., Stutzle, T. Ant colony optimization. The MIT press, 2004
7. Штовба С. Д. Муравьиные алгоритмы. Режим доступа: <http://aco.inf.ua/pdf/ma.pdf>, вільний (дата звернення: 10.01.2018). — Заголовок з екрана
8. Zhou Rong Dynamic job shop scheduling using ant colony optimization algorithm based on a multi-agent system // A thesis submitted for the degree of doctor of philosophy department of mechanical engineering national university of singapore. – 2007. - B.Eng., South China University of Technology, P.R. China. 189pp
9. Atabak Elmi, Seyda Topaloglu. Multi-degree cyclic flow shop robotic cell scheduling problem: Ant colony optimization // Computers & Operations Research. Volume 73, September 2016, Pages 67-83
10. Edson Flórez, Wilfredo Gómez, and MSc. Lola Bautista An Ant Colony Optimization algorithm for job shop scheduling problem Режим доступа [https://www.researchgate.net/publication/256846085\\_An\\_ant\\_colony\\_optimization\\_algorithm\\_for\\_job\\_shop\\_scheduling\\_problem](https://www.researchgate.net/publication/256846085_An_ant_colony_optimization_algorithm_for_job_shop_scheduling_problem), вільний (дата звернення: 10.01.2018). — Заголовок з екрана
11. Ю.Зайченко, Н.Мурга Исследование муравьиных алгоритмов оптимизации в задаче коммивояжер // International Journal "Information Models and Analyses" Vol.2 / 2013, Number 4, 370-384pp.
12. [https://github.com/korenkov/mes/blob/master/ant\\_collony\\_optimization.py](https://github.com/korenkov/mes/blob/master/ant_collony_optimization.py)