Dr inż. Jarosław Eugeniusz Pempera

Katedra Automatyki, Mechatroniki i Systemów Sterowania

Wydział Elektroniki, Politechnika Wrocławska

Załącznik 2. do Wniosku

o przeprowadzenie postępowania habilitacyjnego

AUTOREFERAT

przedstawiający opis osiągnięć naukowych, w szczególności wskazanie osiągnięcia naukowego wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2017 r. poz. 1789)

Pempera

Spis treści

Pemp pem

1 Imię i Nazwisko

Jarosław Eugeniusz Pempera

2 Posiadane dyplomy, stopnie naukowe

- Doktor nauk technicznych, dyscyplina: Automatyka i Robotyka,
  Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, 2001 r.

  Tytuł rozprawy doktorskiej: *Algorytmy szeregowania zadań w pewnym dyskretnym procesie produkcyjnym*

  Promotor: prof. dr hab. inż. Józef Grabowski.

  Rozprawa została wyróżniona przez Radę Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej.

- Magister inżynier, kierunek: Elektronika i telekomunikacja, 1994 r.

3 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

- 2001 r. - obecnie, Adiunkt, Katedra Automatyki, Mechatroniki i Systemów Sterowania, Wydział Elektroniki Politechniki Wrocławskiej,
- 1997 r. - 2001 Asystent, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej.

4 Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2017 r. poz. 1789)

4.1 Tytuł osiągnięcia naukowego/artystycznego

„Opracowanie efektywnych algorytmów harmonogramowania zadań w systemach produkcyjnych, w których występują dodatkowe ograniczenia"

4.2 Publikacje wchodzące w skład osiągnięcia naukowego

Osiągnięcie naukowe stanowi cykl publikacji poświęconych zagadnieniom harmonogramowania zadań w systemach produkcyjnych z dodatkowymi ograniczeniami. Wspólna tematyka dotyczy szerokiej gamy problemów szeregowania zadań, w których występują takie ograniczenia jak: brak buforów pomiędzy stanowiskami, brak możliwości czekania

Pempera

pomiędzy wykonywaniem kolejnych operacji zadania, brak możliwości przestoju maszyny oraz wymóg produkcji cyklicznej.

Cykl składa się z 10 publikacji, z których 8 znajduje się w bazie JCR. Sumaryczny impact faktor (z roku publikacji) prac przedstawionych w ramach cyklu wynosi 14,837 punktów zgodnie z punktacją w roku opublikowania i przeszło 21 prac zgodnie z punktacją z 2019 roku, natomiast sumaryczna liczba punktów według Ministerstwa Nauki i Szkolnictwa Wyższego (MNiSW) wynosi ponad 234 punkty. Łączna liczba cytowań (na dzień 15.04.2019) wynosi 221 według Web of Knowledge Core Collection.

[C1] Grabowski Józef (50%), **Pempera Jarosław (50%)**: The permutation flow shop problem with blocking. Omega (Oxford). 2007, vol. 35, nr 3, s. 302-311.
JCR, IF 01.327, MNiSW 2010: 32, Liczba cytowań WoS 87.

Mój wkład oceniam na 50%. Jestem autorem twierdzenia blokowego dla problemu przepływowego bez buforów. W twierdzeniu, w oparciu o klasyczne bloki zadań oraz tzw. antybloki wyznaczone dla pewnej permutacji wykonywania zadań, sformułowano warunki konieczne jakie musi spełniać permutacja lepsza pod względem funkcji C_{max}. Ponadto skonstruowałem algorytm bazujący na metodzie przeszukiwania z zabronieniami operujący na sąsiedztwie składającym się z rozwiązań spełniających warunki twierdzenia, zaimplementowałem mechanizm multiruchów zwiększający efektywność algorytmu, przeprowadziłem badania efektywności algorytmu oraz sformułowałem wstępne wnioski wynikające z przeprowadzonych badań. Skonstruowałem akcelerator istotnie redukujący czas przeszukiwania sąsiedztwa.

[C2] Grabowski Józef (50%), **Pempera Jarosław (50%)**: Some local search algorithms for no-wait flow-shop problem with makespan criterion. Computers & Operations Research. 2005, vol. 32, nr 8, s. 2197-2212.
JCR, IF 00.746, MNiSW 2010: 32, Liczba cytowań WoS 110.

Mój wkład oceniam na 50%. Sformułowałem warunki jakie muszą spełniać niezależne ruchy tak, aby poprawy uzyskiwane każdym z nich sumowały się. Skonstruowałem algorytm oparty na metodzie przeszukiwania z zabronieniami, w którym zbieżność przeszukiwania do lokalnych minimów wzmocniona jest mechanizmem multiruchów. Skonstruowałem akcelerator wyznaczający w zamortyzowanym czasie $O(1)$ wartość funkcji celu dla każdego rozwiązania, przeprowadziłem badania eksperymentalne algorytmów oraz sformułowałem wstępne wnioski z badań.

[C3] **Pempera Jarosław (75%)**, Smutnicki Czesław: Harmonogramowanie cykliczne w przepływowym systemie produkcyjnym z ograniczeniem bez czekania. W: Automatyzacja procesów dyskretnych : teoria i zastosowania. T. 1 / pod red. Andrzeja Świerniaka i Jolanty Krystek. Gliwice : Wydawnictwo Politechniki Śląskiej, 2018. s. 159-167.

Mój wkład oceniam na 75%. Jestem autorem unikalnej metody wyznaczenia najdłuższych dróg w grafie modelującym problem przepływowy z pomijaniem maszyn z ograniczeniami bez czekania. Należy podkreślić, że wspomniany graf jest grafem cyklicznym, natomiast w prezentowanym algorytmie dzięki odpowiedniej kolejności relaksacji krawędzi złożoność zmniejszono do znacznie mniejszej złożoności proporcjonalnej do liczby łuków.

Pempera

[C4] **Pempera Jarosław (100%)**: An exact block algorithm for no-idle RPQ problem. Archives of Control Sciences. 2017, vol. 27, nr 2, s. 323-330.
JCR, IF 01.545 (2017), MNiSW 2013-2016: 15.

[C5] Bożejko Wojciech (33%), **Pempera Jarosław (33%)**, Smutnicki Czesław (33%): Parallel tabu search algorithm for the hybrid flow shop problem. Computers & Industrial Engineering. 2013, vol. 65, nr 3, s. 466-474.
JCR, IF 01.690, MNiSW 2013: 35, 2013-2016: 40, Liczba cytowań WoS 26.

Mój wkład oceniam na 33%. Polegał on na opracowaniu akceleratora obliczeń wartości funkcji celu dla otoczenia blokowego w elastycznym problemie przepływowym oraz implementacji algorytmu bazującego na metodzie przeszukiwania z zabronieniami, przeprowadzenie testów komputerowych i sformułowanie wniosków z badań.

[C6] **Pempera Jarosław (75%)**, Smutnicki Czesław (25%): Open shop cyclic scheduling. European Journal of Operational Research. 2018, vol. 269, nr 2, s. 773-781.
JCR, IF 03.428 (2017), MNiSW 2013-2016: 40.

Mój wkład oceniam na 75 %. Polegał on na opracowaniu efektywnej czasowo metody wyznaczania czasu cyklu w cyklicznych problemach szeregowania zadań, określeniu warunków jaki musi spełniać kolejność wykonywania operacji, aby można było skonstruować realizowalny harmonogram cykliczny, sformułowaniu twierdzeń blokowych, skonstruowaniu algorytmu opartego na metodzie przeszukiwań z zabronieniami, przeprowadzenie testów komputerowych, współudziale w sformułowaniu wniosków i redakcji pracy.

[C7] Bożejko Wojciech (25%), Gnatowski Andrzej (25%), **Pempera Jarosław (25%)**, Wodecki Mieczysław (25%): Parallel tabu search for the cyclic job shop scheduling problem. Computers & Industrial Engineering. 2017, vol. 113, s. 512-524.
JCR, IF 03.195 (2017), MNiSW 2013-2016: 40, Liczba cytowań 4.

Mój wkład oceniam na 25 %. Polegał on na sformułowaniu twierdzeń blokowych dla cyklicznego problemu gniazdowego, na skonstruowaniu algorytmu opartego na metodzie przeszukiwania z zabronieniami, przeprowadzeniu testów komputerowych, współudziale w sformułowaniu wniosków i redakcji pracy.

[C8] Bożejko Wojciech (33%), **Pempera Jarosław (33%)**, Wodecki M.(33%): A fine-grained parallel algorithm for the cyclic flexible job shop problem. Archives of Control Sciences. 2017, vol. 27, nr 2, s. 169-181.
JCR, IF 01.545 (2017), MNiSW 2012: 35, 2013: 35, 2013-2016: 40.

Mój wkład oceniam na 33 %. Polegał on na opracowaniu efektywnej czasowo metody wyznaczania czasu cyklu oraz sformułowaniu twierdzeń blokowych dla elastycznego cyklicznego problemu gniazdowego, współudział w opracowaniu metody równoległego wyznaczenia czasu cyklu, przeprowadzeniu testów komputerowych, współudziale w sformułowaniu wniosków i redakcji pracy.

Pempera

[C9] Bożejko Wojciech (33%), **Pempera Jarosław (33%)**, Wodecki Mieczysław (33%): Minimal cycle time determination and golf neighborhood generation for the cyclic flexible job shop problem. Bulletin of the Polish Academy of Sciences. Technical Sciences. 2018, vol. 66, nr 3, s. 333-344.
JCR, IF 01.361 (2017), MNiSW 2013-2016: 25.

Mój wkład oceniam na 33 %. Polegał on na opracowaniu efektywnej czasowo metody wyznaczania czasu cyklu oraz sformułowaniu twierdzeń blokowych dla elastycznego cyklicznego problemu gniazdowego, opracowaniu metody wieloetapowego szacowania wartości czasu cyklu, konstrukcji algorytmu, przeprowadzeniu testów komputerowych, współudziale w sformułowaniu wniosków i redakcji pracy.

[C10] Bożejko Wojciech (33%), **Pempera Jarosław (33%)**, Wodecki Mieczysław (33%): Minimization of the number of employees in manufacturing cells. International Joint Conference SOCO'18-CISIS'18-ICEUTE'18, San Sebastián, Spain, June 6–8, 2018 : proceedings / Manuel Grana [i in.] Eds. Cham : (Advances in Intelligent Systems and Computing, ISSN 2194-5357; vol. 771) Springer, cop. 2019. s. 241-248.

Mój wkład oceniam na 33 %. Polegał on na opracowaniu efektywnej czasowo metody wyznaczania czasu cyklu, współudziale w opracowaniu modelu grafowego, konstrukcji algorytmu oraz sformułowaniu wniosków i redakcji pracy.

4.3 Omówienie celu naukowego/artystycznego ww. pracy/prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania

Zestaw prac zebranych w ramach powiązanego tematycznie cyklu publikacji poświęcony jest harmonogramowaniu operacyjnemu w systemach wytwarzania, w których istotnym elementem są dodatkowe ograniczenia. Przedmiotem moich badań były systemy produkcyjne, w których stosuje się następujące ograniczenia:

1. brak buforów między stanowiskowych,
2. brak możliwości czekania pomiędzy wykonywaniem kolejnych operacji zadania,
3. brak możliwości przestoju maszyny,
4. wymóg wytwarzania cyklicznego.

Celem prowadzonych przeze mnie badań naukowych było opracowanie efektywnych obliczeniowo algorytmów optymalizacyjnych wspomagających harmonogramowanie operacyjne w przedsiębiorstwach, w których istotnym elementem są w/w ograniczenia. Ograniczenie te mogą wynikać z wymagań technologii wytwarzania i/lub redukcji kosztów wytwarzania osiąganych przez zmniejszenie liczby urządzeń pomocniczych (buforów, magazynów itd.) oraz zmniejszeniem okresu od momentu rozpoczęcia produkcji do momentu dystrybucji produktów gotowych.

Konkretnym efektem prac badawczych stanowiących cykl są szybkie algorytmy optymalizacyjne oparte na metodzie przeszukiwania z zabronieniami wyznaczające wysokiej

Pempera

jakości rozwiązania dla konkretnych problemów optymalizacyjnych o rozmiarach odpowiadających spotykanym w praktyce. Wspólną cechą algorytmów, odróżniającą je od istniejących, jest zastosowanie tzw. podejścia blokowego, które istotnie poprawia efektywność przeszukiwania przestrzeni rozwiązań przez eliminację z procesu przeszukiwań podprzestrzeni, w których znajdują się rozwiązania gorsze od już znalezionych oraz kierunkującą proces przeszukiwań do obszarów o dobrej jakości rozwiązań. Należy podkreślić, że podejście to zostało w istotnie rozwinięte/rozszerzenone na problemy z dodatkowymi ograniczeniami.

Najważniejszym wynikiem zamieszczonych prac jest:

- sformułowanie twierdzeń blokowych dla w/w ograniczeń ich wykorzystanie w konstrukcji efektywnych algorytmów wspomagających harmonogramowanie,
- opracowanie efektywnej czasowo metody wyznaczenia czasu cyklu bazującej na wyznaczeniu długości najdłuższych dróg w odpowiednio skonstruowanych grafie modelującym problem szeregowania zadań.

Opis osiągnięć prezentowanych w tej części autoreferatu składa się z wprowadzenia, opisu metody konstruowania algorytmów przeszukiwania z zabronieniami bazującego na podejściu blokowym, własności blokowych dla problemów z w/w ograniczeniami, metody wyznaczenia czasu cyklu w problemach szeregowania oraz algorytmów harmonogramowania zadań w cyklicznych systemach produkcyjnych.

4.3.1 Wprowadzenie

Współczesne przedsiębiorstwa produkcyjne wyposażone są w wielofunkcyjne elastyczne maszyny sterowane cyfrowo. Dzięki temu są w stanie produkować wiele różnych produktów korzystając z tego samego parku maszynowego. Spełnione jest zatem jedno z najistotniejszych założeń zachodzącej obecnie rewolucji przemysłowej 4.0 jakim jest produkcja dostosowana zarówno ilościowo jak i pod względem asortymentu do ciągle zmieniającego się zapotrzebowania konsumentów.

Z drugiej strony, konkurencja wolnorynkowa wymusza na przedsiębiorstwach ustawicznego zmniejszania kosztów produkcyjnych. Znana od lat strategia JIT zakłada redukcję kosztów między innymi poprzez zmniejszenie liczby i/lub pojemności buforów międzystanowiskowych, pojemności magazynu centralnego oraz redukcję czasu przepływu produktów przez system wytwarzania.

Harmonogramowanie produkcji w takich systemach jest dużym wyzwaniem najczęściej realizowanym w sposób przybliżony przez operatora procesu zwykle przy wsparciu akusza kalkulacyjnego. Ewentalne niedokładności/błedy w planowaniu są kompensowane poprzez wprowadzanie tzw. rezerw czasowych (buforów czasowych) w harmonogramie, które wynoszą do kilkunastu procent czasu pracy. Redukcja buforów czasowych jest potencjalną możliwością zwiększenia wydajności systemu produkcyjnego.

Najprostsze systemy produkcyjne składają się z jednej maszyny, natomiast systemy wielomaszynowe mogą mieć strukturę przepływową (najczęściej spotykane), gniazdową lub otwartą. W systemach przepływowych produkty przetwarzane są na wszystkich maszynach w tej samej kolejności, natomiast w systemach gniazdowych każdy produkt może być wykonywany na maszynach w innej kolejności. W systemie otwartym kolejność wykonywania operacji wewnątrz zadań może być modyfikowana. W rzeczywistych systemach

Pempera

produkcyjnych z reguły maszyny są zwielokrotnione mamy wówczas do czynienia z systemami z maszynami równoległymi.

Z punktu widzenie komputerowego wspomagania procesu harmonogramowania operacyjnego dużym wyzwaniem jest skonstruowanie efektywnych algorytmów, ponieważ już najprostsze systemy produkcyjne generują NP-trudne problemy optymalizacyjne. Ponadto liczba operacji wykonywanych w systemie produkcyjnym jest bardzo duża co z reguły wyklucza zastosowanie dokładnych algorytmów. Z tego powodu efektywne algorytmy wspomagające harmonogramowanie w systemach produkcyjnych bazują na inteligentnych metodach przeszukiwania przestrzeni rozwiązań.

Różnorodność modeli i algorytmów, nawet dla tego samego problemu praktycznego, jest całkowicie uzasadniona twierdzeniem „no free lunch" Wolpert'a i Macready'ego, z którego wynika m.in. niemożność skonstruowania jednego „uniwersalnego" algorytmu przybliżonego dobrej jakości dla dowolnych danych wejściowych.

W literaturze rozważane są przede wszystkim podstawowe (klasyczne) modele w/w systemów produkcyjnych. Cechują się one pominięciem istotnych ograniczeń występujących w rzeczywistych systemach produkcyjnych takich jak: brak buforów, ograniczone czekanie, produkcja cykliczna. Mimo iż istnieje dużo różnych algorytmów optymalizacyjnych dla klasycznych problemów szeregowania zadań, problemy z dodatkowymi ograniczeniami rzadko występują w literaturze. Powodem są trudności związane z modelowaniem nawet najprostszych ograniczeń oraz konstruowaniem efektywnych algorytmów dla problemów z w/w ograniczeniami algorytmów.

Wobec mnogości problemów generowanych przez praktykę, wyróżnianie szczególnych własności teoretycznych, których wbudowanie w algorytm rozwiązywania poprawia jego jakość i/lub czas działania jest całkowicie uzasadnione.

Należy podkreślić, że większość prac przedstawionych w cyklu publikacji są to jedne z pierwszych prac poświęconych tym problemom. Natomiast o dużym zainteresowaniu tą tematyką świadczy duża liczba cytowań kluczowych prac cyklu.

4.3.2 Metoda konstruowania algorytmów przeszukiwania z zabronieniami bazująca na podejściu blokowym

Systemy produkcyjne z reguły generują problemy optymalizacyjne, które możemy opisać formalnie w postaci zadania programowania całkowitoliczbowego i wyznaczyć rozwiązanie dokładne korzystając z istniejących pakietów oprogramowania. Pomijając trud budowy takiego modelu, dużym problemem jest czas związany z wyznaczeniem takiego rozwiązania. Dlatego, w praktyce pakiety takie stosuje się w przypadku produkcji masowej wyrobów o niewielkiej różnorodności asortymentowej.

Stosowane powszechnie metody konstruowania algorytmów dla problemów szeregowania zadań wykorzystują fakt, że większość tych problemów można sprowadzić do problemów kolejnościowych tj. takich w których zmienną decyzyjną jest kolejność wykonywania operacji. Dodatkowo, dla najistotniejszych funkcji kryterialnych harmonogram wykonywania operacji jest lewostronnie dosunięty na osi czasu.

W swoich pracach badawczych skupiłem się na konstruowaniu algorytmów bazujących na metodach przeszukiwania przestrzeni rozwiązań dla wspomnianych wcześniej kolejnościowych problemów szeregowania zadań z funkcjami celu:

- minimalizacja terminu zakończenia wszystkich zadań,

Pempera

- minimalizacja czasu cyklu w systemach cyklicznych.

W konstrukcji algorytmów optymalizacyjnych zastosowałem tzw. podejście blokowe, które jest rozpoznawalną w świecie cechą środowiska wrocławskiego zajmującego się optymalizacją dyskretną.

Metoda przeszukiwania z zabronieniami (ang. tabu search) jest jedną z najefektywniejszych metod konstruowania algorytmów optymalizacyjnych dla problemów optymalizacji dyskretnej, która została zaproponowana przez Glovera [5, 6]. Algorytmy bazujące na tej metodzie są heurystycznymi algorytmami iteracyjnymi. W każdej iteracji algorytmu generowany jest zbiór rozwiązań sąsiednich rozwiązanego zwany sąsiedztwem lub otoczeniem. Zbiór ten dzieli się na dwa podzbiory: (*i*) podzbiór rozwiązań niezabronionych, (*ii*) podzbiór rozwiązań zabronionych. Ze zbioru rozwiązań niezabronionych wyznacza się rozwiązanie najlepsze, które staje rozwiązaniem bieżącym w kolejnej iteracji algorytmu.

Algorytm kończy działanie po spełnieniu warunku zatrzymania, którym najczęściej jest wykonanie zadanej liczbie iteracji, zatrzymanie po zadanej liczbie iteracji bez poprawy najlepszego znalezionego rozwiązania, przekroczenie zadanego czasu obliczeń.

Mechanizm zabronień implementowany jest najczęściej w postaci listy o ograniczonej długości zawierającej rozwiązania bieżące lub ich atrybuty z kilku iteracji wstecz. Lista zabronień jest pewnego rodzaju krótkoterminową pamięcią obsługiwaną zgodnie z regułą FIFO tj. w przypadku gdy osiągnie maksymalną długość, dodanie nowego elementu listy poprzedzone jest usunięciem najstarszego.

Konstrukcja algorytmu opartego na tej metodzie dla konkretnego problemu optymalizacyjnego wymaga zdefiniowania: (*i*) postaci rozwiązania, (*ii*) postaci rozwiązania, (*iii*) mechanizmu zabronień.

Siłą algorytmu opartego na tej metodzie jest systematyczne przeglądanie całego sąsiedztwa rozwiązania bieżącego co niestety wymaga dużego czasu obliczeń. Czas przeszukiwania sąsiedztwa można istotnie zmniejszyć ograniczając otoczenie do podzbioru rozwiązań rokujących poprawę tj. rozwiązań co do których istnieją przesłanki wskazujące, że mogą one być lepsze od bieżącego.

W konstruowaniu algorytmów stosowałem jednolite podejście składające się z dwóch faz. Faza pierwsza polega na budowie modelu grafowego i składa się z trzech etapów:

1. sformułowanie słownego problemu optymalizacyjnego (specyfikacja),

2. opracowaniu układu nierówności dla zadanej kolejności wykonywania operacji,

3. budowy modelu grafowego.

Kluczowym elementem tej fazy jest punkt 2, ponieważ konstrukcja modelu grafowego oraz algorytmy wyznaczających harmonogram wynikają wprost z tego punktu.

Efektem drugiej fazy jest konstrukcja efektywnej czasowo procedury przeglądania przestrzeni rozwiązań, w której w szczególności stosowano:

1. własności problemu (podejście blokowe),

2. tzw. akcelerację obliczeń, która poprzez zaawansowaną dekompozycję i agregację obliczeń, skutkuje zmniejszeniem złożoności obliczeniowych odpowiednich procedur numerycznych,

Kurpel

3. strategie wyznaczania wartości funkcji celu dla wielu rozwiązań.

Należy zauważyć, że elementy wyżej opisanej metody stosowane są również przez innych autorów. W dalszej części sekcji przedstawione zostaną elementarne informacje dotyczące najistotniejszych punktów opisanego wyżej podejścia.

Opis w postaci układu nierówności

Harmonogram wykonywanania operacji w systemie produkcyjnym możemy opisać przy pomocy momentów rozpoczęcia i zakończenia wykonywania operacji. Oznaczmy, przez S_i (C_i) moment rozpoczęcia (zakończenia) wykonywania operacji $i \in O$, gdzie O jest zbiorem wszystkich operacji wykonywanych w systemie produkcyjnych. Zdecydowaną większość wymagań występujących w systemach produkcyjnych możemy sformułować w postaci: *moment rozpoczęcia pewnej operacji nie może być wcześniejszy od:*

1. *momentu zakończenia innej operacji,*
2. *momentu rozpoczęcia innej operacji,*
3. *innego zdarzenia w systemie.*

Wszystkie z wymienionych ograniczeń możemy formalnie zapisać w formie nierówności matematycznych. Dla przykładu, pierwsze ograniczenie możemy zapisać w postaci $S_j \geq C_i$, $j, i \in O$. Typowym ograniczeniem dla systemów produkcyjnych jest brak możliwości przerywania wykonywania operacji, zatem termin zakończenia zdeterminowany jest momentem rozpoczęcia operacji i wynosi $C_i = S_i + p_i$, gdzie p_i jest czasem wykonania operacji.

Ostatecznie, typowe wymagania słowne można opisać w postaci układu nierówności w postaci:

$$S_j - (S_i + p_i) \geq w_{ji}, \tag{1}$$

gdzie w_{ji} jest czasem zwłoki (wymaganym opóźnieniem) pomiędzy momentem S_j i momentem $(S_i + p_i)$.

Można pokazać, w podobny sposób jak w pracy [16], że rozwiązanie układu nierówności w postaci $S_j - (S_i + p_i) \geq w_{ij}$ można sprowadzić do problemu znajdowania najdłuższych dróg w grafie skierowanym, przy czym zwracane długości najdłuższych dróg odpowiadają najmniejszym z możliwych wartościom S_i.

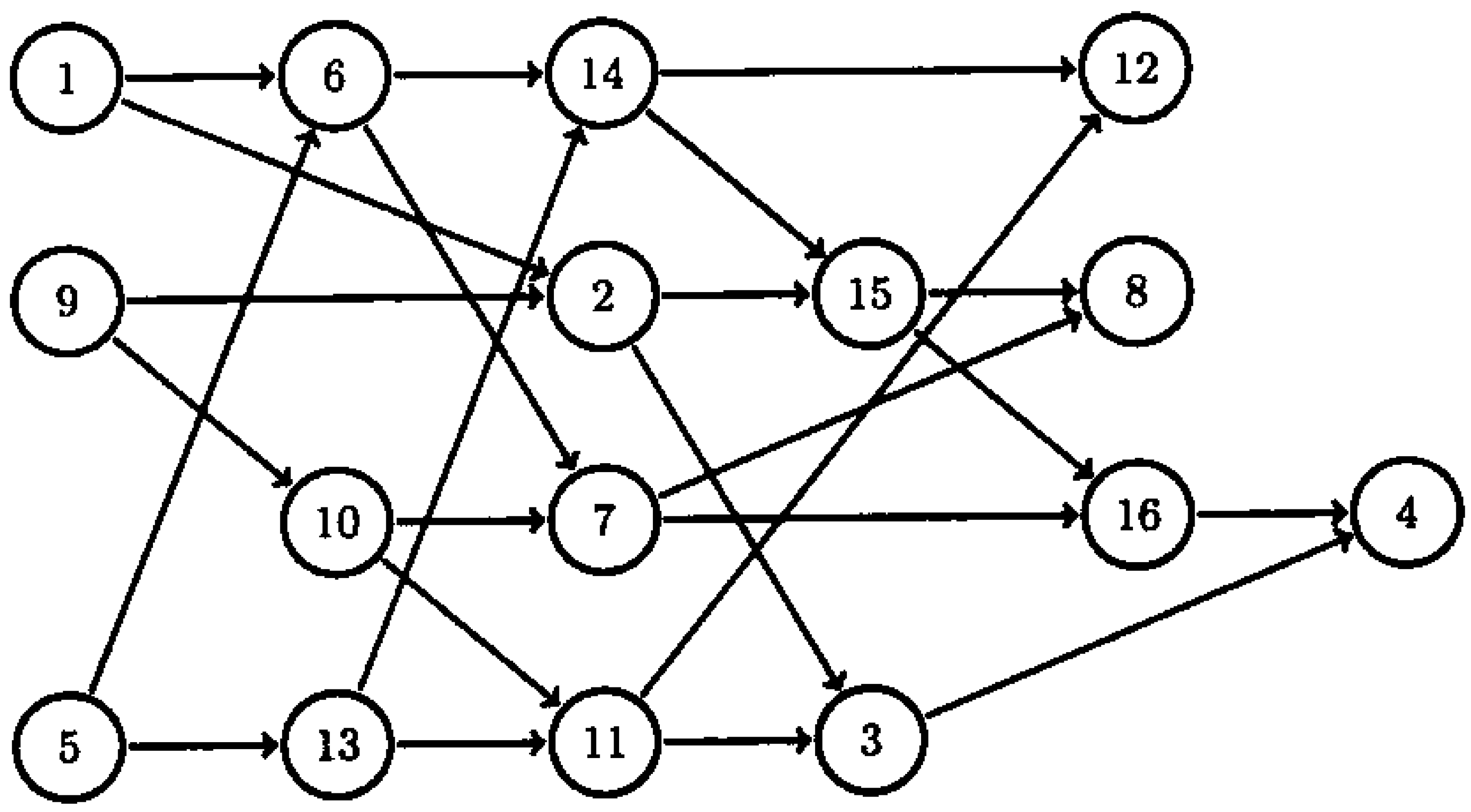
W celu zilustrowania osiągnięcia opisywanego etapu rozpatrzmy przykład konstrukcji układu nierówności dla problemu otwartego.

Przykład 1 *W otwartym systemie produkcyjnym składającym się z $m = 4$ maszyn należy wykonać $n = 4$ zadania. Każde zadanie wykonywane jest na każdej maszynie. W tabeli 1 przedstawiono:*

- *numery operacji przyporządkowanych do zadań,*
- *czasy wykonania operacji p_i,*
- *numery maszyn ν_i, na których wykonywane są te operacje.*

Tabela 1: Dane Przykładu 1

zadanie	i	p_i	ν_i	i	p_i	ν_i	i	p_i	ν_i	i	p_i	ν_i
1	1	9	1	2	8	2	3	9	4	4	8	3
2	5	5	4	6	2	1	7	6	3	8	4	2
3	9	8	2	10	6	3	11	3	4	12	3	1
4	13	7	4	14	4	1	15	2	2	16	4	3



Rysunek 1: Graf $G(\sigma, \pi)$ dla problemu otwartego

Załóżmy, że operacje wykonywane są na maszynach w kolejności

$$\pi = (\pi_1, ..., \pi_4) = ((1,6,14,12),(9,2,15,8),(10,7,16,4),(5,13,11,3)), \tag{2}$$

natomiast kolejność wykonywania operacji w ramach zadań jest następująca

$$\sigma = (\sigma_1, ..., \sigma_4) = ((1,2,4,3),(6,8,7,5),(12,9,10,11),(14,15,16,13)). \tag{3}$$

Ograniczenie słownie w postaci *„i-ta w kolejności π_k operacja wykonywana na maszynie k może rozpocząć się dopiero po zakończeniu wykonywania poprzedniej w tej kolejności operacji"* możemy zapisać w następującej formie matematycznej

$$S_{\pi_k(i)} \geq C_{\pi_k(i-1)}, \quad i = 2, ..., n, \ k = 1, ..., m, \tag{4}$$

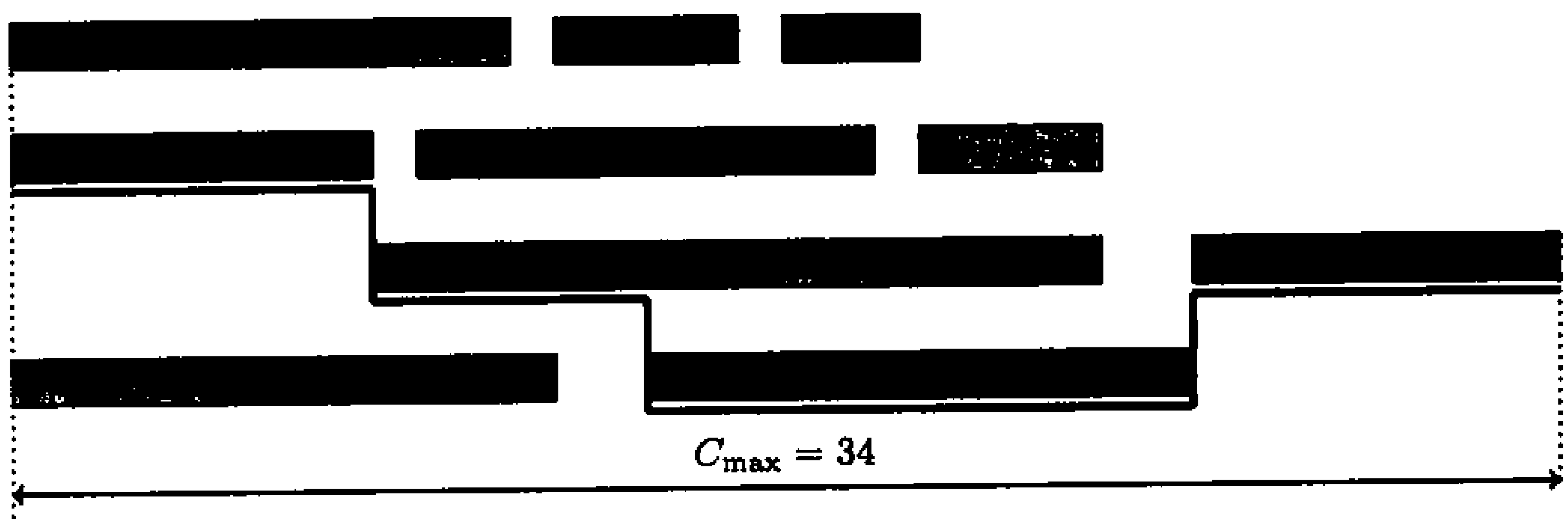
a następnie przekształcić do postaci układu nierówności

$$S_{\pi_k(i)} - (S_{\pi_k(i-1)} + p_{\pi_k(i-1)}) \geq 0, \quad i = 2, ..., n, \ k = 1, ..., m. \tag{5}$$

Po podobnych przekształceniach dla kolejności σ otrzymujemy:

$$S_{\sigma_j(i)} - (S_{\sigma_j(i-1)} + p_{\sigma_j(i-1)}) \geq 0, \quad i = 2, ..., m, \ j = 1, ..., n. \tag{6}$$

Pempera)



Rysunek 2: Harmonogram wykonywania operacji na maszynach

Model grafowy

Skierowany graf ważony $G = (V, E)$ odpowiadający układowi nierówności w postaci (1) składa się ze zbioru węzłów V oraz łuków E. Węzeł $v \in V$ odpowiada operacji $v \in O$ i jest obciążony wagą p_v, natomiast łuk $(i, j) \in E$ odpowiada ograniczeniu $S_j - (S_i + p_i) \geq w_{ij}$ i obciążony jest wagą w_{ij}.

Przeprowadzając podobny sposób dowodzenia jak w pracach [4] można udowodnić, że dla układu nierówności w postaci (1) zachodzi

Twierdzenie 1 *Układ nierówności w postaci (1) jest rozwiązywalny wtedy i tylko wtedy, gdy odpowiadający mu skierowany graf ważony nie ma cykli o dodatniej długości.*

Długości najdłuższych dróg w dowolnym grafie (jeżeli istnieją) można wyznaczyć odpowiednio zmodyfikowanym algorytmem Bellmana-Forda. Algorytm wyznacza długości najdłuższych dróg od wybranego węzła do wszystkich pozostałych węzłów wykonując co najwyżej $|V|$ iteracji, w których relaksowane są wszystkie łuki. Dla grafów pełnych o $O(|V|^2)$ łukach złożoność obliczeniowa algorytmu wynosi $O(|V|^3)$. W acyklicznych grafach skierowanych (ang. DAG) długość najdłuższych dróg możemy wyznaczyć w znacznie krótszym czasie tj $O(|E|)$ algorytmem DAG-LP. Należy, zauważyć, że zarówno algorytm Belmana-Forda jak i DAG-LP (ang. longest path in DAG) są ogólnie znanymi algorytmami grafowymi.

Rysunek 1 przedstawia graf $G(\sigma, \pi)$ skonstruowany dla kolejności wykonywania operacji na maszynach π, kolejności wykonywania operacji w ramach zadań σ oraz danych z Przykładu 1. Łuki poziome odpowiadają nierównościom (5), natomiast łuki ukośne nierównościom (6). Harmonogram wykonywania operacji wyznaczony algorytmem DAG-LP został zamieszczony na Rysunku 2.

Podejście blokowe

Podejście blokowe jest to oryginalny sposób konstruowania algorytmów optymalizacyjnych bazujących na metodach przeszukiwania przestrzeni rozwiązań dla problemów szeregowania, polegające na eliminacji w procesie przeszukiwań podprzestrzeni nie zawierający rozwiązań lepszych od już znanych. Kluczowym elementem tego podejścia są tzw. własności blokowe, których występowanie wskazano we wszystkich klasycznych modelach systemów produkcyjnych.

Początki prac nad podejściem blokowym sięgają wczesnych lat osiemdziesiątych (prace Grabowski i inni [7, 8]) i dotyczą problemów szeregowania zadań z kryterium minimalizacji czasu zakończenia wszystkich zadań ($C_{\max}$). Pierwsza praca, w której pojawia się pojęcie algorytmu opartego na metodzie blokowej dotyczy problemu jednomaszynowego z kryterium minimalizacji maksymalnej nieterminowości $L_{\max}$ [10]. Zastosowanie podejścia blokowego zostało następnie rozszerzone na problemy wielomaszynowe przepływowe i gniazdowe z funkcjami kryterialnymi minimalizacji maksymalnej nieterminowości $L_{\max}$ i maksymalnego przepływu $f_{\max}$ [9, 18]. Od tego czasu własności blokowe są wykorzystywane w wielu algorytmach autorów zagranicznych, takich jak [2, 1, 17].

Początkowo, własności eliminacyjne bloków zostały wykorzystane w konstrukcji algorytmów opartych na metodzie podziału i ograniczeń (B&B). Zastosowano w nich, unikalną regułę podziału zbioru rozwiązań bazującą na własnościach blokowych problemów szeregowania. Podział w drzewie przeszukiwań odbywa się w oparciu o pewne rozwiązanie nazywane reprezentantem zbioru. Własności blokowe określają warunki konieczne jakie muszą spełniać rozwiązania lepsze od danego, w związku z tym w algorytmie B&B opartym na podejściu blokowym nie są rozpatrywane podzbiory rozwiązań zawierające rozwiązania niespełniające warunków twierdzenia blokowego.

Pomimo istotnej redukcji liczby rozwiązań wynikających z własności blokowych, czas działania algorytmów dokładnych w przypadku problemów szeregowania zadań generowanych przez rzeczywiste systemy produkcyjne, czas ich działania jest zbyt długi i jest nie do zaakceptowania w informatycznych systemach wspomagających harmonogramowanie operacyjne. Uwaga ta również dotyczy współczesnych pakietów oprogramowania przeznaczonych do rozwiązywania problemów optymalizacji dyskretnej.

Spektakularny sukces algorytmu przeszukiwania z zabronieniami (TS) stosującego podejście blokowe [13, 14, 15] pokazał, że heurystyczne metody przeszukiwań lokalnych mogą wyznaczać rozwiązania o dobrej jakości w czasie akceptowalnym przez praktyków dla problemów o dużych rozmiarach.

Własności blokowe nierozerwalnie związane są z modelami grafowymi problemów szeregowania zadań. Niech $U_{v_1,v_k} = (v_1, \ldots, v_k)$ będzie najdłuższą drogą (ścieżką krytyczną) w grafie $G(\sigma, \pi)$ skonstruowany dla pary (σ, π) określających kolejność wykonywania operacji w otwartym problem szeregowania zadań.

Maksymalny podciąg zwarty (podciąg następujących po sobie węzłów) ścieżki krytycznej składający się z co najmniej dwóch węzłów reprezentujących operacje wykonywane na tej samej maszynie nazywamy **blokiem operacji**.

Blok rozpoczynający się w węźle v_s i kończący w węźle v_t będziemy oznaczali w następujący sposób

$$B_{st} = (v_s, v_{s+1}, \ldots, v_t). \tag{7}$$

W bloku $B_{st} = (v_s, v_{s+1}, \ldots, v_t)$ wyróżniamy pierwszą v_s i ostatnią operację v_t. Operacje pomiędzy pierwszą i ostatnią operacją z bloku $(v_{s+1} \ldots, v_{t-1})$ nazywamy operacjami wewnętrznymi. Bloki operacji połączone są ścieżkami przebiegającymi przez węzły reprezentujące operacje tego samego zadania.

Twierdzenie 2 *Niech π' będzie kolejnością wykonywania operacji na maszynach otrzymaną z π taką, że*

$$C_{\max}(\sigma, \pi') < C_{\max}(\sigma, \pi), \tag{8}$$

Pempera

wówczas co najmniej jedna operacja z co najmniej jednego bloku w π wykonywana jest przed pierwszą albo za ostatnią operacją z tego bloku.

Z twierdzenia 2 wynika, że warunkiem koniecznym otrzymania rozwiązania lepszego jest przestawienie co najmniej jednej operacji z co najmniej jednego bloku przed pierwszą albo ostatnią operację z tego bloku. Jest to warunek konieczny, niestety nie jest warunkiem wystarczającym. Jednakże warunki twierdzenia 2 pozwalają na eliminację rozwiązań nie lepszych od danego rozwiązania. Są to rozwiązania nie spełniające warunku koniecznego tego twierdzenia, w szczególności są to rozwiązania powstałe przez

- dowolną permutację operacji wewnętrznych bloku,
- dowolną permutację operacji nienależących do ścieżki krytycznej,
- kolejność powstałą przez wstawienie operacji nieznajdującej się na ścieżce krytycznej na dowolną pozycję pomiędzy pierwszą i ostatnią operacją dowolnego bloku.

Na Rysunku 2 zaznaczono pogrubioną linią przebieg ścieżki krytycznej. W ścieżce można wyróżnić jeden blok skladający się z dwóch operacji 11 oraz 3. Rozpatrzmy wszystkie możliwe kolejności powstałe przez przesunięcie każdej operacji na każdą możliwą pozycję na maszynie, na której jest wykonywana. W takim otoczeniu znajduje się $4 \cdot 3 \cdot 3 = 36$ różnych kolejności, z których tylko 3 spełniają warunki twierdzenia 2 tj. kolejności powstałe przez przesunięcie operacji 3 przed 11, 13 oraz 5.

Akceleracja obliczeń

Elementem istotnie przyspieszającym proces przeszukiwania przestrzeni rozwiązań może być akceleracja obliczeń. Ogólnie mówiąc, akceleracja polega na wykonywaniu obliczania wartości funkcji celu w takiej kolejności, aby częściowe wyniki obliczeń dla pewnych rozwiązań mogły być wykorzystane w obliczeniach funkcji celu dla innych rozwiązań.

Ogólnie mówiąc, akceleracja polega na przeglądaniu przestrzeni rozwiązań (wyznaczaniu wartości funkcji celu rozwiązań) w takie kolejności rozwiązań, aby część obliczeń wykonanych dla rozwiązań już przeglądniętych mogła być wykorzystana w obliczeniach funkcji celu dla kolejnych rozwiązań.

W konstrukcji akceleratorów wykorzystuje się fakt, że zarówno maksymalny czas zakończenia wykonywania, jak i czas cyklu równy jest długości najdłuższej drogi pomiędzy pewnymi węzłami w modelu grafowym. Rozwiązania sąsiednie rozpatruje się w takiej kolejności, aby graf odpowiadający kolejnemu rozwiązaniu tylko w niewielkim stopniu różnił się od grafu w poprzednim kroku. Wówczas długości najdłuższych dróg w dużych fragmentach grafu nie zmieniają się, zatem nie ma potrzeby ich wyznaczać po raz wtóry.

Strategie przeglądania przestrzeni rozwiązań

Kolejnym elementem skracającym czas przeszukiwania przestrzeni rozwiązń jest wybór strategii przeszukiwania. W przypadku konstruowanych algorytmów optymalizacyjnych stosowałem dwa podejścia:

1. mechanizmu polegającego na wykonaniu na rozwiązaniu bazowym kilku ruchów (zmian) - tzw. multiruchów,

Pempera)

2. wstępnego dolnego oszacowania wartości funkcji celu.

W drugim przypadku zakłada się, że wyznaczenie dolnego oszacowania jest znacznie mniej czasochłonne niż wyznaczenie wartości funkcji celu. W tej metodzie wstępnie szacuje się wartość funkcji celu dla każdego rozwiązania a następnie wyznacza się dokładnie wartość funkcji celu rozwiązań w kolejności od rozwiązania o najmniejszej do rozwiązania o największej wartości dolnego oszacowania. Łatwo można zauważyć, dokładne wyznaczanie wartości funkcji celu można przerwać po znalezieniu pierwszego rozwiązania o dokładnej wartości funkcji celu nie większej od wartości dolnego oszacowania pozostałych rozwiązań.

4.3.3 Problem przepływowy z ograniczeniami

Problem harmonogramowania zadań w przepływowym systemie produkcyjnym można sformułować następująco

W systemie produkcyjnym składającym się z m maszyn ze zbioru $M = \{1, 2, \ldots, m\}$ należy wykonać n zadań ze zbioru $J = \{1, 2, \ldots, n\}$. Zadanie $j \in J$ składa się z ciągu m operacji z $(O_{j1}, O_{j2}, \ldots, O_{jm})$. Operacja O_{jk} odpowiada wykonywaniu zadania j na maszynie k przez nieprzerywalny czas p_{jk}. Zadania wykonywane są na maszynach w identycznej kolejności.

Należy wyznaczyć taką kolejność wykonywania zadań na maszynach, dla której czas wykonywania zadań jest najkrótszy. Dla zadanej kolejności, harmonogram wykonywania zadań musi spełniać następujące ograniczenia:

1. moment rozpoczęcia wykonywania zadania na maszynie nie może być wcześniejszy niż moment zakończenia tego zadania na poprzedniej maszynie,
2. moment rozpoczęcia wykonywania zadania na maszynie nie może być wcześniejszy niż moment zakończenia poprzedniego zadania wykonywanego na tej samej maszynie.

Oznaczmy przez S_{jk} (C_{jk}) moment rozpoczęcia (zakończenia) wykonywania zadania j na maszynie k. Wówczas, dla kolejności wykonywania zadań $\pi = (\pi(1), \ldots, \pi(n))$ ograniczenie (1) możemy zapisać w postaci $S_{j,k} \geq C_{j,k-1}$, natomiast ograniczenie (2) w postaci $S_{\pi(j),k} \geq C_{\pi(j-1),k}$. Po podstawieniu $C_{j,k} = S_{j,k} + p_{j,k}$ oraz po prostych przekształceniach otrzymujemy układ nierówności:

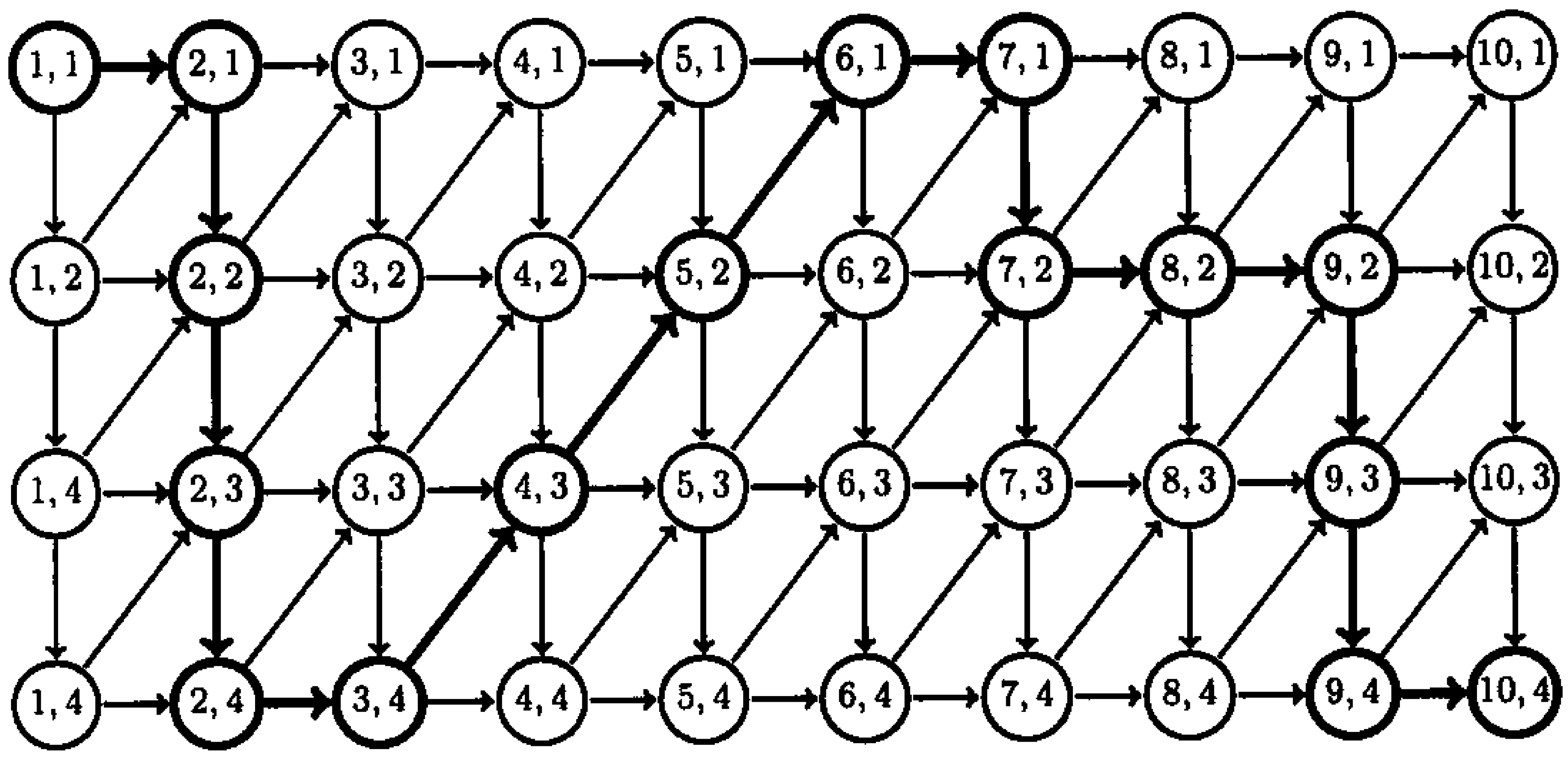
$$S_{j,k} - (S_{j,k-1} + p_{j,k-1}) \geq 0, \quad j \in J, \ k \in M, \tag{9}$$

$$S_{\pi(i),k} - (S_{\pi(i-1),k} + p_{\pi(i-1),k}) \geq 0, \quad i = 2, \ldots, n, \ k \in M. \tag{10}$$

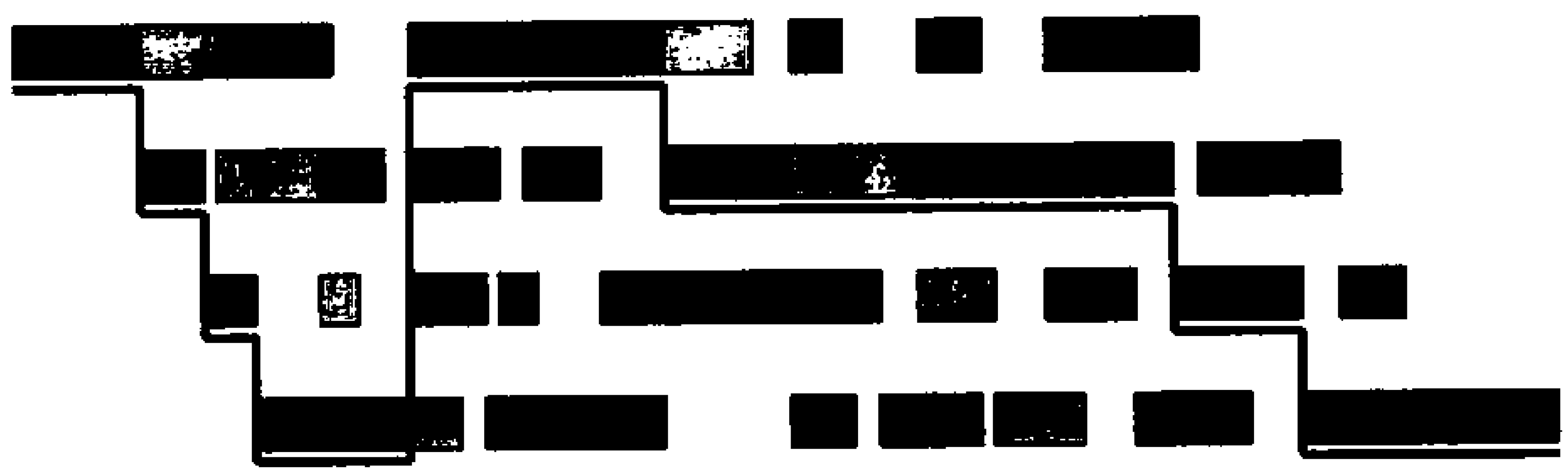
Brak buforów pomiędzy maszynami

Brak buforów pomiędzy maszynami oznacza, że zakończone zadanie wykonywane na pewnej maszynie blokuje ją do momentu, w którym może być przetwarzane na następnej maszynie. Dokładniej, moment rozpoczęcia wykonywania zadania $\pi(j)$ na maszynie k nie może być wcześniejszy od momentu rozpoczęcia wykonywania poprzedniego zadania na następnej maszynie. Wymaganie to możemy zapisać w postaci układu nierówności w postaci

Pempera



Rysunek 3: Graf $G(\pi)$ dla problemu przepływowego z ograniczeniami bez magazynowania



Rysunek 4: Harmonogram wykonywania operacji na maszynach

$$S_{\pi(i),k} - (S_{\pi(i-1),k+1} + p_{\pi(i-1),k+1}) \geq -p_{\pi(i-1),k+1}, \quad i = 2, ..., n, \ k = 1, ..., m-1. \quad (11)$$

Dla układu nierówności (9-11) możemy skonstruować graf skierowany $G(\pi)$, składający się z węzłów reprezentujących operacje oraz trzech podzbiorów łuków odpowiadających ograniczeniom (9-11). Na Rysunku 3 przedstawiono graf $G(\pi)$ skonstruowany dla $n = 10$ oraz $m = 4$. Łuki poziome reprezentują ograniczenia (9), łuki poziome (10), natomiast łuki skośne ograniczenie (11). Łuki poziome i pionowe obciążone są wagą 0, natomiast łuk skośny $((\pi(j-1), k+1), (\pi(j), k))$ wagą $-p_{\pi(j-1),k+1}$. Pogrubioną linią zaznaczono ścieżkę krytyczną. Fragmenty ścieżki krytycznej połączone łukami poziomymi tworzą klasyczne bloki operacji.

Na Rysunku 4 przedstawiono harmonogram wykonywania zadań w systemie przepływowym bez buforów pomiędzy stanowiskami. Pogrubioną linią zaznaczono przebieg ścieżki krytycznej. Efekt blokowania maszyn można zaobserwować w przypadku 3, 4 i 5 zadania, które są blokowane odpowiednio na 3, 2 i 1 maszynie.

Najważniejszym wynikiem moich prac badawczych dotyczących systemów z brakiem buforów jest pokazanie, że fragmenty ścieżki krytycznej połączone łukami skośnymi (reprezentującymi ograniczenia wynikające z braku buforów) posiadają takie same cechy jak klasyczne bloki operacji. Ponadto wykazałem, że w systemach przepływowych produkcyjnych bez buforów kolejność wykonywania zadań na maszynach musi być identyczna na wszystkich maszynach oraz że graf $G(\pi)$ skonstruowany dla kolejności π jest DAG'giem w związku z tym harmonogram można wyznaczyć szybkim algorytmem DAG-LP.

Najistotniejsza praca dotycząca rozważanym ograniczeniom [C1] dedykowania jest problemowi przepływowemu z brakiem buforów. Własności eliminacyjne klasycznych bloków i tzw. antybloków (bloków skośnych) wykorzystałem w konstrukcji algorytmu optymalizacyjnego opartego na metodzie przeszukiwania z zabronieniami. W celu skrócenia czasu przeszukwania przestrzeni rozwiązań skonstruowałem akcelerator, który skrócił czas obliczeń blisko n-krotnie. Dodatkowo zbieżność procesu przeszukiwań zwiększyłem przez zastosowanie tzw. multiruchów. Mechanizm ten pozwolił na osiągnięcie dużych popraw szczególnie w pierwszych iteracjach algorytmu. Z badań eksperymentalnych algorytmu, które niedawno przeprowadziłem na współczesnych komputerach klasy PC wynika, że po zastosowaniu pełnego otoczenia wynikającego z metody blokowej jakość dostarczanych rozwiązań porównywalna jest z jakością rozwiązań generowanych przez najlepsze algorytmy spotykane w literaturze.

Ograniczenie bez czekania

Ograniczenie bez czekania oznacza, że rozpoczęcie wykonywania zadania na maszynie musi się rozpocząć niezwłocznie po zakończeniu wykonywania na poprzedniej maszynie. Można to zapisać w postaci równości $S_{j,k} = C_{j,k-1}$ co jest równoważne spełnieniu dwóch nierówności $S_{jk} \geq C_{j,k-1}$ oraz $C_{j,k-1} \geq S_{j,k}$. Stosując podobne przekształcenia, jak w przypadku przedstawionym powyżej otrzymujemy układ nierówności:

$$S_{j,k} - (S_{j,k-1} + p_{j,k-1}) \geq 0, \quad j \in J, \ k = 2, ..., m, \tag{12}$$

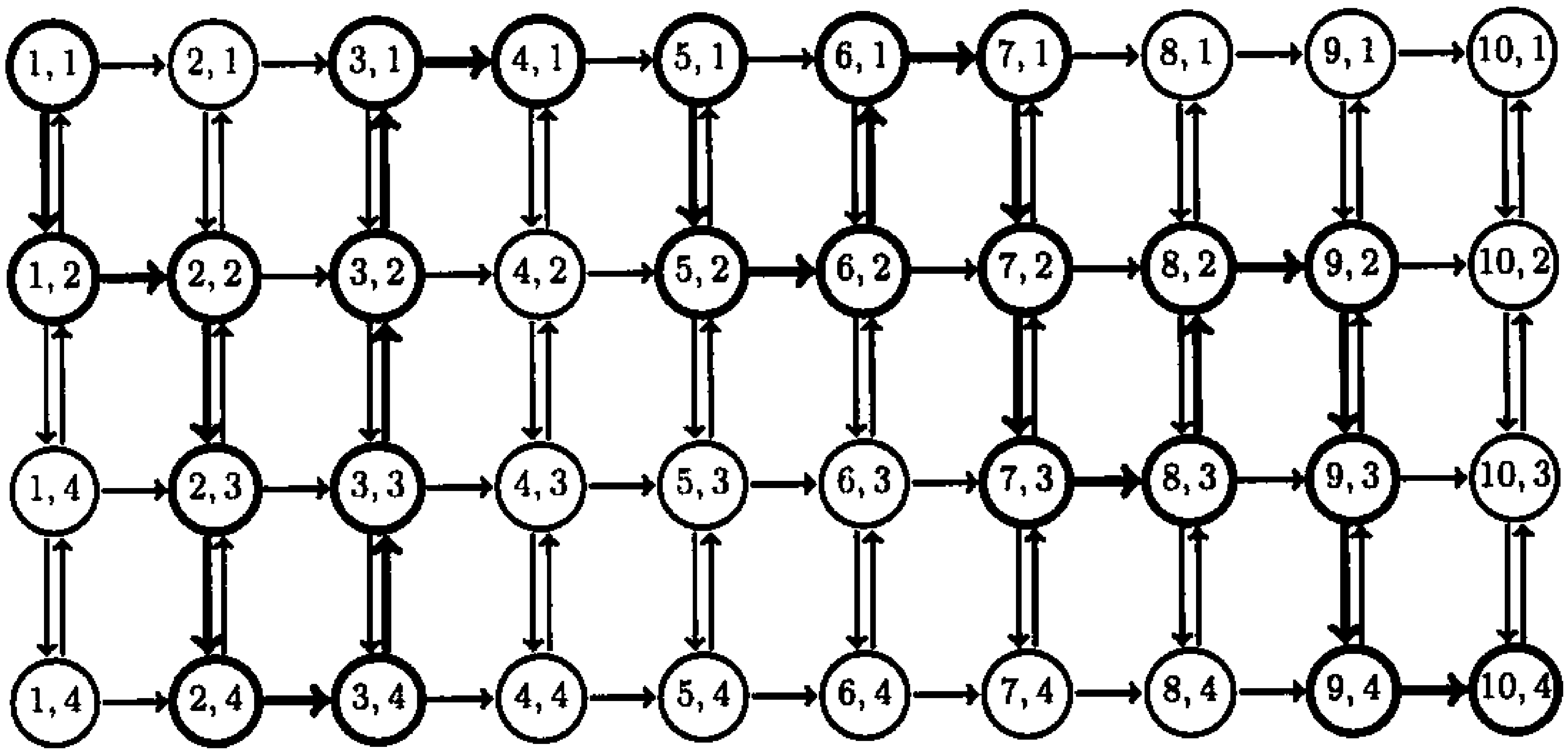
$$S_{j,k-1} - (S_{j,k} + p_{j,k}) \geq -p_{j,k} - p_{j,k-1}, \quad j \in J, \ k = 2, ..., m. \tag{13}$$

Łatwo można zauważyć, że ograniczenia (12) są tożsame z ograniczeniami (9) zatem mogą być pominięte. Układ nierówności (9–10) oraz (13) zdefiniowany dla kolejności wykonywania zadań π generuje graf skierowany $G(\pi)$. Struktura grafu różni się od poprzedniej tylko jednym podzbiorem łuków, tj. łuki odpowiadające nierównościom (11) zostały zastąpione łukami odpowiadającymi nierównościom (13). W grafie $G(\pi)$ zilustrowanym na Rysunku 5 łuki skośne zostały zastąpione łukami pionowymi skierowanymi w górę. Łuk $((\pi(j), k-1), (\pi(j), k))$ obciążony jest wagą $-p_{\pi(j),k-1} - p_{\pi(j),k}$. W tym wypadku graf $G(\pi)$ jest grafem cyklicznym zatem wyznaczenie harmonogramu wymaga użycia czasochłonnego algorytmu Bellmana - Forda ($O(|V|^3)$).

Najważniejszym wynikiem moich prac badawczych dotyczących problemów szeregowania zadań z ograniczeniami bez czekania jest opracowanie sposobu modelowania grafowego problemów z takimi ograniczeniami oraz opracowanie efektywnych metod wyznaczania harmonogramu w oparciu o algorytmy grafowe.

Praca [C3] dotyczy harmonogramowania zadań w cyklicznym problemie przepływowym z pomijaniem maszyn i wymaganiami bez czekania. Przedstawiony jest w niej, opracowany przeze mnie, algorytm wyznaczania długość najdłuższych dróg w grafie modelującym problem o złożoności obliczeniowej $O(nm)$. Jest on blisko n^2m^2 razy szybszy od

Pempera



Rysunek 5: Graf $G(\pi)$ dla problemu przepływowego z ograniczeniami bez czekania

algorytmu Belmana-Forda, który standardowo wykorzystuje się w przypadku grafów cyklicznych, do których należy graf modelujący problem.

Najistotniejsza praca dotyczy problemu przepływowego z ograniczeniem bez czekania [C2]. Dla tego problemu skonstruowałem algorytm optymalizacyjny oparty na metodzie przeszukiwania z zabronieniami, w którym zastosowałem sąsiedztwo składające się z tzw. multiruchów tj kilku ruchów wykonywanych na rozwiązaniu jednocześnie. W oparciu o własności wynikające z ograniczeń bez czekania sformułowałem warunki niezależności ruchów (wykonanie niezależnych ruchów gwarantuje sumowanie się popraw uzyskanych przez każdy z ruchów z osobna). W celu przyspieszenia procesu przeszukiwań skonstruowałem akcelerator wyznazający wartość funkcji celu w zamortyzowanym (w przeliczeniu na jedno rozwiązanie) czasie $O(1)$.

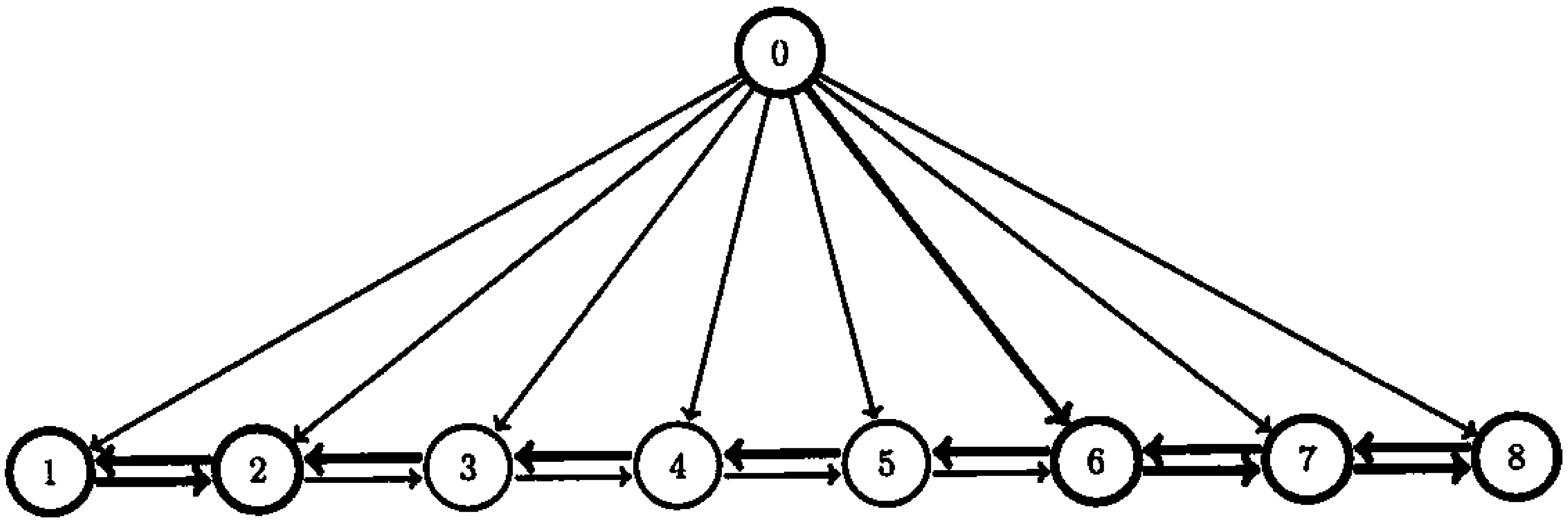
4.3.4 Problem jednomaszynowy z brakiem przestoju maszyny

W jednomaszynowym problemie szeregowania zadań produkcyjnych należy wykonać n zadań ze zbioru $J = \{1, \ldots, n\}$. Każde zadanie $j \in J$ ma określony termin dostępności r_j oraz czas dostawy q_j. Ograniczenie *brak przestoju maszyny* oznacza, że począwszy od pierwszego zadania maszyna musi wykonać kolejne zadania bez przerw (bezpośrednio jedno po drugim) aż do zakończenia wykonywania ostatniego zadania. Harmonogram wykonywania zadań, dla ustalonej kolejności wykonywania π jest dopuszczalny, jeżeli są spełnione następujące ograniczenia

1. moment rozpoczęcia zadania $j \in J$ nie jest wcześniejszy niż moment dostępności zadania r_j,

2. moment rozpoczęcia i-tego w kolejności π zadania $\pi(i)$ nie jest wcześniejszy niż moment zakończenia poprzedniego zadania, tj. zadania $\pi(i-1)$,

oraz ze względu na brak możliwości przestoju maszyny



Rysunek 6: $I-$blok skladajacy sie z ciagu operacji (6,7,8,1,2)

3. moment rozpoczęcia i-tego w kolejności π zadania $\pi(i)$ musi być dokładnie równy momentowi zakończenia poprzedniego zadania, tj. $\pi(i-1)$,

Dla ustalonej kolejności wykonywania π zadań na maszynie, momenty rozpoczęcia wykonywania zadań S_j, $j \in J$ możemy wyznaczyć rozwiązując następujący układ nierówności

$$S_j - (S_0 + 0) \geq r_j, \quad j \in J, \tag{14}$$

$$S_{\pi(i)} - (S_{\pi(i-1)} + p_{\pi(i-1)}) \geq 0, \quad i = 2, \ldots, n, \tag{15}$$

$$S_{\pi(i)} - (S_{\pi(i-1)} + p_{\pi(i-1)}) \geq 0, \quad i = 2, \ldots, n, \tag{16}$$

$$S_{\pi(i-1)} - (S_{\pi(i)} + p_{\pi(i)}) \geq -p_{\pi(i)}) - p_{\pi(i-1)}), \quad i = 2, \ldots, n, \tag{17}$$

Nierówność (14) wymaga dodania fikcyjnego węzła (nie reprezentującego żadnego z zadań) o wadze 0 oraz n łuków $(0, j)$, $j \in J$ obciążonych wagami r_i, natomiast nierówności (16) i (17) odpowiadają ograniczeniu (3). Graf $G(\pi)$ odpowiadający układowi nierówności (16)–(17) został przedstawiony na Rysunku 6.

Celem optymalizacji jest wyznaczenie harmonogramu spełniającego w/w wymagania, który minimalizuje maksymalny z momentów dostarczenia zadań. W pracu [C4] pokazano, że harmonogram wykonywania zadań można wyznaczyć w czasie $O(n)$ co nie jest oczywiste, ponieważ jak można zauważyć graf $G(\pi)$ jest grafem zawierającym cykle. Złożoność obliczeniową $O(n)$ otrzymano dzięki wykonywaniu relaksacji łuków w odpowiedniej kolejności.

Najważniejszym wynikiem przedstawionym w pracy [C4] jest zdefiniowanie tzw. i-bloków, tj. fragmentów ścieżki krytycznej zawierających łuki modelujące ograniczenie bez przestoju, które mają takie same własności jak klasyczne bloki. Konkretnym efektem prac badawczych jest algorytm oparty na metodzie B&B, w którym wykorzystano własności blokowe w rozgałęzieniach i cięciach metody. Testy komputerowe algorytmu wykazały, że jest on w stanie rozwiązywać w sposób dokładny problemy składające się z dziesiątek zadań w czasie nie przekraczającym 1s.

Pempera

4.3.5 Akcelerator przeszukiwania sąsiedztwa dla elastycznego problemu przepływowego

Elastyczne systemy przepływowe są najczęściej spotykanymi w praktyce systemami produkcyjnymi. Duża liczba operacji wykonywanych w systemie w połączeniu z elastycznością marszrut (duża liczba alternatywnych maszyn na każdym stanowisku) stanowi duże wyzwanie pod względem czasu przeszukiwania otoczenia dla algorytmów bazujących na metodach przeszukiwania przestrzeni rozwiązań. W pracy [C5] zdefiniowano sposób generowania otoczenia bazujący na podejściu blokowym oraz przedstawiono metodę pozwalającą na istotne przyspieszenie procesu przeszukiwania. Składa się ona z dwóch elementów:

- akceleratora bazującego na odpowiednio skonstruowanym grafie,
- przetwarzania równoległego dedykowanego na procesory wektorowe.

Najważniejszym moim wynikiem było opracowanie akceleratora obliczenia wartości funkcji celu dla wielu rozwiązań, który umożliwił wyznaczenie wartości funkcji celu w zamortyzowanym czasie $O(1)$.

Efektem końcowym badań jest algorytm oparty na metodzie przeszukiwania z zabronieniami wykorzystujący podejście blokowe oraz akcelerację procesu przeglądania sąsiedztwa. Z przeprowadzonych przeze mnie badań eksperymentalnych wynika, że na komputerach klasy PC, dzięki mechanizmom przetwarzania równoległego zimplementowanym we współczesnych procesorach, przeszukiwanie można przyspieszyć przeszło 5-cio krotnie.

4.3.6 Efektywna metoda wyznaczania czasu cyklu w problemie otwartym

W wielu rzeczywistych systemach produkcyjnych stosowana jest cykliczna strategia wytwarzania. Polega ona na produkcji ustalonej mieszanki wyrobów wielokrotnie w tak zwanych cyklach. W związku z tym plany kolejnych cykli można utworzyć przez przesunięcie o odpowiednią wielokrotność okresu zwanego czasem cyklu. Znacząco to upraszcza zarówno planowanie produkcji jak i planowanie działań pomocniczych takich jak logistyka produkcji czy zarządzanie łańcuchem dostaw, itd.

Niestety, wyznaczenie wydajnych harmonogramów cyklicznych (minimalizujących czas cyklu) w systemach produkcyjnych stanowi nadal duże wyzwanie dla naukowców. W znaczącej liczbie prac, harmonogramy cykliczne generuje się poprzez powielenie optymalnego harmonogramu wykonywania zadań dla kryterium minimalizacji czasu zakończenia wykonywania wszystkich zadań. Można pokazać, że w ten sposób generowane harmonogramy da się istotnie poprawić.

Powolny rozwój algorytmów optymalizacyjnych dla problemów cyklicznych związany jest z brakiem efektywnych metod wyznaczania czasu cyklu dla zadanej kolejności wykonywania zadań. W literaturze proponowane są algorytmy oparte na programowaniu całkowitoliczbowym [3], wyznaczeniu maksymalnego przepływu w odpowiednio skonstruowanym problemie przepływu w sieciach [12]. Najbardziej zaawansowane i uniwersalne podejście zaproponowane zostało przez Howard'a [11].

W ramach prac badawczych dotyczących wyznaczenia czasu cyklu w systemach produkcyjnych opracowałem nową metodę wyznaczania minimalnego czasu cyklu (krótko czasu cyklu) dla zadanej kolejności wykonywania operacji, której zakres stosowalności

Rempen

jest systematycznie powiększany na kolejne problemy szeregowania zadań. Opracowana metoda bazuje na modelu grafowym. Dla zadanej kolejności wykonywania operacji sprowadza się ona do konstrukcji specyficznego skierowanego grafu ważonego oraz wyznaczeniu długości odpowiednich najdłuższych dróg w tym grafie. Opis metody przedstawię na przykładzie problemu otwartego.

Opis otwartego cyklicznego problemu szeregowania zadań

System produkcyjny składa się z m maszyn ze zbioru $\mathcal{M} = \{1, \ldots, m\}$. W długoterminowym planie produkcji mamy wykonać wielokrotne zbiór zadań $\mathcal{J} = \{1, 2, \ldots, n\}$. Zadanie $j \in \mathcal{J}$ składa się z n_j operacji, które mogą być przetwarzane na maszynach w dowolnej kolejności. Oznaczamy przez $\mathcal{O} = \{1, \ldots, o\}$, $o = \sum_{j=1}^{n} n_j$ zbiór wszystkich operacji wykonywanych w systemie.

Operacje zadania $j \in \mathcal{J}$ są indeksowane przez kolejne liczby naturalne i tworzą zbiór $\mathcal{O}_j = \{l_j + 1, \ldots, l_j + n_j\}$, gdzie $l_j = \sum_{s=1}^{j-1} n_s$. Operacja $i \in \mathcal{O}$ jest przetwarzana na maszynie $\mu_i \in \mathcal{M}$ w czasie $p_i > 0$. W dowolnym momencie maszyna może przetwarzać co najwyżej jedno zadanie, oraz w dowolnym momencie można przetwarzać co najwyżej jedną operację tego samego zadania.

Niech $\pi_k = (\pi_k(1), \pi_k(2), \ldots, \pi_k(m_k))$ będzie permutacją elementów zbioru $\mathcal{O}^k = \{i \in \mathcal{O} : \nu_i = k\}$ określającą kolejność operacji na maszynie $k \in \mathcal{M}$. Kolejność wykonywania operacji na wszystkich maszynach jest określona przez ciąg permutacji $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$. W podobny sposób definiujemy kolejność wykonywania operacji w ramach zadań $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, gdzie $\sigma_j = (\sigma_j(1), \sigma_j(2), \ldots, \sigma_j(n_j))$ jest permutacją elementów zbioru $\mathcal{O}_j$.

Oznaczmy przez $S_i^{(x)}$ moment rozpoczęcia wykonywania operacji i w x-tym cyklu. Dla pary (π, σ) określającej kolejność wykonywania operacji na maszynach i kolejność wykonywania operacji w ramach zadań dla każdego cyklu x można zdefiniować następujący układ nierówności:

$$S_{\sigma_j(i)}^{(x)} - (S_{\sigma_j(i-1)}^{(x)} + p_{\sigma_j(i-1)}) \geq 0, \quad i = 2, \ldots, n_j, j \in \mathcal{J},\ x = 1, \ldots, \tag{18}$$

$$S_{\pi_k(i)}^{(x)} - (S_{\pi_k(i-1)}^{(x)} + p_{\pi_k(i-1)}) \geq 0, \quad i = 2, \ldots, n_k,\ k \in \mathcal{M},\ x = 1, \ldots. \tag{19}$$

Dodatkowo, operacje wykonywane jako pierwsze na maszynach w danym cyklu mogą rozpocząć się dopiero po zakończeniu ostatnich operacji z cyklu poprzedniego co możemy zapisać w postaci:

$$S_{\pi_k(1)}^{(x)} - (S_{\pi_k(n_k)}^{(x-1)} + p_{\pi_k(n_k)}) \geq 0, \quad i = 2, \ldots, n_j,\ j \in \mathcal{J},\ x = 2, \ldots. \tag{20}$$

Kostrukcja grafu

Dla kolejnych r cykli produkcyjnych, na podstawie układu nierówności (18–20) możemy skonstruować graf $G(\pi, \sigma)$

$$G(\pi, \sigma) = (V, D(\sigma) \cup E(\pi) \cup F(\pi)), \tag{21}$$

Pempen)

składający się za zbioru węzłów

$$V = \bigcup_{x=1}^{r} V^{(x)}, \quad V^{(x)} = \{i^{(x)} : i \in \mathcal{O}\}, \tag{22}$$

węzeł $i^{(x)} \in V$ reprezentuje operację $i \in \mathcal{O}$ wykonywaną w x-tym cyklu (waga tego węzła wynosi p_i) oraz trzech zbiorów łuków

$$D(\sigma) \cup E(\pi) \cup F(\pi) \subseteq V \times V, \tag{23}$$

odpowiadających kolejnym ograniczeniom

$$D(\sigma) = \bigcup_{x=1}^{r} D^{(x)}(\sigma), \quad D^{(x)}(\sigma) = \bigcup_{j=1}^{n} \bigcup_{i=2}^{n_j} \{(\sigma_j(i-1)^{(x)}, \sigma_j(i)^{(x)})\}, \tag{24}$$

$$E(\pi) = \bigcup_{x=1}^{r} E^{(x)}(\pi), \quad E^{(x)}(\pi) = \bigcup_{k=1}^{m} \bigcup_{i=2}^{m_k} \{(\pi_k(i-1)^{(x)}, \pi_k(i)^{(x)})\}, \tag{25}$$

$$F(\pi) = \bigcup_{x=2}^{r} F^{(x)}(\pi), \quad F^{(x)}(\pi) = \bigcup_{k=1}^{m} \{(\pi_k(m_k)^{(x-1)}, \pi_k(1)^{(x)})\}. \tag{26}$$

Wyznaczenie czasu cyklu

W pracy [C6] pokazałem, że w problemie otwartym, dla dopuszczalnej pary kolejności wykonywania operacji (σ, π), czas cyklu nie może być mniejszy od następującego dolnego oszacowania

$$L(\pi, \sigma) = \max_{2 \le y \le r} \max_{1 \le k \le m} \{\frac{L_{s^{(1)}, s^{(y)}}}{y-1} : \; L_{s^{(1)}, s^{(y)}} < \infty\}|_{s=\pi_k(1)}, \tag{27}$$

gdzie $L_{s^{(1)}, s^{(y)}}$ jest długością najdłuższej drogi w grafie $G(\sigma, \pi)$ z węzła reprezentującego operację $s^{(1)}$ do węzła reprezentującego operację $s^{(y)}$, natomiast $r = m + 1$. Ponadto w pracy [C6] udowodniono, że można skonstruować harmonogram cykliczny o czasie cyklu $T = L(\pi, \sigma)$ wyznaczonym 27.

Dla dopuszczalnej pary (σ, π), graf $G(\sigma, \pi)$ jest grafem acyklicznym, zatem dla ustalonego k wartości $L_{(\pi_k(1))^{(1)}, (\pi_k(1))^{(y)}}$, $y = 2, 2+1 \ldots, m+1$ można wyznaczyć w czasie $O(mo)$. Ostatecznie złożoność obliczeniowa wyznaczania czasu cyklu wynosi $O(m^2 o)$.

Należy podkreślić, że opracowana metoda może być zastosowana do wyznaczania czasu cyklu dla problemów takich jak: gniazdowy, przepływowy oraz ich wariantów z maszynami równoległymi.

Złożoność obliczeniowa algorytmu jest znacznie mniejsza od nielicznych algorytmów opisanych w literaturze wyznaczających czas cyklu dla problemów szeregowania zadań. Dla przykładu w pracy [12] rozpatrywany jest cykliczny problem przepływowy z ograniczeniem bez magazynowania. Wyznaczenie czasu cyklu, dla ustalonej kolejności n zadań wykonywanych na m maszynach, sprowadza się tam do wyznaczenia minimalnego przepływu w odpowiednio skonstruowanej sieci. Biorąc pod uwagę fakt, że graf składa się z mn węzłów oraz $O(mn)$ łuków, otrzymujemy złożoność wyznaczenia minimalnego przepływu $O(n^3 m^3)$, zatem czas obliczeń jest rzędu n^2 razy większy od czasu wynikającego z proponowanej metody, który wynosi $O(nm^3)$.

Pempera

Własności problemu

Innymi istotnymi moimi wynikami dotyczącymi omawianego problemu są:

1. określenie warunków, jakie musi spełniać para (σ, π) określająca kolejność wykonywania operacji w systemie tak, aby można było skonstruować realizowalny harmonogram cykliczny,

2. sformułowanie własności blokowych dla cyklicznego problemu otwartego.

W pierwszym przypadku, sprawdzenie dopuszczalności sprowadziłem do problemu badania cykliczności w grafie ze zbiorem węzłów $V^{(1)}$ i łuków $D^{(1)}$ (złożoność sprawdzenia acykliczności grafu wynosi $O(o)$). Natomiast, w drugim przypadku sformułowałem pojęcie bloku maszynowego oraz bloku zadaniowego, a następnie sformułowałem twierdzenia blokowe dla obu typów bloków.

Przedstawione własności problemu oraz metodę wyznaczania czasu cyklu zaimplementowałem w algorytmie opartym na metodzie przeszukiwania z zabronieniami. Z badań eksperymentalnych algorytmu, które przeprowadziłem na literaturowych instancjach testowych wynika, że zdecydowanej liczbie instancji algorytm wyznaczył rozwiązanie dokładne w czasie nie przekraczającym kilku minut w przypadku instancji o największych rozmiarach.

4.3.7 Algorytmy optymalizacyjne dla innych problemów cyklicznych

Przedstawiona w poprzedniej sekcji metoda konstruowania efektywnych algorytmów optymalizacyjnych dla problemów cyklicznych jest kosenkwencją wcześniejszych prac dotyczących rozwiązywania tych problemów.

W pracy [C7] przedstawiono równoległy algorytm oparty na metodzie przeszukiwania z zabronieniami dla cyklicznego problemu gniazdowego. Atutem tego algorytmu jest zrównoleglenie procesu wyznaczania czasu cyklu dzięki czemu został istotnie zmniejszony czas przeszukiwania otoczenia.

Najważniejszym moim wynikiem prezentowanym w pracy [C7] jest sformułowanie własności blokowych dla rozpatrywanego problemu. Własności te umożliwiły istotne zredukowanie liczby przeglądanych rozwiązań sąsiednich poprzez eliminację rozwiązań nie rokujących poprawy. W ten sposób ograniczone sąsiedztwo zastosowałem w konstrukcji algorytmu opartego na metodzie przeszukiwania z zabronieniami. Wyniki testów komputerowych zamieszczone w pracy potwierdzają wysoką efektywność algorytmu. Dla wszystkich instancji testowych badany algorytm wyznaczył rozwiązania o czasie cyklu mniejszym od czasu cyklu rozwiązań wyznaczonych przez jeden z najlepszych wówczas algorytmów prezentowanych w literaturze.

Kolejne dwie prace z cyklu [C8] i [C9] dotyczą elastycznego cyklicznego problemu gniazdowego. Ze względu na ogromną liczbę sąsiednich rozwiązań oraz stosunkowo długi czas wyznaczenia czasu cyklu opracowałem trzy sposoby zmniejszenia czasu przeszukiwań:

1. zredukowanie sąsiedztwa do rozwiązań rokujących poprawę wartości funkcji celu,

2. zastosowanie technik przetwarzania równoległego w wyznaczaniu czasu cyklu,

Rempen

3. zastosowanie wstępnego szacowania wartości czasu cyklu.

Zredukowanie sąsiedztwa osiągnąłem dzięki sformułowaniu własności blokowych dla tego problemu. Natomiast w opracowaniu procedury przetwarzania równoległego wykorzystano fakt, że długości dróg $L_{(\pi_k(1))^{(1)},(\pi_k(1))^{(v)}}$ (27) mogą być wyznaczone niezależnie dla każdego $k \in \mathcal{M}$, zatem teoretycznie proces obliczeń można przyspieszyć m – *krotnie*. Mechanizm równoległego wyznaczenia wartości czasu cyklu zaimplementowałem w technologii SSE, która jest zaimpleentowana w praktycznie wszystkich współczesnych procesorach. Ograniczenia technologi umożliwiają, dla komputerowej reprezentacji przetwarzanych danych w algorytmie, na przyspieszenie 4-ro krotne. W przeprowadzonych przeze mnie badaniach eksperymentalnych przyspieszenie wynosiło od 3.3 do 3.8 razy.

Z równania (27) wynika, że jednym z dolnych ograniczeń czasu cyklu jest wartość

$$LB = \max_{1 \le k \le m} \{L_{s^{(1)},s^{(2)}} : \ L_{s^{(1)},s^{(2)}} < \infty\}|_{s=\pi_k(1)}. \tag{28}$$

Wyznaczenie wartości LB, dla zadanej kolejności wykonywania operacji, ma złożoność $O(mo)$ i sprowadza się do wyznaczenia długości odpowiednich najdłuższych dróg w fragmencie grafu obejmującym tylko jeden cykl.

Najważniejszym moim wynikiem prezentowanym w pracy [C9] jest opracowanie metody wyznaczenia dolnego oszacowania w postaci (28) dla wszystkich kolejności powstałych przez przesunięcie wybranej operacji na wszystkie możliwe pozycje w łącznym czasie $O(mo)$ oraz opracowanie dwufazowej metody przeglądania sąsiedztwa.

Przeglądanie sąsiedztwa w oparciu o wstępne szacowanie polega na wykonaniu obliczeń w dwóch fazach:

1. szybkie wyznaczenie dolnego oszacowania czasu cyklu dla wszystkich rozwiązań z sąsiedztwa,

2. dokładne wyznaczenie czasu cyklu rozwiązań w kolejności niemalejącej wartości dolnych oszacowań wyznaczonych w pierwszej fazie.

Łatwo można zauważyć, że w przypadku wyznaczania rozwiązania o najmniejszym czasie cyklu, realizację drugiej (czasochłonnej) fazy można przerwać po znalezieniu rozwiązania o czasie cyklu niemniejszym od dolnego oszacowania pozostałych rozwiązań.

Badania eksperymentalne skonstruowanego przeze mnie algorytmu opartego na metodzie przeszukiwania z zabronieniami potwierdzają wysoką efektywność algorytmu. Dla prawie wszystkich instancji testowych algorytm wyznaczył rozwiązania o czasie cyklu znacznie mniejszym od wartości funkcji C_{max}. Ponadto, zastosowanie prezentowanej strategii pozwoliło na zmniejszenie czasu obliczeń od 3 do ponad 300 razy.

4.3.8 Wykorzystanie wyników badań

Zastosowanie systemów informatycznych (w szczególności systemów klasy ERP II) we wspomaganiu różnych obszarów działalności przedsiębiorstw produkcyjnych jest jednym z najistotniejszych elementów obserwowanej obecnie 4 rewolucji przemysłowej. Niestety, pomimo nieustającego rozwoju takich systemów, nadal dużym wyzwaniem jest wspomaganie na etapie planowania operacyjnego, polegającego na określeniu momentów rozpoczęcia i zakończenia każdej istotnej czynności wykonywanej w systemie produkcyjnym.

Remen

Z naukowego punktu widzenia, przyczyna jest znana. Duża liczba różnego rodzaju ograniczeń występująca już w najprostszych systemach produkcyjnych powoduje duże trudności w modelowaniu systemów produkcyjnych oraz konstrukcji efektywnych algorytmów wspomagających harmonogramowanie.

Jednym z najtrudniejszy wyzwań przemysłowych, w których brałem udział było opracowanie systemu informatycznego wyznaczającego minimalną liczbę pracowników obsługujących moduły produkcyjne w przedsiębiorstwie produkującym siedzenia samochodowe dla jednego z największych producentów samochodów na świecie. Opracowanie modelu modułu produkcyjnych dla potrzeb wyznaczenia minimalnej liczby pracowników obsługujących je z pełną wydajnością wymagało, uwzględnienia następujących ograniczeń jednocześnie: (i) ograniczeń bez buforów, (i) wymogu produkcji cyklicznej.

Najważniejszym wynikiem naukowym moich prac badawczych dotyczących tego problemu jest opracowanie modelu grafowego dla dwóch ograniczeń jednocześnie, który został wykorzystany w konstrukcji algorytmu dokładnego. Rezultaty prac razem z wynikami testów na danych rzeczywistych zostały przedstawione w pracy [C10]. Korzyścią dla przedsiębiorstwa była możliwość wykorzystania prototypu aplikacji do weryfikacji obsady modułów produkcyjnych.

4.3.9 Podsumowanie

Cykl publikacji składa się z 8 publikacji oraz 2 referatów wygłoszonych na krajowych i międzynarodowych konferencjach naukowych. W referatach konferencyjnych zostały zaprezentowane wstępne wyniki najnowszych prac badawczych, które po rozszerzeniu zostaną skierowane do renomowanych czasopism naukowych.

Tematyka cyklu dotyczy harmonogramowania operacji w systemach produkcyjnych, w których występują dodatkowe ograniczenia. Biorąc pod uwagę fakt, że coraz więcej przedsiębiorstw odchodzi od tzw. produkcji masowej (którą dość dobrze opisywały klasyczne modele) występowanie dodatkowych ograniczeń w systemach produkcyjnych będzie coraz powszechniejsze. Niestety, skonstruowanie efektywnych algorytmów wspomagających harmonogramowanie w takich systemach jest dużym wyzwaniem dla naukowców. Świadczy o tym chociażby duża dysproporcja pomiędzy liczbą publikacji dla problemów klasycznych i problemów z dodatkowymi ograniczeniami. Ponadto stosowane przez naukowców podejścia do rozwiązywania takich problemów są z reguły dedykowane konkretnym ograniczeniom tj. nie umożliwiają one uwzględnienia kilku ograniczeń jednocześnie.

Prace cyklu oprócz tematyki łączy wieloetapowa metoda konstrukcji efektywnych algorytmów optymalizacyjnych dla problemów harmonogramowania (opisana szczegółowo w sekcji 4.3.2) opierająca się

- na opracowaniu modelu grafowego,
- badaniu własności modelu grafowego pod kątem opracowania własności eliminacyjnych (blokowych),
- opracowaniu akceleratorów obliczeń przyspieszających przeszukiwania przestrzeni rozwiązań.

Zastosowanie wymienionych elementów pozwala na konstrukcję algorytmów wyznaczających dobre rozwiązania dla problemów o rozmiarach odpowiadających rzeczywistym w czasie akceptowalnym przez praktyków.

Pempera)

Najważniejszymi wynikami prac badawczych są

- opracowanie modeli grafowych dla problemów szeregowania zadań z dodatkowymi ograniczeniami,
- opracowanie własności eliminacyjnych (blokowych) dla fragmentów ścieżek krytycznych objętych omawianymi ograniczeniami oraz dla problemów cyklicznych,
- opracowanie efektywnej czasowo metody wyznaczania czasu cyklu,
- opracowanie metod przyspieszania procesu przeszukiwania otoczenia,
- opracowanie strategii efektwnych czasowo przeszukiwania otoczenia.

Prace dotyczące problemów cyklicznych są jednymi z najnowszych osiągnięć. Sądzę, że opracowana metoda wyznaczania czasu z jednej strony będzie wykorzystywana przez Zespół, w którym pracuję do rozwiązywania nowych problemów cyklicznych bądź też konstruowania systemów informatycznych dla firm produkcyjnych, natomiast z drugiej strony będzie rozwijana przez badaczy z innych ośrodków naukowych na świecie, jak to miało miejsce w w przypadku problemów z ograniczeniami bez buforów i bez czekania.

Literatura

[1] Balas E., Vazacopoulos A.: A guided local search with shifting bottleneck for job-shop scheduling. Management Science, 262–275, (1998).

[2] Brucker P., Jurish B., Sievers B.: A fast branch&bound algorithm for the job shop problem. Discrete Applied Mathematics, 107–127, (1994).

[3] Brucker P., Kampmeyer T.: A general model for cyclic machine scheduling problems. Discrete Applied Mathematics, 156(13), 2561–2572, (2008).

[4] Cormen Thomas H., Leiserson Charles E., Rivest Ronald L.: Introduction to Algorithms. MIT Press and McGraw-Hill., Second Edition. MIT Press and McGraw-Hill, (2001).

[5] Glover F.: Tabu Search. Part I.. ORSA Journal of Computing, 190–206, (1989).

[6] Glover F.: Tabu Search. Part II. ORSA Journal of Computing , 4–32, (1990).

[7] Grabowski J.: On two-machine scheduling with release and due dates to minimize maximum lateness. Opsearch, 133–154, (1980).

[8] Grabowski J.: A new algorithm of solving the flow-shop problem. Operations Research in Progress, 57–75, (1982).

[9] Grabowski J., Nowicki E., Smutnicki C.: Block algorithm for scheduling of operations in job shop system. Przegląd Statystyczny, 67–80, (1988).

[10] Grabowski J., Nowicki E., Zdrzałka S.: A block approach for single machine scheduling with release dates and due dates. European Journal of Operational Research, 278–285, (1986).

Pempera

[11] Howard R.A.: Dynamic Programming and Markov Processes. Technology Press and Wiley, New York, NY, 1960.

[12] McCormick S.T., Pinedo M.L., Shenker S., Wolf B.: Sequencing in an assembly line with blocking to minimize cycle time. Operations Research, 37 (6), 925–935, (1989).

[13] Nowicki E., Smutnicki C.: A fast tabu search algorithm for the permutation flow-shop problem. European Journal of Operational Research, 160–175, (1996).

[14] Nowicki E., Smutnicki C.: A fast tabu search algorithm for the job-shop problem. Management Science, 797–813, (1996).

[15] Nowicki E., Smutnicki C.: Flow shop with parallel machines. A tabu search approch. European Journal of Operational Research, 226–253, (1998).

[16] Ramalingam G., Song J., Joskowicz L., R. E. Miller R. E.: Solving systems of difference constraints incrementally. Algorithmica, 261–275, (1999).

[17] Reeves C. R., Yamada T.: Genetic algorithms, path relinking and the flowshop sequencing problem. Evolutionary Computation, 45–60, (1998).

[18] Zdrzałka S., Grabowski J.: An algorithm for single machine sequencing with release dates to minimize maximum cost. Discrete Applied Mathematics, 73–89, (1989).

5 Omówienie pozostałych osiągnięć naukowo - badawczych (artystycznych)

Moja działalność naukowa skoncentrowana jest na opracowaniu teoretycznych aspektów konstrukcji efektywnych algorytmów wspomagających planowanie na poziomie operacyjnym w przedsiębiorstwach produkcyjnych. Wspomaganie harmonogramowania w systemach produkcyjnych wymaga od algorytmów wykonywanych na współczesnych systemach komputerowych wyznaczania dobrej jakości harmonogramów w czasie akceptowalnym przez praktyków. Z tych powodów prace badawcze prowadzone przeze mnie obejmują następujące etapy

1. budowę modeli obliczeniowych pozwalających na rozwiązanie problemu,
2. badanie istnienia własności eliminacyjnych,
3. budowę mechanizmów dekompozycji i agregacji obliczeń (akceleratorów),
4. wykorzystanie technik przetwarzania równoległego.

Realizacja etapu 1 pozwala na konstrukcję algorytmu optymalizacyjnego, natomiast etapu 2 prowadzi do zwiększenia jego efektywności tj. wyznaczenia dobrej jakości rozwiązań w mniejszej liczbie iteracji. Prace obejmujące etapy 3 i/lub 4 mają na celu przyspieszenie

Peyper

procesu obliczeń, które jest najczęściej niezbędne w przypadku rozwiązywania problemów o dużych rozmiarach.

Oprócz prac badawczych zawartych w cyklu publikacji postępowanie to zastosowałem do rozwiązywania problemów optymalizacyjnych związanych z logistyką produkcji, transportem towarów.

5.1 Logistyka produkcji

Współczesne przedsiębiorstwa coraz częściej konkurują ze sobą tzw. doskonałością operacyjną. Dotyczy ona również logistyki produkcji, której zadaniem jest zaopatrzenie systemu każdego z punktów produkcyjnych w niezbędne surowce, materiały, półprodukty, części oraz transport produktów pomiędzy kolejnymi stanowiskami. Efektywne zarządzanie logistyką w elastycznych systemach produkcyjnych wymaga dostosowania harmonogramu realizacji zadań systemu logistycznego do nadrzędnych planów produkcyjnych i/lub jednoczesnego uwzględnienia logistycznych i operacyjnych w harmonogramowaniu produkcji.

W ramach moich prac badawczych zajmowałem się opracowaniem algorytmów wspomagających harmonogramowanie czynności logistycznych w pewnym przedsiębiorstwie produkującym wiązki elektryczne [1]. W celu rozwiązania problemu zbudowałem model grafowy, a następnie skonstruowałem algorytm optymalizacyjny wykorzystujący własności eliminacyjne problemu oraz akcelerację obliczeń. W kolejnym z rozważanych problemów [3], tj. w systemie produkcyjnym z transportem karuzelowym, w naturalny sposób, transport produktów pomiędzy stanowiskami oraz produkcja są zintegrowane. Dla tego systemu produkcyjnego podobnie jak w poprzednim przypadku opracowałem model grafowy oraz skonstruowałem efektywny algorytm optymalizacyjny.

Innym istotnym problemem związanym z logistyką produkcji, jest organizacja transportu wewnątrz magazynów centralnych. W pracy [3] wykazałem, że dla problemu harmonogramowania zadań transportowych można wykazać pewne własności eliminacyjne. W oparciu o te własności skonstruowałem efektywny algorytm optymalizacyjny, który można również wykorzystać w planowaniu czynności logistycznych w dużych centrach logistycznych.

[1] **Jarosław Pempera**: Algorytmy wspomagające planowanie czynności towarowania produkcji. Zeszyty Naukowe - Politechnika Śląska. Automatyka. 2008, z. 151, s. 167-172.

[2] **Jarosław Pempera**: Harmonogramownie operacyjne w przepływowym systemie produkcyjnym z transportem karuzelowym. Automatyzacja procesów dyskretnych: teoria i zastosowania. T. 2 / pod red. Andrzeja Świerniaka i Jolanty Krystek. Gliwice: Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2012. s. 183-192.

[3] **Jarosław Pempera**: Zastosowanie teorii blokowej w konstrukcji efektywnych algorytmów dla problemu harmonogramowania zadań transportowych w systemie magazynowym. Komputerowo zintegrowane zarządzanie. T. 2 / pod red. Ryszarda Knosali. Opole: Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, 2011. s. 240-250.

Pempera

5.2 Transport towarów

Transport towarów jest usługą, która z jednej strony ma istotne znaczenie w rozwoju gospodarczym społeczeństw, natomiast z drugiej strony generuje duże koszty oraz jest uciążliwe dla środowiska. Umiejętne zarządzanie transportem pozwala na zmniejszenie obu niepożądanych zjawisk. Planowanie transportu towarów nie jest procesem łatwym ponieważ występuje w nim dużo ograniczeń. W swoich badaniach dotyczących tego obszaru skupiłem się na często pomijanych przez badaczy ograniczeniach w systemach transportowych wynikających z przepisów dotyczących przerw w pracy kierowców [1]. Dla tego problemu skonstruowałem efektywny algorytm optymalizacyjny.

Kolejne prace [2] i [3] dotyczą optymalizacji transportu w określonych branżach produkcyjnych. W pracy [2] przedstawiono algorytm optymalizacyjny dla wspomagający zaopatrzenia sieci zakładów przetwarzających mleko w surowiec. Opracowany przeze mnie algorytm pozwala na zminimalizowanie kosztów transportu mleka oraz zmaksymalizowanie zysków z przetwórstwa mleka dla średniej wielkości sieci zakładów. W przypadku problemu dystrybucji paliw [3] skonstruowałem algorytm optymalizacji dwukryterialnej. W przypadku konstruowania tego algorytmu zostały uwzględnione ograniczenia związane z dystrybucją paliw ciekłych.

Ostatnią z prac prezentowanych w tej sekcji jest praca poświęcona transportowi intermodalnemu realizowanemu za pomocą pociągów towarowych [4]. W pracach badawczych dotyczących tej problematyki skonstruowałem algorytmy optymalizacyjne, w których uwzględniłem najistotniejsze ograniczenia wynikające z realizacji transportu kolejowego.

[1] **Jarosław Pempera**: Planowanie tras pojazdów z ograniczeniami czasu pracy kierowców. Logistyka i zarządzanie produkcją - nowe wyzwania, odległe granice. Red. Marek Fertsch, Katarzyna Grzybowska, Agnieszka Stachowiak. Poznań: Instytut Inżynierii Zarządzania PPozn., 2007. s. 156-163.

[2] Anna Burduk, Wojciech Bożejko, **Jarosław Pempera**, Kamil Musiał: On the simulated annealing adaptation for tasks transportation optimization. Logic Journal of the IGPL. 2018, vol. 26, nr 6, s. 581–592.

[3] **Jarosław Pempera**, Dominik H. Żelazny: Multi-criteria optimization in fuel distribution. Research in Logistics & Production. 2015, vol. 5, nr 1, s. 77-84.

[4] Wojciech Bożejko, Radosław Grymin, **Jarosław Pempera**: Scheduling and routing algorithms for rail freight transportation. Procedia Engineering [Dokument elektroniczny]. 2017, vol. 178, s. 206-212.

5.3 Przetwarzanie równoległe

Przetwarzanie równoległe jest jednym ze sposobów zmienjszenia czasu działania algorytmów realizowanych na komputerach. Współczesnie dostępne są różne technologie umożliwiające realizację obliczeń równoległych. Niestety, pomimo zwiększenia liczby procesorów oraz dostępnej pamięci, skonstruowanie efektywnych algorytmów równoległych nadal nie jest łatwe.

Pempera

Dużą barierą w przenoszeniu algorytmów sekwencynych na równolegle jest dostęp do pamięci. Dlatego wyzwaniem dla badaczy konstruujacych algorytmy realizujące przetwarzanie równoległe jest opracownie metody dekompozycji i agregacji obliczeń tak, aby w pełni można było wykorzstać zalety przetwarzania równoległego.

Głównymi technologiami wspomagającymi obliczenia równoległe wykorzystywanymi w moich badaniach są: (*i*) SSE zaiplementowana w większości współczesnych procesorów, (*ii*) CUDA zaimplenetowana w kartach graficznych firmy NVIDIA. Bazując na wynikach teoretycznych dotyczących przetwarzania równoległego dr hab. Wojciecha Bożejki oraz własności problemów skontruowałem algorytmy optymalizacyjne dla: cyklicznego elastycznego problemu gniazdowego [1], problemu przepływowego z kryterium minimalizacji sumy czasów zakończenia zadań [2] oraz dla problemu gniazdowego [3]. Dla wszystkich z wymienionych problemów zrównoleglenie dotyczyło wyznaczenia wartości funcji celu.

[1] Wojciech Bożejko, **Jarosław Pempera**, Mieczysław Wodecki: Parallel simulated annealing algorithm for cyclic flexible job shop scheduling problem. Artificial intelligence and soft computing: 14th International Conference, ICAISC 2015, Zakopane, Poland, June 14-18, 2015: proceedings. Pt. 2 / Leszek Rutkowski [i in.] (eds.). Cham [i in.]: Springer, cop. 2015. s. 603-612.

[2] Wojciech Bożejko, **Jarosław Pempera**: Parallel tabu search algorithm for the permutation flow shop problem with criterion of minimizing sum of job completion times. Conference on Human System Interaction [Dokument elektroniczny]: conference proceedings, Krakow, Poland, May 25-27, 2008. [Piscataway, NJ]: IEEE, cop. 2008.

[3] Wojciech Bożejko, **Jarosław Pempera**, Czesław Smutnicki: Parallel simulated annealing for the job shop scheduling problem. Lecture Notes in Computer Science [Dokument elektroniczny]. 2009, vol. 5544, s. 631-640.

5.4 Prace z doktorantami

Od 2012 współpracuję z doktorantami Politechniki Wrocławskiej w charakterze opiekuna naukowego i/lub promotora pomocniczego. Wspólny dorobek publikacyjny obejmuje 12 artykułów z których dwa są na liście JCR. Tematyka publikacji związana jest z próbami znalezienia obszarów badawczych dla przyszłej pracy doktorskiej, bądź realizacją badań zwiazanych z wybranym tematem pracy doktorskiej. Obecnie współpracuję w roli promotora pomocniczego z następującymi doktorantami:

[1] mgr inż. Radosław Grymin, temat „Metody rozwiazywania dyskretnociągłych problemów marszrutyzacji", Wydział Elektroniki Politechniki Wrocławskiej, promotor dr hab. Wojciech Bożejko prof. PWr, **promotor pomocniczy dr inż. Jarosław Pempera**,

[2] mgr inż. Kamil Musiał, temat „Metodyka harmonogramowania produkcji w aspekcie optymalizacji zasobów krytycznych z wykorzystaniem metaheurystyk", promotor dr hab. Anna Burduk prof. PWr, Wydział Mechaniczny Politechniki Wrocławskiej, promotor dr hab. Anna Burduk prof. PWr, **promotor pomocniczy dr inż. Jarosław Pempera**.

Pempera

Tematyka pracy doktorskiej mgr inż. Radosława Grymina dotyczy rozwiązywania dyskretnociągłych problemów marszrutyzacji. W ramach współpracy został opracowany algorytm wyznaczania optymalnej trasy obiektu latającego (drona), którego trasa lotu musi przebiegać przez punkty znajdujące się w zadanej odległości od punktu inspekcji. W ramach prac badawczych sformułowałem własności problemu, które zostały wykorzystane w konstrukcji algorytmu optymalizacyjnego. Wyniki badań zostały przedstawione w pracy [1]. Wcześniejsze prace z doktorantem dotyczyły zagadnień transportowych w szczególności transportu kolejowego [2].

Mgr inż. Kamil Musiał jest doktorantem Wydziału Mechanicznego Politechniki Wrocławskiej. Jego praca doktorska dotyczy harmonogramowania produkcji poprzez optymalizację wąskich gardeł w produkcji średnio seryjnej w pewnym przedsiębiorstwie. W ramach prac badawczych powstała aplikacja komputerowa wyznaczająca minimalną liczbę pracowników niezbędną do realizacji zadań produkcyjnych. Prezentacja aplikacji została przychylnie przyjęta na spotkaniu w firmie przez kierowników kilku kluczowych działów. Wcześniej doktorant aktywnie uczestniczył w realizacji pracy dotyczącej marszrutyzacji pojazdów dla sieci zakładów przetwarzających mleko [3], która została opublikowana wVczasopiśmie z listy JCR.

Wcześniej współpracowałem z dr inż. Dominikiem Żelaznym w charakterze opiekuna naukowego. Promotorem jego pracy doktorskiej pt. „Wielokryterialna optymalizacja w szeregowaniu zadań" jest prof. dr hab. Czesław Smutnicki. Problematyka jego badań naukowych dotyczyła optymalizacji wielokryterialnej w szeregowaniu zadań. Moim głównym udziałem w realizacji wspólnych prac było badanie własności problemów pod kątem wykorzystania ich konstrukcji algorytmów optymalizacyjnych. Dla przykładu, na potrzeby najważniejszej wspólnej pracy [4], która została opublikowana w czasopiśmie z listy JCR, opracowałem oryginalną strategię przeglądania otoczenia w algorytmie opartym na metodzie symulowanego wyżarzania dla problemu przepływowego. Polega ona na wyznaczeniu w każdej iteracji wartości funkcji celu dla kilku rozwiązań. W kolejnej iteracji algorytmu rozwiązaniem bazowym jest pierwsze zaakceptowane rozwiązanie. Badania naukowe dotyczące problemów wielokryterialnych dotyczyły również optymalizacji w dystrybucji dóbr [5] oraz optymalizacji w problemach przepływowych [6].

[1] Radosław Grymin, Wojciech Bożejko, **Jarosław Pempera**, Mieczysław Wodecki: Algorytm rozwiązywania dyskretno-ciągłego problemu inspekcji. Automatyzacja procesów dyskretnych: teoria i zastosowania. T. 1 / pod red. Andrzeja Świerniaka i Jolanty Krystek. Gliwice: Wydawnictwo Politechniki Śląskiej, 2018. s. 87-94.

[2] Wojciech Bożejko, Radosław Grymin, **Jarosław Pempera**: Scheduling and routing algorithms for rail freight transportation. Procedia Engineering [Dokument elektroniczny]. 2017, vol. 178, s. 206-212.

[3] Anna Burduk, Wojciech Bożejko, **Jarosław Pempera**, Kamil Musiał: On the simulated annealing adaptation for tasks transportation optimization. Logic Journal of the IGPL. 2018, vol. 26, nr 6, s. 581–592.

[4] Czesław Smutnicki, **Jarosław Pempera**, Jarosław S. Rudy, Dominik H. Żelazny: A new approach for multi-criteria scheduling. Computers & Industrial Engineering. 2015, vol. 90, s. 212-220.

Pempera

[5] **Jarosław Pempera**, Dominik H. Żelazny: Multi-criteria optimization in fuel distribution. Research in Logistics & Production.

[6] Dominik H. Żelazny, **Jarosław Pempera**: Solving multi-objective permutation flowshop scheduling problem using CUDA. MMAR 2015: 20th International Conference on Methods and Models in Automation & Robotics [Dokument elektroniczny]: 24-27 August, 2015, Międzyzdroje, Poland. [Piscataway, NJ]: IEEE, cop. 2015. s. 347-352.

5.5 Współpraca z przemysłem

W roku 2014 na zlecenie firmy Crusar został opracowany i zaimplementowany system informatyczny wyznaczający synergie w transporcie drogowym. W projekcie tym opracowałem i zaimplementowałem algorytmy wyznaczające synergie w transporcie międzynarodowych realizowanym przez setki pojazdów zarządzanych przez firmę spedycyjną zleceniodawcę projektu.

W roku 2018 byłem głównym wykonawcą systemu informatycznego wspomagającego harmonogramowanie zadań w dziale pras w firmie Eletrolux. W realizacji projektu wykorzystałem wiedzę i doświadczenie wynikające z moich prac badawczych. Nie wchodząc w szczegóły projektu, w konstruowaniu algorytmów wspomagających harmonogramowanie pracy pras należało uwzględnić takie ograniczenia jak: (*i*) dostępność pras, (*ii*) dostępność narzędzi (sztanc), (*iii*) dostępność surowców, (*iv*) żądane terminy zakończenia realizacji określonych części.

Najtrudniejszym w realizacji ograniczeniem było uwzględnienie w harmonogramach pracy brygady przezbrajaczy, tj. wszystkie przezbrojenia wykonywane w dziale pras wykonywała tylko jedna brygada przezbrajaczy przy czym w tym samym czasie mogła wykonywać tylko jedno. W wyniku bardzo intensywnych prac badawczo-implementacyjnych powstał aplikacja komputerowa wyznaczająca dobrej jakości harmonogramy w czasie kilkudziesięciu sekund dla 2 tygodniowego horyzontu czasowego.

W ramach współpracy z firmą TOYOTA zrealizowane zostały dwa zadania tj. opracowanie systemu optymalizującego transport wewnętrzny realizowany przez wózki ciągnikowe (R-cart) oraz przeprowadzenie symulacji linii produkcyjnej pewnej grupy podzespołów na potrzeby analizy metodą yamazumi. W obu zadaniach byłem głównym konstruktorem algorytmów komputerowych oraz głównym wykonawcą aplikacji komputerowych.

Pierwsze z zadań dotyczyło logistyki produkcji. W wyniku prac badawczych powstał system informatyczny wyznaczający harmonogram wykonywania zadań transportowych dla wózków ciągnikowych zaopatrujących linię montażu w części i materiały.

W drugim przypadku symulacja linii produkcyjnej sprowadzała się do wyznaczenia długoterminowego (obejmującego ok 1 miesiąca) harmonogramu operacyjnego w pełni zautomatyzowanej linii produkcji pewnej grupy podzespołów. Linia składała się z wielu sekcji maszyn obsługiwanych przez zrobotyzowane środki transportu, natomiast pomiędzy sekcjami znajdowały się bufory o ograniczonej pojemności (wewnątrz sekcji nie było wewnętrznych buforów). W harmonogramach należało uwzględnić okresy niezbędne na wymianę narzędzi oraz kontrolę jakości. Kolejnym wyzwaniem, tym razem związanym z prędkością przetwarzania, była liczba operacji wykonywana w tak długim okresie, która była bliska 1 miliona.

Pempera

Pełna lista projektów realizowanych z przemysłem jest następująca:

[1] Crusar – Koncepcja modelowego zintegrowanego procesu transportu Synergy Spotting System (umowa o wykonanie pracy badawczo-rozwojowej nr S/557/14 z dnia 31.10.2014 r.), 2015.

[2] Electrolux Polska– System harmonogramowania zadan dla działu pras Electrolux Polska w Swidnicy, 2018.

[3] Toyota Motor Manufacturing Poland – TNGA 2.0 - Optimalization of internal transport wykonania prac badawczo-rozwojowych dla TMMP dotyczących symulacji i optymalizacji transportu (Purchase Order No. 51516205 for TMMP), 2018.

[4] Toyota Motor Manufacturing Poland – RDMC System for optimization of the chip collection process by AGVs (Purchase Order No. 29500037), 2018.

6 Dane bibliograficzne

	Przed uzyskaniem stopnia doktora	Po uzyskaniu stopnia doktora	Razem
Artykuły w czasopismach z listy JCR (Journal Citation Reports)	2	10	12
Rozdziały w monografiach	0	14	14
Redakcja monografii	0	2	2
Rozdziały w książkach	0	22	22
Prace w materiałach konferencyjnych	11	14	25
RAZEM	13	83	96

6.1 Analiza cytowań

Liczba cytowań według różnych baz danych

Baza danych	H-index	Cytowania bez autocytowań	Cytowania łącznie
Web of Science Core Collection	7	414	444
Scopus	9	553	581
Google Scholar	9	—*	857

*brak danych

Wskaźnik	Przed uzyskaniem stopnia doktora	Po uzyskaniu stopnia doktora	Razem
Sumaryczny współczynnik wpływu (Impact Factor) według daty publikacji	0.928	17,372	18,3
Liczba prac w bazie Web of Science	2	22	24
Liczba punktów za publikacje według listy MNiSW	0	391	391

Pempera