ZESZYTY
NAUKOWE
POLITECHNIKI
ŚLĄSKIEJ

FRANCISZEK MARECKI

MODELE MATEMATYCZNE I ALGORYTMY ALOKACJI OPERACJI I ZASOBÓW NA LINII MONTAŻOWEJ

P 3341/86

AUTOMATYKA

Z. 82
GLIWICE
1986

POLITECHNIKA ŚLĄSKA

ZESZYTY NAUKOWE

Nr 868

FRANCISZEK MARECKI

P.3341/86

MODELE MATEMATYCZNE I ALGORYTMY ALOKACJI OPERACJI I ZASOBÓW NA LINII MONTAŻOWEJ

GLIWICE 1986

cz. 13.033

OPINIODAWCY:

Prof. zw. dr inż. Henryk Górecki
Prof. dr hab. inż. Jan Węglarz

KOLEGIUM REDAKCYJNE

REDAKTOR NACZELNY — Prof. dr hab. inż. Wiesław Gabzdyl
REDAKTOR DZIAŁU — Dr inż. Anna Skrzywan-Kosek
SEKRETARZ REDAKCJI — Mgr Elżbieta Stinzing
CZŁONKOWIE KOLEGIUM — Prof. dr hab. inż. Adolf Maciejny
— Prof. dr inż. Stanisław Malzacher
— Prof. dr hab. inż. Bronisław Skinderowicz

OPRACOWANIE REDAKCYJNE
Mgr Anna Błażkiewicz

Wydano za zgodą Rektora
Politechniki Śląskiej

PL ISSN 0434-0760

Praca była częściowo finansowana przez R. P. I. 02
„Teoria sterowania i optymalizacji ciągłych układów dynamicznych
i procesów dyskretnych" oraz CPBR 7.4
„Mechanizacja i automatyzacja procesów montażu".

Dział Wydawnictw Politechniki Śląskiej
ul. Kujawska 3, 44-100 Gliwice

Nakł. 170+85 Ark. wyd. 15,5 Ark. druk 15,125 Papier offset. kl III 70x100, 70 g
Oddano do druku 16.12.85 Podpis. do druku 8.04.86 Druk ukończ. w czerwcu 1986
Zam. 1269/85 K-23 Cena zł 233,—

Skład, fotokopie, druk i oprawę
wykonano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach

P 301/86

Moim Rodzicom

SPIS TREŚCI

Str.

SPIS OZNACZEŃ 19

1. WPROWADZENIE 23

1.1. Wstęp 23
1.2. Opis linii i procesu montażu 26
1.3. Charakterystyka podstawowych problemów 29
1.4. Problem BLM 31
1.4.1. Klasyczny problem BLM 32
1.4.2. Algorytmy BLM 34
1.5. Metoda programowania wieloetapowego 35

2. BLM METODĄ PROGRAMOWANIA WIELOETAPOWEGO 39

2.1. Wstęp 39
2.2. Podstawowe określenia 39
2.3. Reguły eliminacji stanów niepersepektywicznych 42
2.4. Algorytmy programowania wieloetapowego 46
2.4.1. Algorytm binarny 46
2.4.2. Algorytm szeregowania 47
2.4.3. Algorytm przydziału 47
2.4.4. Algorytm harmonogramowania 48
2.5. Przykłady 49
2.5.1. Testy komputerowe 60
2.6. Analiza algorytmów 67
2.6.1. Efektywność algorytmów 67
2.6.2. Rozwiązanie przybliżone 70
2.6.3. Wybór algorytmu 72

3. UOGÓLNIONE MODELE BLM 73

3.1. Wstęp 73
3.2. Dodatkowe parametry operacji 73
3.2.1. Model matematyczny 74
3.2.2. Algorytm 75
3.3. Enklawy operacji 76
3.3.1. Model matematyczny 77
3.3.2. Algorytm dla enklaw podzielnych 79
3.3.3. Algorytm dla enklaw niepodzielnych 81

Str.

3.4. Montaż przestrzenny - fronty pracy 82
3.4.1. Model matematyczny 83
3.4.2. Algorytm 84
3.5. Koszt montażu 86
3.5.1. Model matematyczny 87
3.5.2. Algorytm 87
3.5.3. Eliminowanie stanów nieperspektywicznych 91

4. ALOKACJA OPERACJI. SZEREGOWE STANOWISKA PRACY 97

4.1. Wstęp 97
4.2. Ograniczenia pozycyjne 98
4.2.1. Model matematyczny 98
4.2.2. Algorytm 100
4.3. Ograniczenia czasowe 105
4.3.1. Model matematyczny 105
4.3.2. Algorytm 106
4.4. Ograniczenia zakresowe 109
4.4.1. Model matematyczny 109
4.4.2. Algorytm 111
4.5. Linia asymetryczna 116
4.5.1. Model matematyczny 116
4.5.2. Algorytm 118

5. ALOKACJA OPERACJI. RÓWNOLEGŁE STANOWISKA PRACY 122

5.1. Wstęp 122
5.2. Model podstawowy 122
5.2.1. Model matematyczny 122
5.2.2. Algorytm 125
5.2.3. Eliminowanie stanów nieperspektywicznych 129
5.3. Relacja wykluczania operacji 135
5.3.1. Model matematyczny 135
5.3.2. Algorytm 136
5.4. Relacja synchronizacji operacji 138
5.4.1. Model matematyczny 138
5.4.2. Algorytm 139

6. ALOKACJA OPERACJI I ZASOBÓW 144

6.1. Wstęp 144
6.2. Alokacja operacji i kontenerów 145
6.2.1. Model matematyczny 145
6.2.2. Algorytm 147
6.2.3. Eliminowanie stanów nieperspektywicznych 150
6.2.4. Polioptymalizacja 156

Str.

6.3. Alokacja operacji i narzędzi 158
6.3.1. Model matematyczny 158
6.3.2. Algorytm 159
6.4. Alokacja operacji i monterów 164
6.4.1. Model matematyczny 164
6.4.2. Algorytm 165

7. ZAKOŃCZENIE 171

DODATEK - Modele matematyczne i algorytmy sterowania montażem dynamicznym 175

SPIS OZNACZEŃ 175

1. WPROWADZENIE 178

2. ZAŁOŻENIA 179

3. PODSTAWOWE OKREŚLENIA 180
3.1. Stan procesu montażu 180
3.2. Sterowanie procesem montażu 183

4. RÓWNANIA STANU 185

5. PROBABILISTYCZNY MODEL MONTAŻU 190

6. MODEL MONTAŻU I ZAPASÓW DETALI 194

7. STEROWANIE MONTAŻEM JEDNOWERSYJNYM 199
7.1. Sterowanie statyczne 200
7.2. Sterowanie pseudostatyczne 201
7.3. Sterowanie pseudodynamiczne 203
7.4. Sterowanie dynamiczne 205

8. MODEL MONTAŻU WIELOWERSYJNEGO 207

9. STEROWANIE DYNAMICZNE MONTAŻEM WIELOWERSYJNYM 210

10. ZAKOŃCZENIE 213

STRESZCZENIA DODATKU 214

LITERATURA 216

STRESZCZENIA ROZPRAWY 232

SPIS RYSUNKÓW W PRACY

Rys. 1.1. Linia montażowa: 1 - transporter główny, 2 - montowany obiekt wchodzący na pierwszą stację linii (chassis montowanego obiektu), 3 - zmontowany kompletnie obiekt schodzący z ostatniej stacji linii, 4 - montowany obiekt wchodzący na k-tą stację, 5 - pola dla narzędzi i kontenerów z detalami, 6 - k-ta stacja montażowa

Rys. 1.2. Charakterystyka ekonomiczna linii montażowej

Rys. 1.3. Charakterystyka statyczna linii montażowej

Rys. 2.1. Diagram bezpośrednich poprzedników i następników (numery operacji zapisano wewnątrz węzłów, natomiast czasy operacji na zewnątrz)

Rys. 2.2. Sieć stanów algorytmu binarnego

Rys. 2.3. Sieć stanów algorytmu szeregowania

Rys. 2.4. Sieć stanów algorytmu przydziału

Rys. 2.5. Sieć stanów algorytmu harmonogramowania

Rys. 2.6. Schemat blokowy algorytmu BLM z regułą wyboru FIFO

SPIS RYSUNKÓW W DODATKU

Rys. D.1. Schemat blokowy obliczeń dla funkcji logicznej h

Rys. D.2. Ogólny schemat blokowy obliczeń dla deterministycznego procesu montażu

Rys. D.3. Schemat blokowy obliczeń dla funkcji logicznej h_s (R - generator liczb pseudolosowych)

Rys. D.4. Schemat blokowy obliczeń dla funkcji logicznej h_d

Rys. D.5. Schemat blokowy obliczeń czasów cykli c^l oraz chwil τ^l uruchamiania transportera głównego

SPIS TABEL

Tabela 2.1. Porównanie sieci stanów w algorytmach z regułą dominacji stanów oraz bez dominacji stanów

Tabela 2.2. Porównanie obliczeń komputerowych bez dominacji stanów oraz z dominację stanów

СОДЕРЖАНИЕ

Стр.

СПИСОК ОБОЗНАЧЕНИЙ 19

1. ВВЕДЕНИЕ 23

1.1. Вступление 23
1.2. Описание линии и процесса сборки 26
1.3. Характеристика основных проблем 29
1.4. Проблема балансирования сборочной линией (БСЛ) 31
1.4.1. Классическая проблема БСЛ 32
1.4.2. Алгоритм БСЛ 34
1.5. Метод многоэтапного программирования 35

2. БСЛ МЕТОДОМ МНОГОЭТАПНОГО ПРОГРАММИРОВАНИЯ 39

2.1. Вступление 39
2.2. Основные определения 39
2.3. Правила исключения неперспективных состояний 42
2.4. Алгоритм многоэтапного программирования 46
2.4.1. Бинарный алгоритм 46
2.4.2. Алгоритм упорядочения 47
2.4.3. Алгоритм приписания 47
2.4.4. Алгоритм календарного планирования 48
2.5. Примеры 49
2.5.1. Компьютерные тесты 60
2.6. Анализ алгоритмов 67
2.6.1. Эффективность алгоритмов 67
2.6.2. Приближенное решение 70
2.6.3. Выбор алгоритма 72

3. ОБОБЩЁННЫЕ МОДЕЛИ БСЛ 73

3.1. Вступление 73
3.2. Добавочные параметры операций 73
3.2.1. Математическая модель 74
3.2.2. Алгоритм 75
3.3. Энклавы операций 76
3.1.1. Математическая модель 77
3.2.2. Алгоритм для делимых энклав 79
3.3.3. Алгоритм для неделимых энклав 81

Стр.

3.4. Пространственная сборка - фронты работ 82
3.4.1. Математическая модель 83
3.4.2. Алгоритм 84
3.5. Стоимость сборки 86
3.5.1. Математическая модель 87
3.5.2. Алгоритм 87
3.5.3. Исключение неперспективных состояний 91

4. АЛЛОКАЦИЯ ОПЕРАЦИЙ. ПОСЛЕДОВАТЕЛЬНЫЕ РАБОЧИЕ МЕСТА 97

4.1. Вступление 97
4.2. Позиционные ограничения 98
4.2.1. Математическая модель 98
4.2.2. Алгоритм 100
4.3. Временные ограничения 105
4.3.1. Математическая модель 105
4.3.2. Алгоритм 106
4.4. Диапазонные ограничения 109
4.4.1 Математическая модель 109
4.4.2. Алгоритм 111
4.5. Асимметрическая линия 116
4.5.1. Математическая модель 116
4.5.2. Алгоритм 118

5. АЛЛОКАЦИЯ ОПЕРАЦИЙ. ПАРАЛЛЕЛЬНЫЕ РАБОЧИЕ МЕСТА 122

5.1. Вступление 122
5.2. Основная модель 122
5.2.1. Математическая модель 122
5.2.2. Алгоритм 125
5.2.3. Исключение неперспективных состояний 129
5.3. Отношение исключения операций 135
5.3.1. Математическая модель 135
5.3.2. Алгоритм 136
5.4. Отношение синхронизации операций 138
5.4.1. Математическая модель 138
5.4.2. Алгоритм 139

6. АЛЛОКАЦИЯ ОПЕРАЦИЙ И РЕСУРСОВ 144

6.1. Вступление 144
6.2. Аллокация операций и контейнеров 145
6.2.1. Математическая модель 145
6.2.2. Алгоритм 147
6.2.3. Исключение неперспективных состояний 150
6.2.4. Полиоптимализация 156

Стр.

6.3. Аллокация операций и инструментов 158
6.3.1. Математическая модель 158
6.3.2. Алгоритм 159
6.4. Аллокация операций и монтёров 164
6.4.1. Математическая модель 164
6.4.2. Алгоритм 165

7. ОКОНЧАНИЕ 171

ДОПОЛНЕНИЕ - Математические модели и алгоритмы управления динамической сборкой 175

СПИСОК ОБОЗНАЧЕНИЙ 175

1. ВВЕДЕНИЕ 178

2. ПРЕДПОЛОЖЕНИЯ 179

3. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ 180
3.1. Состояние сборочного процесса 180
3.2. Управление сборочным процессом 183

4. УРАВНЕНИЯ СОСТОЯНИЯ 185

5. ВЕРОЯТНОСТНАЯ МОДЕЛЬ СБОРКИ 190

6. МОДЕЛЬ СБОРКИ И ЗАПАСОВ ДЕТАЛЕЙ 194

7. УПРАВЛЕНИЕ ОДНОВЕРСИОННОЙ СБОРКОЙ 198
7.1. Статическое управление 200
7.2. Псевдостатическое управление 201
7.3. Псевдодинамическое управление 208
7.4. Динамическое управление 205

8. МОДЕЛЬ МНОГОВЕРСИОННОЙ СБОРКИ 207

9. ДИНАМИЧЕСКОЕ УПРАВЛЕНИЕ МНОГОВЕРСИОННОЙ СБОРКОЙ 210

10. ОКОНЧАНИЕ 213

РЕЗЮМЕ ДОПОЛНЕНИЯ 214

СПИСОК ЛИТЕРАТУРЫ 216

РЕЗЮМЕ РАБОТЫ 232

СПИСОК РИСУНКОВ

Рис. 1.1. Сборочная линия 1 - главный конвейер, 2 - собираемый объект, входящий на первую станцию линии (шасси собираемого объекта), 3 - полностью собранный объект выходящий с последней станции, 4 - собираемый объект входящий на к-тую станцию, 5 - место для инструментов и контейнеров с деталями, 6 - к-тая сборочная станция

Рис. 1.2. Экономическая характеристика сборочной линии

Рис. 1.3. Статическая характеристика сборочной линии

Рис. 2.1. Диаграмма непосредственных предшественников и наследников (номера операций записаны внутри узлов, времена операций - снаружи)

Рис. 2.2. Сеть состояний бинарного алгоритма

Рис. 2.3. Сеть состояний алгоритма упорядочения

Рис. 2.4. Сеть состояний алгоритма приписания

Рис. 2.5. Сеть состояний алгоритма календарного планирования

Рис. 2.6. Блок-схема алгоритма БСЛ с правилом выбора FIFO

СПИСОК РИСУНКОВ К ДОПОЛНЕНИЮ

Рис. Д.1. Блок-схема расчёта логической функции h

Рис. Д.2. Общая блок-схема расчёта детерминистического сборочного процесса

Рис. Д.3. Блок-схема расчёта логической функции (R-генератор псевдослучайных чисел) h_s

Рис. Д.4. Блок-схема расчёта логической функции h_d

Рис. Д.5. Илок-схема расчёта времён циклов c^1 и моментов T^1 пуска главного конвейера

СПИСОК ТАБЛИЦ

Таблица 2.1. Сравнение сети состояний в алгоритмах с правилом доминации состояний и без доминации состояний

Таблица 2.2. Сравнение компьютерных расчётов без доминации состояний и с доминацией состояний

CONTENTS

Page

LIST OF SYMBOLS 19

1. INTRODUCTION 23

1.1. Preface 23
1.2. Description of a line and an assembly process 26
1.3. Characterization of fundamental problems 29
1.4. ALB problem 31
1.4.1. The standard ALB problem 32
1.4.2. Algorithms for the ALB problem 34
1.5. Method of multi-stage programming 35

2. ALB BY THE MULTI-STAGE PROGRAMMING METHOD 39

2.1. Introduction 39
2.2. Basic definitions 39
2.3. Rules of unperspective states elimination 42
2.4. Algorithms of the multi-stage programming 46
2.4.1. Binary algorithm 46
2.4.2. Sequencing algorithm 47
2.4.3. Allocation algorithm 47
2.4.4. Scheduling algorithm 48
2.5. Examples 49
2.5.1. Computational tests 60
2.6. Analysis of the algorithms 67
2.6.1. Efficiency of the algorithms 67
2.6.2. Approximate solution 70
2.6.3. Selection of algorithm 72

3. GENERALIZED ALB MODELS 73

3.1. Introduction 73
3.2. Additional parameters of operations 73
3.2.1. Mathematical model 74
3.2.2. Algorithm 75
3.3. Enclaves of operations 76
3.3.1. Mathematical model 77
3.3.2. Algorithm for divisible enclaves 79
3.3.3. Algorithm for indivisible enclaves 81

Page

3.4. Spacial assembly - work fronts 82
3.4.1. Mathematical model 83
3.4.2. Algorithm 84
3.5. Cost of assembly 86
3.5.1. Mathematical model 87
3.5.2. Algorithm 87
3.5.3. Elimination of unperspective states 91

4. ALLOCATION OF OPERATIONS, SEQUENTIAL WORK STATIONS 97

4.1. Introduction 97
4.2. Positional constraints 98
4.2.1. Mathematical model 98
4.2.2. Algorithm 100
4.3. Time constraints 105
4.3.1. Mathematical model 105
4.3.2. Algorithm 106
4.4. Range constraints 109
4.4.1. Mathematical model 109
4.4.2. Algorithm 111
4.5. Asymmetric line 116
4.5.1. Mathematical model 116
4.5.2. Algorithm 118

5. ALLOCATION OF OPERATIONS, PARALLEL WORK STATIONS 122

5.1. Introduction 122
5.2. Basic model 122
5.2.1. Mathematical model 122
5.2.2. Algorithm 125
5.2.3. Elimination of unperspective states 129
5.3. Exclusion relation of operations 135
5.3.1. Mathematical model 135
5.3.2. Algorithm 136
5.4. Synchronization relation of operations 138
5.4.1. Mathematical model 138
5.4.2. Algorithm 139

6. RESOURCE AND OPERATION ALLOCATION 144

6.1. Introduction 144
6.2. Container and operation allocation 145
6.2.1. Mathematical model 145
6.2.2. Algorithm 147
6.2.3. Elimination of unperspective states 150
6.2.4. Multi-criterion optimization 156

Page

6.3. Tool and operation allocation 158
6.3.1. Mathematical model 158
6.3.2. Algorithm 159
6.4. Assembler and operation allocation 164
6.4.1. Mathematical model 164
6.4.2. Algorithm 165

7. FINAL REMARKS 171

APPENDIX. Mathematical models and algorithms for control of dynamic assembly 175

SYMBOLS 175

1. INTRODUCTION 178

2. ASSUMPTIONS 179

3. BASIC DEFINITIONS 180
3.1. State of assembly process 180
3.2. Control of assembly process 183

4. STATE EQUATIONS 185

5. PROBABILISTIC MODEL OF ASSEMBLY 190

6. MODEL OF ASSEMBLY PROCESS AND ASSEMBLY ELEMENTS SUPPLY 194

7. CONTROL OF SINGLE VERSION ASSEMBLY 199
7.1. Static control 200
7.2. Pseudostatic control 201
7.3. Pseudodynamic control 203
7.4. Dynamic control 205

8. MODEL OF MULTI-VERSION ASSEMBLY 207

9. DYNAMIC CONTROL OF MULTI-VERSION ASSEMBLY 210

10. FINAL REMARKS 213

SUMMARY OF THE APPENDIX 214

REFERENCES 216

SUMMARIES OF THE DISSERTATION 232

FIGURES IN DISSERTATION

Fig. 1.1. Assembly line (1 - main conveyor, 2 - assembled object entering the first line station (chassis), 3 - complete assembled object leaving the last line station, 4 - assembled object entering the k-th line station, 5 - places for tools and containers with assembly elements, 6-k-th assembly station)

Fig. 1.2. An economic characteristic of assembly line

Fig. 1.3. A static characteristic of assembly line

Fig. 2.1. Diagram of immediate predecessors and successors

Fig. 2.2. Network of states for binary algorithm

Fig. 2.3. Network of states for sequencing algorithm

Fig. 2.4. Network of states for allocation algorithm

Fig. 2.5. Network of states for scheduling algorithm

Fig. 2.6. Block diagram of algorithm for ALB with selection rule FIFO

FIGURES IN APPENDIX

Fig. D.1. Block diagram for computation of the logical function h

Fig. D.2. General block diagram of computation for deterministic assembly process

Fig. D.3. Block diagram for computation of the logical function h_b, (R - pseudorandom numbers generator)

Fig. D.4. Block diagram for computation of the logical function h_d

Fig. D.5. Block diagram for computation of the cycle-time c^1 and the moment T^1 to start main conveyor

TABLES

Table 2.1. Comparison of state networks for algorithms with and without state domination rules

Table 2.2. Comparison of computations with and without state domination

OD AUTORA

Zagadnienia przedstawione w niniejszej dysertacji wywodzę się z doświadczeń zebranych przez autora w latach 1973-1985 w pracach naukowo-badawczych prowadzonych przez Zakład Automatyzacji Dyskretnych Procesów Przemysłowych w Instytucie Automatyki Politechniki Śląskiej w Gliwicach. Prace te dotyczyły podstaw teoretycznych procesów dyskretnych oraz zastosowań - głównie w przemyśle motoryzacyjnym.

Od momentu ogłoszenia przez M.E. Salvesona w roku 1955 pracy pt.: "The Assembly Line Balancing Problem" liczba publikacji na temat procesu montażu stale rośnie. Aktualność tej problematyki przejawia się dużą różnorodnością rozważanych modeli oraz algorytmów.

Zamierzeniem autora było wprowadzenie klasyfikacji podstawowych problemów sterowania montażem oraz przedstawienie jednolitej metody rozwiązania tych problemów. Z tego względu w dysertacji wyróżniono: standardowy problem balansowania linii montażowej, uogólnione modele balansowania linii montażowej, alokację operacji na linii o szeregowych i równoległych stanowiskach pracy oraz alokację operacji i zasobów. Wszystkie algorytmy rozwiązania tych problemów oparto na metodzie programowania wieloetapowego.

W dysertacji zamieszczono również obszerny dodatek opisujący dynamikę procesu montażu. Wprowadzone w tym dodatku równania stanu montażu tworzą pewne analogie pomiędzy procesami ciągłymi i dyskretnymi. Z teoretycznego punktu widzenia udało się tym sposobem pokazać, że problemy rozważane w części zasadniczej dysertacji należą do sterowania statycznego montażem. Przedstawione modele i algorytmy sterowania dynamicznego montażem nabierają znaczenia w związku z wprowadzaniem na linie montażowe robotów przemysłowych.

Ukończenie pracy nie byłoby możliwe bez ustawicznej zachęty i inspiracji Prof. dra inż. H. Kowalowskiego, dlatego poczuwam się do miłego obowiązku złożenia Mu tą drogą podziękowań. Ostateczną postać pracy zawdzięczam w dużej mierze jej opiniodawcom Prof. drowi inż. H. Góreckiemu i Prof. drowi hab. inż. J. Węglarzowi. Składam Im za to gorące podziękowania. Trud przygotowania maszynopisu pracy wzięła na siebie moja koleżanka mgr inż. E. Król, której również składam wyrazy podziękowania.

Franciszek Marecki

SPIS OZNACZEŃ

Ω - zbiór operacji montażowych

ω - operacja montażowa

K - liczba stacji na linii montażowej

M - liczba stanowisk pracy na linii montażowej (lub liczba typów dodatkowych zasobów)

N - liczba operacji

k - numer stacji montażowej ($k = 1,\ldots,K$)

m - numer stanowiska pracy ($m = 1,\ldots,M$)

n - numer operacji montażowej ($n = 1,\ldots,N$)

C - horyzont procesu montażu

c - cykl (rytm lub takt) montażu

η - numer etapu decyzyjnego ($\eta = 1,\ldots,N$)

l - numer stanu w ramach etapu decyzyjnego

$P^{\eta,l}$ - l-ty stan η-tego etapu

L_η - dopuszczalna liczba stanów η-tego etapu

$P^{0,1}$ - stan początkowy

$P^{N,1}$ - stan końcowy

$P^{\eta-1,\lambda}$ - stan aktywny wybrany do generowania dalszych stanów

P^a - aktualnie najlepszy stan końcowy

$P^{N,1^o}$ - stan lokalnie optymalny

P^o - stan globalnie optymalny

$P^{N,1^p}$ - stan lokalnie Pareto-optymalny

$P^{\bar{p}}$ - stan polioptymalny

$V^{\eta,l}$ - wartość stanu $P^{\eta,l}$

V - funkcja wartości stanu, tzn. $V^{\eta,l} = V(P^{\eta,l})$

$\mathcal{A}_\eta$ - zbiór stanów aktywnych η-tego etapu

$I^{\eta,l}$ - indeks stanu $P^{\eta,l}$

Ω_m - zbiór operacji na m-tym stanowisku pracy

Ω_k - zbiór operacji na k-tej stacji

Ω_m^o - optymalny zbiór operacji na m-tym stanowisku pracy

$\Omega_{k,j}$ - zbiór operacji przydzielonych do j-tego stanowiska na k-tej stacji

$\Omega^{\eta,l}$ - zbiór operacji wykonanych w stanie $P^{\eta,l}$

t_n - chwila zakończenia operacji ω_n

$t_{m,n}$ - chwila zakończenia operacji ω_n wykonanej za pomocą zasobu m-tego typu

$T^{\eta,l}$ - chwila zakończeni. wszystkich operacji ze zbioru $\Omega^{\eta,l}$

$K^{\eta,l}$ - numer ostatniej stacji wykorzystanej w stanie $P^{\eta,l}$

$b^{\eta,l}$ - dolne ograniczenie wartości stanu V^{N,l^o}

b^o - najmniejsze dolne ograniczenie wartości stanu V^o

$\delta_{n,j}$ - j-ty parametr operacji ω_n

$d_{k,j}$ - dopuszczalna suma wartości parametrów j-tego typu dla operacji na k-tej stacji

$D_j^{\eta-1,\lambda}$ - suma wartości parametrów j-tego typu, operacji znajdujących się na $K^{\eta-1,\lambda}$-tej stacji

E - podzbiór operacji nazywany enklawą

e_n - parametr alokacji operacji ω_n do enklawy E

f_n - numer frontu pracy operacji ω_n

$\tau_{i,j}$ - czas przejścia od i-tego do j-tego frontu pracy

j^0 - podstawowy front linii

j^1 - front pracy pierwszej operacji na stacji $K^{\eta-1,\lambda}$ w stanie $P^{\eta-1,\lambda}$

j^2 - front pracy ostatniej operacji na stacji $K^{\eta-1,\lambda}$ w stanie $P^{\eta-1,\lambda}$

w_n - koszt wykonania operacji ω_n

$g_k^{\eta,l}$ - koszt wykonania operacji na k-tej stacji w stanie $P^{\eta,l}$

$v_i^{\eta,l}$ - koszt wykonania operacji na wszystkich stacjach od i do $K^{\eta,l}-1$ w stanie $P^{\eta,l}$

$v_2^{\eta,l}$ - koszt wykonania operacji na stacji $K^{\eta,l}$ w stanie $P^{\eta,l}$

$b_G^{\eta,l}$ - dolne ograniczenie kosztu montażu

$s_{n,j}$ - parametr przyporządkowania operacji ω_n do j-tej pozycji

$s_j^{\eta-1,\lambda}$ - parametr przyporządkowania operacji na $K^{\eta-1,\lambda}$-tej stacji do j-tej pozycji

$q_k^{\eta,l}$ - parametr przydziału stanowiska pracy do k-tej stacji w stanie $P^{\eta,l}$

$\varphi_{i,n}$ - minimalny czas, jaki musi upłynąć od zakończenia operacji ω_i do rozpoczęcia operacji ω_n

$\psi_{i,n}$ - maksymalny czas, jaki może upłynąć od zakończenia operacji ω_i do rozpoczęcia operacji ω_n

$T_{n,\varphi}^{\eta,l}$ - chwila najwcześniejszego rozpoczęcia operacji ω_n w stanie $P^{\eta,l}$

$T_{n,\psi}^{\eta,l}$ - chwila najpóźniejszego rozpoczęcia operacji ω_n w stanie $P^{\eta,l}$

$\vartheta_{m,n}$ - czas wykonania operacji ω_n przy pomocy zasobu m-tego typu

k_n^1 - najmniejszy numer stacji, na której może być wykonana operacja ω_n

k_n^2 - największy numer stacji, na której może być wykonana operacja ω_n

$\vartheta_{n,j}$ - czas wykonania operacji ω_n po j-tej stronie linii(j = 1,2)

$K_j^{\eta,l}$ - numer ostatniej stacji, na której po j-tej stronie linii znajduje się ostatnie stanowisko pracy w stanie $P^{\eta,l}$

$v_j^{\eta,l}$ - liczba stanowisk pracy po j-tej stronie linii w stanie $P^{\eta,l}$

$\Omega_{\gamma,n}$ - zbiór operacji, które są bezpośrednimi poprzednikami operacji ω_n

$\Omega_{\beta,n}^{\eta,l}$ - zbiór operacji, które wykluczają się z operacją ω_n i zostały wykonane w stanie $P^{\eta,l}$

$\Omega_{\alpha,n}$ - zbiór operacji, które muszę być wykonane równocześnie z operacją ω_n

$T_{\gamma,n}^{\eta,l}$ - chwila zakończenia bezpośrednich poprzedników operacji ω_n w stanie $P^{\eta,l}$

$\Gamma_{\nu,n}$ - element macierzy Γ relacji kolejności operacji ω_ν oraz ω_n

$\beta_{\mu,n}$ - element macierzy $\mathcal{B}$ relacji wykluczania operacji ω_μ oraz ω_n

$\alpha_{\mu,n}$ - element macierzy $\mathcal{A}$ relacji synchronizacji operacji ω_μ oraz ω_n

$T_{\beta,n}^{\eta,l}$ - chwila najwcześniejszego rozpoczęcia operacji ω_n w stanie $P^{\eta,l}$ z uwagi na relację wykluczania

z_m - liczba dodatkowych zasobów m-tego typu

$d_{m,n}$ - parametr przyporządkowania zasobu m-tego typu do operacji ω_n

Q_O - wskaźnik kryterialny liczby stanowisk pracy

Q_m - wskaźnik kryterialny liczby zasobów m-tego typu

$q_{m,k}^{\eta,l}$ - parametr przydziału jednostki zasobu m-tego typu do k-tej stacji w stanie $P^{\eta,l}$

ω_n^m - operacja wymagająca zasobu m-tego typu

δ_j - tolerancja bezwzględna dla wskaźnika kryterialnego Q_j

$w_{?.1}$ - zbiór monterów wykorzystanych w stanie $p_{?.1}$

1. WPROWADZENIE

1.1. WSTĘP

W dziedzinie automatyzacji procesów przemysłowych wyróżnia się procesy ciągłe oraz kompleksy operacji [33, 35]. Wyróżnienie to wynika z charakteru procesu rzeczywistego lub poziomu abstrakcji opisu matematycznego. Z punktu widzenia przepływu sygnału lub medium można mówić o systemach wejściowo-wyjściowych. Kompleksy operacji stanowią zbiory operacji z ograniczeniami czasowymi. Rozróżnienie systemów wejściowo-wyjściowych i kompleksów operacji wprowadza się z punktu widzenia powiązań między ich podsystemami. Do opisu matematycznego procesów ciągłych wykorzystywane są równania różniczkowe. W kompleksach operacji model matematyczny pojedynczej operacji może być przedstawiony w postaci równań różniczkowych. Problemy powiązań pomiędzy operacjami są rozwiązywane za pomocą metod badań operacyjnych - głównie programowania matematycznego. Obszerną klasę stanowią problemy, w których operacje są opisane przez czas ich wykonania.

Kompleks operacji z ograniczeniami: logicznymi, czasowymi i przestrzennymi, realizowany w systemie wejściowo-wyjściowym (w sensie przepływu obiektów), nazywany jest dyskretnym procesem przemysłowym [146]. Przez system rozumiany jest zbiór agregatów (maszyn itp.), które łączy wspólny cel działania, określony przez odpowiednie kryterium lub kryteria [68]. W dyskretnym procesie przemysłowym pomiędzy agregatami przepływają obiekty. Wejścia i wyjścia takiego systemu są wyznaczone przez dopuszczalne marszruty obiektów. W trakcie procesu obiekty mogą być dzielone lub łączone. W dyskretnym procesie przemysłowym wyróżnia się operacje transportowe (przemieszczanie obiektów między agregatami) oraz operacje technologiczne (obróbki obiektów w agregatach). Istotne znaczenie mają magazyny buforowe pomiędzy agregatami [214]. Do realizacji procesu oprócz agregatów wykorzystywane są dodatkowe zasoby.

Problematyka automatyzacji dyskretnych procesów przemysłowych obejmuje: identyfikację, modele matematyczne i algorytmy, systemy komputerowe oraz elementy wykonawcze, którymi są operatorzy lub roboty przemysłowe [11, 12, 13, 14]. Identyfikacji podlegają parametry operacji oraz struktura kompleksu [34]. Z uwagi na dużą różnorodność procesów dyskretnych, formułowanie adekwatnych modeli matematycznych jest równie istotne jak dla procesów ciągłych [62, 139, 140]. Modele te dotyczą głównie alokacji zadań i zasobów [26] oraz harmonogramowania [233, 214]. Komputerowe systemy automatyzacji dyskretnych procesów przemysłowych charakteryzuję się

dużą liczbą danych zbieranych sporadycznie [230]. Roboty przemysłowe stosowane jako elementy wykonawcze zapewniają elastyczną automatyzację dyskretnych procesów przemysłowych [15, 213, 143, 146, 147].

Do opisu dyskretnych procesów przemysłowych stosowane są różne podejścia oparte na teoriach: automatów [64, 65, 66, 67], stanu [327, 261, 28], grafów [1, 19, 63, 70, 71] oraz wieloetapowych procesów decyzyjnych [76, 78, 193, 200]. W problemach optymalizacyjnych istotne znaczenie ma złożoność obliczeniowa algorytmów [41, 318, 319]. Dla problemów nie należących do klasy P szerokie zastosowanie znajdują algorytmy: przeglądowe [266], heurystyczne [240] oraz symulacyjne [311]. Algorytmy dokładne oparte na programowaniu dynamicznym [20, 21, 47, 76, 78] oraz na binarnym programowaniu całkowitoliczbowym [17, 18, 58, 157] nie znajdują zastosowania w praktyce z uwagi na duże zapotrzebowanie na pamięć komputerową. W obliczeniach istotne znaczenie dla efektywności algorytmów mają komputerowe struktury danych [75, 322, 324]. Modyfikacje wprowadzane w algorytmach dokładnych [86, 285] mają na celu uzyskanie bardziej efektywnego przeglądu rozwiązań, poprzez odrzucanie gałęzi (trajektorii) nieperspektywicznych. Skraca to czas obliczeń i zmniejsza zajętość pamięci komputera.

W złożonych problemach kombinatorycznych preferowane są algorytmy przeglądowe typu podziału i ograniczeń [155, 137, 57, 136]. W konwencjonalnym ujęciu algorytmy takie są oparte na teorii grafów. Pozwalają one uzyskać rozwiązanie dokładne, a w przypadku limitowanego czasu obliczeń - rozwiązanie przybliżone z oszacowaniem dokładności. W niniejszej rozprawie metoda podziału i ograniczeń zostanie przedstawiona w sposób niekonwencjonalny, w oparciu o język wieloetapowych procesów decyzyjnych [193, 199, 200]. Ten sposób rozwiązywania problemów kombinatorycznych nazwiemy programowaniem wieloetapowym. Optymalizacja dyskretnego procesu przemysłowego może być rozważana jako przegląd dopuszczalnych wariantów przebiegu procesu. Symulacja każdego wariantu jest rezultatem realizacji pewnej strategii (ciągu decyzji) wieloetapowego procesu decyzyjnego. A zatem język wieloetapowych procesów decyzyjnych daje dobrą interpretację algorytmów optymalizacji dyskretnych procesów przemysłowych. W tym sensie algorytmy heurystyczne lub symulacyjne generują tylko jedną trajektorię wieloetapowego procesu decyzyjnego. Stąd też przejście od algorytmów heurystycznych lub symulacyjnych do algorytmów programowania wieloetapowego jest naturalne [146]. Stosowanie algorytmów przeglądowych z oszacowaniem dokładności rozwiązania przybliżonego jest uzasadnione kilkoma względami. Po pierwsze w złożonych problemach kombinatorycznych czas potrzebny na dojście do rozwiązania dokładnego często przekracza dopuszczalny limit. Ponadto dane liczbowe nie są dokładne, stąd prawdopodobieństwo realizacji wyznaczonego rozwiązania jest mniejsze od 1 [109, 110, 111, 112, 113, 114]. Wreszcie problemy kombinatoryczne z dodatkowymi zasobami są zwykle wielokryterialne [267, 268]. W takich przypadkach wyznacza się zbiór rozwiązań niezdominowanych (Pareto-optymalnych [4, 6, 238]) lub rozwiązanie

polioptymalne, które stanowi kompromis oparty na subiektywnych preferencjach decydenta [2, 5, 7, 69, 267, 268]. Z powyższych względów preferowane są algorytmy przeglądowe typu podziału i ograniczeń pozwalające wyznaczyć rozwiązanie dokładne lub przybliżone z oszacowaniem dokładności.

Przedmiotem rozważań niniejszej rozprawy będzie montaż jako wybrany dyskretny proces przemysłowy. Specyfika procesu montażu jest zależna od systemu, w którym jest on realizowany (por. np. [241] i [244]). Z tego względu rozważania dotyczyć będą linii montażowej. Linia montażowa jest systemem, którego agregatami są stacje montażowe.

Genezę niniejszych rozważań są prace naukowo-badawcze prowadzone przez Instytut Automatyki Politechniki Śląskiej w Gliwicach dla przemysłu motoryzacyjnego, w latach 1974-82 [141, 142, 145]. Problematyka tych prac obejmowała: alokację operacji i zasobów, harmonogramowanie oraz sterowanie statyczne i dynamiczne montażem [194]. W niniejszej rozprawie uwaga zostanie skoncentrowana na alokacji operacji i zasobów.

W ramach alokacji operacji i zasobów zostaną przedstawione deterministyczne modele i algorytmy:

- balansowania linii montażowej (BLM)[1)],
- alokacji operacji (AO) do szeregowych oraz równoległych stanowisk pracy,
- równoczesnej alokacji operacji i zasobów (AOiZ) do stanowisk pracy.

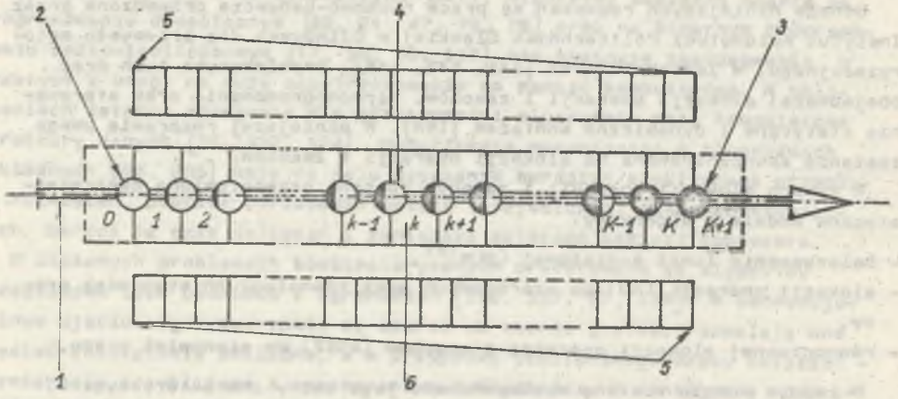
W każdym modelu zostaną wyeksponowane jego cechy charakterystyczne. Z tego względu, zamiast modelu uniwersalnego zostaną przedstawione wyspecyfikowane problemy, mające odzwierciedlenie w praktyce produkcyjnej. Wprowadzona klasyfikacja tych problemów wynika z ich odmienności (w ramach rozdziału) oraz stopnia ogólności (w kolejnych rozdziałach). Przedstawione w rozprawie problemy AOiZ stanowią rozwinięcie i uogólnienie analizowanego w literaturze [26, 278] problemu BLM. Problem BLM był rozwiązywany w literaturze za pomocą różnych metod programowania matematycznego. W rozprawie wszystkie problemy AOiZ będą rozwiązywane za pomocą metody programowania wieloetapowego. Z tego względu nieco dalej we wprowadzeniu zostanie przedstawiona koncepcja metody programowania wieloetapowego.

Literatura przytoczona w pracy wywodzi się z dwóch dziedzin: procesu montażu na liniach oraz programowania wieloetapowego. Dla linii montażowych starano się wskazać pozycje źródłowe (w tym prace doktorskie), celem podkreślenia wagi problematyki. Publikacje przedstawiające zastosowania metody programowania wieloetapowego przytoczono z uwagi na fakt, że wykorzystanie języka wieloetapowych procesów decyzyjnych w metodzie podziału i ograniczeń stanowi podejście niekonwencjonalne.

[1)] Stosowane są również inne terminy, np. bilansowanie, równoważenie lub synchronizacja.

1.2. OPIS LINII I PROCESU MONTAŻU

Przedmiotem rozważań w rozprawie będzie proces montażu realizowany na linii pokazanej na rys. 1.1. Linia montażowa składa się z transportera głównego, stacji montażowych oraz pól, na których znajduję się narzędzia i kontenery z detalami. Transporter główny przenosi montowany obiekt wzdłuż linii. Chassis montowanych obiektów są instalowane w równych odstępach na transporterze głównym. Transporter główny porusza się jednostajnie lub skokowo. Stacje montażowe są wyróżnionymi polami usytuowanymi szeregowo wzdłuż linii. Na stacjach montażowych lokalizowane są stanowiska pracy. Stanowisko pracy jest miejscem na stacji montażowej, w którym znajduje się jeden operator (monter lub robot przemysłowy) w trakcie wykonywania montażu. Detal jest częścią montowanego obiektu. Narzędzia oraz kontenery z detalami znajdują się na polach rozlokowanych po obu stronach transportera głównego. Założymy, że długość stacji montażowej jest równa odstępowi pomiędzy punktami instalacji dwóch sąsiednich obiektów na transporterze głównym. Czas pobytu montowanego obiektu na stacji montażowej nazwiemy cyklem (rytmem lub taktem). W trakcie cyklu na każdej stacji znajduje się jeden obiekt.



Rys. 1.1. Linia montażowa

1 - transporter główny, 2 - montowany obiekt wchodzący na pierwszę stację linii (chassis montowanego obiektu), 3 - zmontowany kompletnie obiekt schodzący z ostatniej stacji linii, 4 - montowany obiekt wchodzący na k-tę stację, 5 - pola dla narzędzi i kontenerów z detalami, 6 - k-ta stacja montażowa

Fig. 1.1. Assembly line

1 - main conveyor, 2 - assembled object entering the first line station (chassis), 3 - complete assembled object leaving the last line station, 4 - assembled object entering the k-th line station, 5 - places for tools and containers with assembly elements, 6 - k-th assembly station

Jeżeli transporter główny porusza się skokowo, to operatorzy pracuję w czasie postoju transportera, a obiekt znajduje się na środku stacji montażowej. Po wykonaniu wszystkich operacji na każdej stacji montażowej transporter przesuwa się o jedną stację. Przy takich uwarunkowaniach operatorami są zwykle roboty przemysłowe.

W przypadku gdy transporter główny porusza się jednostajnie, operatorami są zwykle monterzy. Obiekt pojawia się na stacji na początku cyklu i od tej chwili jest dostępny dla montera. Monter obsługuje obiekt poruszając się z nim w ramach stanowiska pracy. W chwili gdy mija cykl, obiekt przestaje być dostępny dla montera, ponieważ wchodzi na kolejnę stację montażową. Równocześnie monter może rozpocząć obsługę kolejnego obiektu.

W rozprawie przedmiotem rozważań będzie proces montażu w przedziale czasu $[0,C]$, gdzie C jest wielokrotnością cyklu c. Warunek początkowy procesu montażu jest określony przez zaawansowanie montażu obiektów na poszczególnych stacjach w chwili t=0. Analogicznie warunek końcowy jest określony przez zaawansowanie procesu montażu obiektów na poszczególnych stacjach w chwili t=C. Z uwagi na to, że operacje są niepodzielne, w procesie montażu istotne znaczenie mają jedynie chwile rozpoczęcia i zakończenia operacji. W ten sposób w dyskretnym procesie przemysłowym wyróżnia się stany istotne [50, 101, 102, 103, 165, 299, 300, 301, 302, 328]. W rozprawie proces montażu będzie analizowany z dokładnością do stanów istotnych.

Jeżeli na każdej stacji montażowej znajduje się co najwyżej jedno stanowisko pracy, to proces montażu jest sekwencyjny, a linia składa się z szeregowych stanowisk pracy. W przypadku przeciwnym mówimy o linii z równoległymi stanowiskami pracy. W rozprawie przyjmiemy, że na każdej stacji montażowej mogą się znajdować co najwyżej dwa stanowiska pracy (po różnych stronach transportera głównego). Jeżeli parametry (np. czas) wykonania operacji po różnych stronach transportera głównego są takie same, to linia jest symetryczna. W przypadku przeciwnym linia jest asymetryczna. Proces montażu może mieć charakter przestrzenny, jeżeli nie można pominąć czasów przejścia między punktami lokalizacji detali w montowanym obiekcie. W przypadku przeciwnym proces montażu jest punktowy.

Na liniach montażowych wyróżnia się fragmenty zwane strefami, w których wykonywane są określone podzbiory operacji. Strefy są związane wspólnym transporterem głównym. Z tego względu rozróżnia się linie: jednostrefowe oraz wielostrefowe. Ponadto poza linię mogą znajdować się stanowiska podmontażowe. Na stanowiskach tych montowane są zespoły, które na linii są traktowane jako detale. W przedziale czasu $[0,C]$ zespół może być montowany wyłącznie na linii lub na stanowisku podmontażowym. Jeżeli zespół jest montowany na linii, to liczba operacji montażu obiektu zwiększa się. Proces montażu może być realizowany na linii bez lub ze stanowiskami podmontażowymi. Linie wielostrefowe oraz linie ze stanowiskami podmontażowymi były przedstawione w pracach [166, 167, 171, 194].

W procesie montażu operatorzy pracują cyklicznie na swych stanowiskach. Równocześnie w trakcie przepływu obiektów przez kolejne stacje montażowe liczba zamontowanych detali rośnie - co na rys. 1.1 przedstawiono symbolicznie w postaci kół o narastającym stopniu wypełnienia. Jak widać z rys. 1.1, przyjmujemy warunki brzegowe procesu montażu, zgodnie z którymi:

- do linii są dostarczane tylko chassis montowanego obiektu,
- z linii schodzą obiekty zmontowane kompletnie.

A zatem na stanowiskach pracy linii muszą być zamontowane wszystkie detale. Czynność zamontowywania detalu nazwiemy operacją. Podstawowym parametrem operacji jest czas jej trwania. W rozprawie przyjmuje się deterministyczne czasy operacji. Cechę charakterystyczną procesu montażu jest relacja kolejności operacji, którą przedstawia się w postaci macierzy (lub grafu) bezpośrednich poprzedników i następników.

Stacje montażowe stanowią podstawowy zasób procesu montażu (agregaty linii montażowej). Zasobami dodatkowymi są operatorzy, narzędzia oraz kontenery z detalami. Monterzy nie są uniwersalni, tzn. mają kwalifikacje do wykonania określonych operacji. Niektóre operacje mogą być wykonywane przez różnych monterów. A zatem monterzy stanowią zasób substytucyjny. Należy przy tym podkreślić, że monter może pracować na stanowisku pracy, jeżeli ma kwalifikacje do wykonania każdej operacji przydzielonej do tego stanowiska. Do wykonania operacji wymagane jest narzędzie określonego typu. Niektóre operacje mogą być wykonywane narzędziami różnych typów. Tym samym narzędzia stanowią zasób substytucyjny. Na każdym stanowisku pracy wymagane jest co najwyżej jedno narzędzie określonego typu. Detale wykorzystywane w montażu obiektów znajdują się w kontenerach (lub transporterach specjalnych) określonego typu. Przyporządkowanie detalu do kontenera (transportera specjalnego) jest jednoznaczne, a zatem kontenery (transportery specjalne) stanowią zasób niesubstytucyjny.

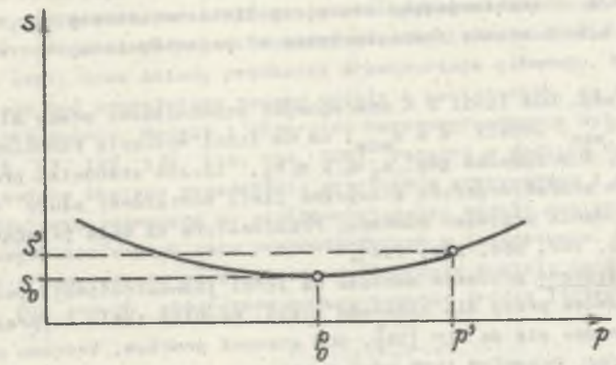
W procesie montażu wyróżnia się wersje obiektów określone przez zbiory detali, z których składa się obiekt. W jednowersyjnym procesie montażu każdy obiekt składa się z tych samych detali. W montażu wielowersyjnym obiekty montowane na linii mogą składać się z różnych detali. Jeżeli wszystkie obiekty tej samej wersji tworzą na linii sekwencję, która nie zawiera obiektów innych wersji, to mówimy o montażu seryjnym. Seria jest sekwencją obiektów jednakowej wersji. W rozprawie przyjmiemy, że liczba obiektów w serii jest większa niż liczba stacji na linii montażowej. W ten sposób na linii mogą znajdować się obiekty co najwyżej dwóch wersji. Jeżeli obiekty pewnej wersji nie tworzą serii, to montaż jest mieszany. W rozprawie będzie analizowany szczegółowo jednowersyjny proces montażu.

Jeżeli w każdym cyklu przedziału $[0,C]$ na stanowiskach pracy linii wykonywane są te same podzbiory operacji, to proces montażu jest statyczny. W przypadku przeciwnym montaż jest dynamiczny. Szczególnym przypadkiem

montażu dynamicznego jest proces przejściowy, w którym proces statyczny następuje po K cyklach. Proces przejściowy występuje przy zmianie wersji obiektów montowanych seryjnie na linii. Opis dynamicznego procesu montażu zostanie przedstawiony w dodatku do rozprawy.

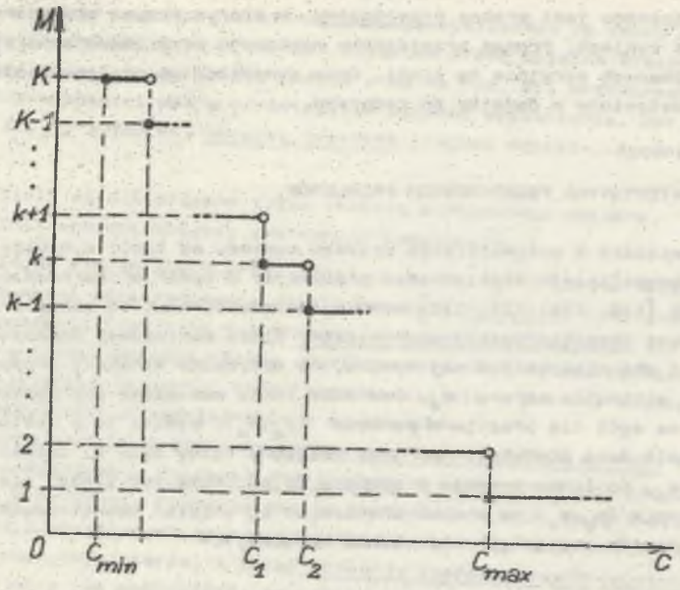
1.3. CHARAKTERYSTYKA PODSTAWOWYCH PROBLEMÓW

Problemy związane z optymalizacją procesu montażu na linii wynikają z dążenia do maksymalizacji efektywności produkcji. Z badań przeprowadzonych w pracach [123, 124, 127, 310] wynika, że efektywność tę można przedstawić za pomocą charakterystyki ekonomicznej linii montażowej pokazanej na rys. 1.2. Z charakterystyki tej wynika, że optymalna wielkość produkcji p_o daje minimalne straty s_o. Jednakże linia montażowa z przyczyn zewnętrznych na ogół nie pracuje w punkcie (p_o, s_o). Wynika to z faktu, że z różną wielkością produkcji p jest związany różny zysk z. Jeżeli $z - z_o > s - s_o$, to linia pracuje w punkcie (p, s). Poza tym linia nie pracuje w punkcie (p_o, s_o) na skutek chwilowych ograniczeń wewnętrznych (np. zbiór monterów dostępnych na zmianie roboczej).



Rys. 1.2. Charakterystyka ekonomiczna linii montażowej
Fig. 1.2. An economic characteristic of assembly line

Pomiędzy produkcją p oraz cyklem c istnieje prosta zależność $c = 1/p$. A zatem wraz z wyborem punktu pracy linii montażowej powstaje problem wyznaczenia podzbiorów operacji, które powinny być wykonywane na stanowisku pracy. Z tego względu w przedziale $[0,C]$ można korzystać z charakterystyki statycznej linii montażowej [148, 279, 165, 167, 168, 169]. Charakterystykę taką pokazano na rys. 1.3. Zauważmy, że liczba stanowisk pracy jest funkcją schodkową zależną od cyklu. Dla cyklu $c \geq c_{max}$ (c_{max} jest sumą czasów wszystkich operacji montażowych) na linii wystarcza jed-



Rys. 1.3. Charakterystyka statyczna linii montażowej
Fig. 1.3. A static characteristic of assembly line

no stanowisko pracy. Dla linii z K szeregowymi stanowiskami pracy minimalny cykl wynosi c_{min}. Jeżeli $c < c_{min}$, to na linii wystąpią równoległe stanowiska pracy. W przypadku gdy $c_1 \leqslant c \leqslant c_2$, liczba stanowisk pracy wynosi k. A zatem charakterystyka statyczna linii montażowej służy do statycznego sterowania procesem montażu. Problematyka ta była przedstawiona w pracach [148, 167, 168, 169, 194].

Sterowanie statyczne procesem montażu na linii jednostrefowej wymaga wyznaczenia stanowisk pracy dla zadanego cyklu. Problem ten w najprostszym ujęciu sprowadza się do BLM [26]. BLM stanowi problem, któremu poświęcono wiele prac. Przegląd tych prac zostanie przedstawiony w punkcie 1.4.

Klasyczny model BLM uwzględnia jedynie: zbiór operacji z relacją kolejności oraz czasy operacji (przy zadanym cyklu). Rozwiązanie tego problemu za pomocą różnych algorytmów programowania wieloetapowego zostanie przedstawione w rozdziale 2 rozprawy.

W procesie montażu występują często dodatkowe parametry, np. koszt operacji, fronty pracy lub enklawy operacji. Uwzględniając te parametry można sformułować uogólnione modele BLM. W modelach tych nie rozróżnia się stanowisk pracy i stacji montażowych. Uogólnione modele BLM zostaną przedstawione w rozdziale 3 rozprawy.

Rezultatem rozwiązania problemu BLM jest minimalna liczba podzbiorów operacji. Każdy podzbiór operacji tworzy jedno stanowisko pracy. Przyjmuje się przy tym, że nie ma ograniczeń lokalizacji operacji do stacji montażowych. W praktyce takie ograniczenia istnieją i mają istotne znaczenie. Są to ograniczenia: pozycyjne, zakresowe oraz czasowe. Istota tych ograniczeń powoduje przydzielanie operacji bezpośrednio do stacji montażowych. W ten sposób na pewnych stacjach zostają zlokalizowane stanowiska pracy. Odpowiednie modele AO do szeregowych stanowisk pracy zostaną przedstawione w rozdziale 4 rozprawy.

Jeżeli wymuszony cykl montażu jest odpowiednio krótki, to na linii występują równoległe stanowiska pracy. Cechą charakterystyczną procesu montażu z równoległymi stanowiskami pracy są relacje: kolejności, wykluczania oraz synchronizacji operacji. Odpowiednie modele AO na liniach z równoległymi stanowiskami pracy zostaną przedstawione w rozdziale 5 rozprawy.

W problemach BLM oraz AO przyjmuje się, że dodatkowe zasoby nie wnoszą żadnych ograniczeń. W wielu przypadkach ograniczenia takie istnieją. Wynikają one z kwalifikacji monterów oraz liczby narzędzi i kontenerów określonego typu. Modele równoczesnej AOiZ zostaną przedstawione w rozdziale 6 rozprawy.

W przypadku montażu seryjnego istotne znaczenie ma harmonogramowanie. Celem harmonogramowania jest wyznaczenie chwil: rozpoczęcia i zakończenia montażu serii oraz zmiany prędkości transportera głównego. W trakcie montażu muszą być uzupełnione zapasy detali w kontenerach, co również wymaga harmonogramowania. Modele i algorytmy harmonogramowania były przedstawione w [16, 93, 145, 146, 154, 194, 205]. Ponadto w dodatku do rozprawy przedstawione zostaną zagadnienia sterowania statycznego i dynamicznego.

W rozprawie rozważane są niekonwencjonalne modele montażu, które stanowią uogólnienie problemów przedstawionych w literaturze. Wszystkie algorytmy przedstawione szczegółowo w rozprawie zostały opracowane przez autora. Dla innych algorytmów podano jedynie źródło literaturowe.

1.4. PROBLEM BLM

Problem BLM został po raz pierwszy opublikowany przez M.E. Salvesona w [258, 259], aczkolwiek niepublikowaną pracę z tej dziedziny wykonał wcześniej B. Bryton [31]. Problem ten polega na pogrupowaniu operacji montażowych w dopuszczalne podzbiory, które tworzą stanowiska pracy. Przyjmuje się, że dany jest zbiór operacji z relacją kolejności oraz czasy operacji. Wyróżnia się dwa warianty BLM:

- dla danego cyklu należy wyznaczyć minimalną liczbę podzbiorów operacji (stanowisk pracy),
- dla danej liczby stanowisk pracy należy wyznaczyć minimalny cykl.

Problem BLM jest NP-zupełny w sensie złożoności obliczeniowej [158]. Wyróżnione warianty BLM są w pewnym sensie dualne [278]. Rozwiązując wariant pierwszy wielokrotnie dla różnych cykli, można uzyskać rozwiązanie dla wariantu drugiego.

Do rozwiązania problemu BLM opublikowano dotąd wiele algorytmów [26, 278]. Prawie wszystkie algorytmy dokładne (z wyjątkiem [42]) dotyczą pierwszego wariantu BLM. Niektóre algorytmy (np. [116]) uwzględniają losowe czasy operacji montażowych. Warto podkreślić, że w problemie BLM (dla wariantu pierwszego) należy wyznaczyć nie tylko minimalną liczbę podzbiorów operacji, ale również operacje należące do tych podzbiorów. Tak rozumiany problem BLM jest formą sterowania procesem montażu [26].

W procesie montażu występuje wiele ograniczeń, co prowadzi do różnorodnych modeli. Z tego względu porównywanie algorytmów dla różnorodnych modeli jest utrudnione. Do rozwiązania problemów BLM z wieloma ograniczeniami stosowane były dotąd algorytmy heurystyczne. Algorytmy dokładne stosowano dla modelu BLM bez dodatkowych ograniczeń.

1.4.1. Klasyczny problem BLM

Klasycznym problemem BLM nazwiemy model bez dodatkowych ograniczeń, który można sformułować następująco:

Dany jest zbiór operacji montażowych:

$$\Omega = \left\{ \omega_n \right\}_{n=1,\dots,N} \tag{1.1}$$

gdzie: ω_n - n-ta operacja, N - liczba operacji.
Relacje kolejności operacji jest opisana macierzą:

$$\Gamma = \left[\gamma_{\nu,n} \right]_{\substack{\nu=1,\dots,N \\ n=1,\dots,N}} \tag{1.2}$$

Elementy tej macierzy są określone następująco:

$$\gamma_{\nu,n} = \begin{cases} 1; \text{ jeśli operacja } \omega_\nu \text{ jest bezpośrednim poprzednikiem operacji } \omega_n, \\ 0; \text{ w przypadku przeciwnym.} \end{cases} \tag{1.2a}$$

Czasy operacji dane są wektorem:

$$\Theta = \left[\vartheta_n \right] \tag{1.3}$$

gdzie: ϑ_n - czas operacji ω_n.

Cykl c procesu montażu spełnia warunek:

$$\max_{1 \le n \le N} \vartheta_n \le c < \sum_{n=1}^{n=N} \vartheta_n \tag{1.4}$$

Oznaczmy przez Ω_m ($m = 1,\dots,M$), m-ty kolejny podzbiór operacji tworzący stanowisko pracy (gdzie: M - nieznana liczba podzbiorów operacji).

Dopuszczalny balans linii musi spełniać następujące ograniczenia:

$$\bigcup_{m=1}^{m=M} \Omega_m = \Omega \tag{1.5}$$

$$\Omega_\mu \cap \Omega_m = 0 \qquad \mu \neq m \tag{1.6}$$

$$\bigvee_{1 \le m \le M} \sum_{\omega_n \in \Omega_m} \vartheta_n \le c \tag{1.7}$$

$$\bigvee_{\nu} \bigvee_{n} \exists_{\mu \le m} (\gamma_{\nu,n}=1) \wedge (\omega_\nu \in \Omega_\mu) \Rightarrow (\omega_n \in \Omega_m) \tag{1.8}$$

Optymalny balans linii musi spełniać kryterium minimalizacji nie wykorzystanego czasu pracy:

$$Q = \sum_{m=1}^{m=M} \left(c - \sum_{\omega_n \in \Omega_m} \vartheta_n \right) \longrightarrow \min \tag{1.9a}$$

lub w innej postaci:

$$Q = M \cdot c - \sum_{n=1}^{n=N} \vartheta_n \longrightarrow \min \tag{1.9b}$$

Jak widać z (1.9), Q jest funkcją dwóch parametrów: c oraz M. A zatem jeśli c=const, to należy minimalizować M, natomiast gdy M=const, to minimalizujemy c. W ten sposób można wyznaczyć charakterystykę statyczną linii montażowej pokazaną na rys. 1.3. Warto podkreślić, że minimalizacja nie wykorzystanego czasu pracy jest równoważna minimalizacji liczby stanowisk pracy.

1.4.2. Algorytmy BLM

Dla rozwiązania problemu BLM opublikowano dotąd wiele algorytmów, które można podzielić na dokładne i heurystyczne. Algorytmy te dotyczą montażu jednowersyjnego (z wyjątkiem [289, 290, 291, 292]) o zdeterminowanych danych (z wyjątkiem [116]). Za pomocą algorytmów dokładnych rozwiązywany był klasyczny problem BLM, natomiast do rozwiązywania problemów BLM z dodatkowymi ograniczeniami wykorzystywano algorytmy heurystyczne. Jedynie w komentarzach do algorytmów dokładnych sugerowano możliwość uwzględnienia dodatkowych ograniczeń, które nazwano (niejednoznacznie) - strefowymi. Przegląd algorytmów BLM zamieszczono w uzupełniających się pracach [26, 278]. Algorytmy te porównywano z uwagi na wymagany obszar pamięci komputerowej (liczba zmiennych i ograniczeń lub generowana sieć) oraz czas obliczeń.

A l g o r y t m y d o k ł a d n e

Algorytmy dokładne są oparte na metodach: liniowego programowania dyskretnego, programowania sieciowego, programowania dynamicznego oraz na metodzie podziału i ograniczeń.

W algorytmach liniowego programowania dyskretnego [157, 57, 137] model matematyczny BLM jest przedstawiony za pomocą zmiennej binarnej [30, 317, 134, 313, 287, 288, 236]. Zasadniczy problem w tych modelach stanowi duża liczba zmiennych i ograniczeń. Z tego względu w kolejnych algorytmach BLM opartych na liniowym programowaniu dyskretnym przedstawiono różne sposoby redukcji zmiennych. Algorytmy te są oparte na metodzie przeglądowej [17, 18, 58]. W ostatniej pracy z tej dziedziny [284] wprowadzono zmienne dyskretne (niebinarne) oraz wykorzystano metodę cięć [285] dla zwiększenia efektywności algorytmu.

Algorytmy BLM oparte na metodzie programowania dynamicznego [20, 21] są zróżnicowane. Dla wariantu pierwszego BLM (c=const) można wyróżnić prace [89, 90, 77]. W pracy [42] przedstawiono algorytm dla wariantu drugiego BLM (M=const). Algorytmy programowania dynamicznego wymagają dużego obszaru pamięci komputerowej do zapamiętania generowanej sieci.

Metoda programowania sieciowego była wykorzystywana w algorytmach BLM przedstawionych w pracach [99, 134, 72, 73]. Ponadto w [162, 45, 46] przedstawiono modyfikacje algorytmu z pracy [73]. Modyfikacje te wymagają oszacowania poszukiwanej liczby stanowisk pracy. Algorytmy programowania sieciowego bez modyfikacji wymagają dużej pamięci komputera. Efektywność algorytmów programowania sieciowego z modyfikacją zależy od oszacowania liczby stanowisk pracy.

BLM metodą podziału i ograniczeń przedstawiono w pracach [91, 160, 161, 162, 220, 226, 229, 159]. Algorytmy prezentowane w tych pracach cechuje różny stopień skomplikowania i różna efektywność. Na podstawie przeprowadzonych testów komputerowych można uznać, że algorytm przedstawiony w pracy [159] jest najbardziej efektywnym algorytmem BLM. Algorytmy metody podziału i ograniczeń wymagają niewielkiej pamięci komputerowej. Ich efektywność jest zależne od sposobu generowania dendrytu decyzyjnego oraz eliminacji gałęzi nieperspektywicznych.

W rozprawie do rozwiązania problemów BLM, AO oraz AOiZ zostanie wykorzystana metoda programowania wieloetapowego, która jest niekonwencjonalnym wariantem metody podziału i ograniczeń. Algorytmy programowania wieloetapowego dla problemów BLM przedstawiono w pracach [182, 187, 189, 196, 197, 201, 203, 206, 207, 208, 209, 210, 211, 212, 213, 247].

A l g o r y t m y h e u r y s t y c z n e

Algorytmy heurystyczne cechuje duża różnorodność. Ogólnie można je podzielić na algorytmy: szeregowania oraz przydziału. Do pierwszej grupy należą algorytmy, które pozwalają wyznaczyć bezpośrednio dopuszczalną sekwencję wykonywania operacji. Sekwencję tę określa się na podstawie heurystycznych reguł. Algorytmy szeregowania przedstawiono w pracach [8, 9, 10, 79, 80, 81, 297]. Do drugiej grupy należą algorytmy, które pozwalają wyznaczyć bezpośrednio dopuszczalne podzbiory operacji na stanowiskach pracy. Podzbiory te wyznacza się na podstawie reguł heurystycznych. Algorytmy przydziału przedstawiono w pracach [82, 83, 84, 121, 122, 126, 296, 315, 316]. Algorytmy szeregowania zaleca się, gdy cykl jest krótki względem czasów operacji, tzn. liczba operacji na stanowiskach pracy nie jest duża. W przeciwnym przypadku lepsze rezultaty dają algorytmy przydziału, ponieważ kolejność operacji niezależnych może być dowolna.

1.5. METODA PROGRAMOWANIA WIELOETAPOWEGO

Do rozwiązania wielu problemów kombinatorycznych (np. alokacji zadań i zasobów oraz harmonogramowania) wykorzystywane są algorytmy przeglądowe typu podziału i ograniczeń [155, 26, 92, 137, 57, 136]. Konwencjonalny sposób przedstawienie tych algorytmów bazuje na teorii grafów [1, 70, 71, 63]. Efektywność tych algorytmów zależy od sposobu przeszukiwania drzewa decyzyjnego [2, 86]. W niniejszej rozprawie do przedstawienia algorytmów podziału i ograniczeń zostanie wykorzystany język wieloetapowych procesów decyzyjnych [20, 27]. Podejście takie, nazwane programowaniem wieloetapowym, zostało zaproponowane w [193, 200, 199].

Podstawowymi elementami konstrukcyjnymi algorytmów programowania wieloetapowego są: sten procesu decyzyjnego, wartość stanu, procedury generowania stanów oraz reguły eliminowania stanów nieperspektywicznych. Idea algorytmu polega na generowaniu dopuszczalnych trajektorii i eliminowaniu trajektorii nieperspektywicznych, które nie prowadzą do rozwiązania optymalnego.

Oznaczmy przez η numer etapu decyzyjnego, $\eta = 0,\ldots,N$. Przez l oznaczamy numer stanu w ramach η-tego etapu, $l=1,\ldots,L_\eta$, gdzie: L_η - dopuszczalna liczba stanów η-tego etapu.

Określenie 1.1

Stan P interpretuje sytuację, jaka powstaje w systemie po podjęciu decyzji. W ogólnym przypadku stan jest macierzą [200].

Stan początkowy $P^{0,1}$ określa warunki początkowe procesu w systemie.

Stan końcowy $P^{N,1}$ każdej trajektorii daje dopuszczalne rozwiązanie problemu.

Stan aktywny $P^{\eta,l}$ (perspektywiczny) pozwala wygenerować dalsze stany.

Stan wybrany $P^{\eta-1,\lambda}$ jest stanem aktywnym, który został wybrany dla generowania dalszych stanów.

Stan wygenerowany P jest stanem otrzymanym ze stanu wybranego $P^{\eta-1,\lambda}$. Stan P jest testowany z uwagi na perspektywiczność. Jeżeli P okaże się stanem perspektywicznym, to zostaje stanem aktywnym. W przypadku przeciwnym stan ten zostaje wyeliminowany z dalszych obliczeń.

Stan wyczerpany P^w jest stanem wygenerowanym P, z którego nie można otrzymać żadnego stanu końcowego.

Stan aktualnie najlepszy P^a jest najlepszym stanem końcowym wyznaczonym w ograniczonym czasie obliczeń.

Stan lokalnie optymalny P^{N,l^o} jest najlepszym stanem końcowym, który można otrzymać ze stanu $P^{\eta,l}$.

Stan globalnie optymalny P^o jest najlepszym stanem końcowym.

Określenie 1.2

Wartość stanu $v^{\eta,l}$ jest oceną stanu $P^{\eta,l}$ z punktu widzenia przyjętego kryterium optymalizacji. A zatem wartość stanu koresponduje z przyjętym kryterium optymalizacji. Rozwiązanie optymalne daje stan końcowy o ekstremalnej wartości. W ogólnym przypadku wartość stanu jest wektorem, którego współrzędne wyznacza się na podstawie danego stanu, za pomocą określonych funkcji wartości stanu. Współrzędne wartości stanu można wyznaczyć również rekurencyjnie w trakcie generowania stanu $P^{\eta,l}$ z wybranego stanu $P^{\eta-1,\lambda}$. W problemach wielokryterialnych poszczególne współrzędne wektora wartości stanu korespondują z odpowiednimi kryteriami. W problemach jednokryterialnych wartość stanu jest (najczęściej) skalarem. Dla kryterium minimalizacji stan globalnie optymalny P^o spełnia warunek:

$$(\min_l v^{N,l} = v^{N,l^o}) \Longrightarrow (P^{N,l^o} = P^o) \qquad (1.10)$$

W obliczeniach stan P^o jest ostatnim z wyznaczonych stanów P^a.

Dopuszczalne (i optymalne) rozwiązania problemu reprezentują stany końcowe. Żeby zatem otrzymać rozwiązanie dopuszczalne, należy wygenerować

kompletną trajektorię. Każda trajektoria wychodzi ze stanu $P^{0,1}$. Generowanie stanów polega na stosowaniu reguł: wyboru, podziału oraz odpowiednich procedur. Do generowania stanów wykorzystywane są stany aktywne, które są elementami uporządkowanych zbiorów $\mathcal{E}_\eta$, $\eta = 0,\ldots,N-1$ (stany końcowe nie są aktywne). Liczba elementów zbioru $\mathcal{E}_\eta$ może być ograniczona do L_η, ($L_\eta \geq 1$).

Do wyboru stanu aktywnego $P^{\eta-1,\lambda}$ można stosować klasyczne reguły wyboru, np. FIFO, LIFO, LLB itp. oraz strategie czyste lub mieszane. Jeżeli z wybranego stanu $P^{\eta-1,\lambda}$ są generowane wszystkie jego bezpośrednie następniki $P^{\eta,l}$, to reguła podziału jest zupełna. W przypadku przeciwnym podział jest częściowy. Stosując podział zupełny, stan $P^{\eta-1,\lambda}$ eliminujemy z dalszych obliczeń, ponieważ przestaje być aktywny. Przy podziale częściowym stan $P^{\eta-1,\lambda}$ pozostaje aktywny. W tym przypadku z każdym stanem $P^{\eta-1,\lambda}$ jest związany indeks $I^{\eta-1,\lambda}$. Indeks stanu $I^{\eta-1,\lambda}$ określa, które bezpośrednie następniki $P^{\eta,l}$ zostały już wygenerowane ze stanu $P^{\eta-1,\lambda}$. W ten sposób każdy stan jest generowany jednokrotnie, a ponadto żaden stan nie zostaje pominięty.

Procedura generowania stanów jest zdaniem logicznym określającym warunki, jakie muszą być spełnione, by wygenerować stan. Procedura jednoetapowa pozwala wygenerować bezpośrednie następniki $P^{\eta,l}$ wybranego stanu aktywnego $P^{\eta-1,\lambda}$. Procedura wieloetapowa generuje następniki $P^{e,l}$ ($\eta \leq e \leq N$) danego stanu $P^{\eta-1,\lambda}$. Generowanie stanów rozpoczyna się od stanu $P^{0,1}$ i trwa dopóty, dopóki zbiory $\mathcal{E}_\eta$, $\eta = 0,\ldots,N-1$ nie są puste. Jeżeli zbiory $\mathcal{E}_\eta$ są puste, to obliczenia zostają zakończone.

W trakcie obliczeń eliminowane są stany nieperspektywiczne, które nie pozwalają uzyskać optymalnego rozwiązania problemu. Do eliminacji stanów nieperspektywicznych służą reguły: wyczerpywania, sondowania oraz dominacji. Reguła wyczerpywania eliminuje stan $P^{\eta,l}$, z którego nie można otrzymać dopuszczalnego stanu końcowego $P^{N,l}$ (wyczerpane zostały możliwości generowania rozwiązań dopuszczalnych). Reguła sondowania eliminuje stan $P^{\eta,l}$, którego stan lokalnie optymalny P^{N,l^o} jest gorszy od stanu aktualnie najlepszego P^a. W tym celu dla stanu $P^{\eta,l}$ wyznacza się dolne ograniczenie $b^{\eta,l}$ wartości stanu v^{N,l^o_η}, tzn.:

$$b^{\eta,l} \leq v^{N,l^o_\eta} \qquad (1.11)$$

Tak więc, jeżeli:

$$b^{\eta,l} \geq v^a \qquad (1.12)$$

to etan $P^{\eta,l}$ jest nieperspektywiczny.

Określenie 1.3

Stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}, jeżeli jest spełniona nierówność:

$$V^{N,l^{\circ}_{\eta_1}} < V^{N,l^{\circ}_{\eta_2}} \qquad (1.13)$$

A zatem stan lokalnie optymalny $P^{N,l^{\circ}_{\eta_1}}$ daje lepsze rozwiązanie niż stan lokalnie optymalny $P^{N,l^{\circ}_{\eta_2}}$.

Reguła dominacji stanów pozwala wyeliminować jeden z dwóch stanów aktywnych P^{η_1,l_1} lub P^{η_2,l_2}, którego odpowiedni stan lokalnie optymalny jest gorszy. Ponieważ w obliczeniach wartości odpowiednich stanów lokalnie optymalnych nie są znane, zatem trzeba formułować twierdzenia, które muszą spełniać stany P^{η_1,l_1} i P^{η_2,l_2}, by zachodziła nierówność (1.13). Na twierdzeniach takich oparte są reguły dominacji stanów.

Rozwiązywanie problemów: BLM, AO oraz AO i Z przy pomocy metody programowania wieloetapowego polega na generowaniu trajektorii - poprzez przydzielanie operacji montażowych do kolejnych stacji. Dopuszczalne rozwiązanie reprezentuje stan końcowy, który otrzymujemy po przydzieleniu wszystkich operacji. Zatem w metodzie programowania wieloetapowego warunki (1.5) i (1.6) są zawsze spełnione, dla linii szeregowych. Dlatego w przedstawionych dalej modelach matematycznych nie będę one wypisywane w sposób jawny.

2. BLM METODĄ PROGRAMOWANIA WIELOETAPOWEGO

2.1. WSTĘP

W niniejszym rozdziale rozważymy jednowersyjny proces montażu na linii o szeregowych stanowiskach pracy. Założymy zatem, że dla zadanego cyklu c liczba stanowisk pracy M nie przekracza liczby stacji montażowych K, tzn.:

$$M \leq K \qquad (2.1)$$

Ponadto zakładamy, że kolejne podzbiory operacji $\Omega_1,\ldots,\Omega_m,\ldots,\Omega_M$ są lokowane na kolejnych stacjach montażowych od pierwszej do k-tej, $(k \leq K)$. Tym samym lokalizacja stanowisk pracy na stacjach montażowych jest jednoznaczna.

Przy powyższych założeniach alokacja operacji na linii montażowej sprowadza się do klasycznego problemu BLM. W niniejszym rozdziale problem ten rozwiążemy za pomocą algorytmów opartych na metodzie programowania wieloetapowego. W tym celu zostaną określone: stan procesu decyzyjnego, wartość stanu, procedura generowania stanów oraz reguły eliminacji stanów nieperspektywicznych.

Wyróżnione zostaną algorytmy: binarny, szeregowania, przydziału oraz harmonogramowania. Nazwa algorytmu wynika z charakteru informacji zawartej w stanie. Algorytmy te przy jednakowej złożoności pozwalają rozwiązywać problemy o różnym stopniu ogólności.

Wyszczególnione wyżej algorytmy programowania wieloetapowego dla problemu BLM zostaną zilustrowane przykładami obliczeniowymi. Ponadto podane zostaną elementy mające wpływ na efektywność tych algorytmów. Do dalszych rozważań (uogólnionych problemów BLM, AO i AOiZ) uzasadnimy wybór algorytmu harmonogramowania.

2.2. PODSTAWOWE OKREŚLENIA

Rozważmy klasyczny problem BLM, sformułowany w rozdziale 1. Dla rozwiązania tego problemu metodą programowania wieloetapowego, stan procesu decyzyjnego zdefiniujemy następująco:

Definicja 2.1

Stan $P^{\eta,l}$ jest wektorem:

$$P^{\eta,l} = \left[p_n^{\eta,l}\right]_{\substack{n=1,\ldots,N \\ \eta=0,\ldots,N \\ l=1,\ldots,L_\eta}} \tag{2.2}$$

gdzie: L_η - liczba stanów η-tego etapu.

Współrzędne stanu $P^{\eta,l}$ zostaną określone różnie dla poszczególnych algorytmów (binarnego, szeregowania, przydziału oraz harmonogramowania). W ogólnym przypadku:

$$p_n^{\eta,l} \neq 0 \tag{2.3a}$$

jeżeli podjęta została decyzja o realizacji ω_n; w przypadku przeciwnym przyjmiemy:

$$p_n^{\eta,l} = 0 \tag{2.3b}$$

Tak więc stan początkowy $P^{0,1}$ jest wektorem zerowym, natomiast wszystkie współrzędne stanu końcowego $P^{N,l}$ są różne od zera. W zależności od określenia współrzędnych stanu, ze stanu końcowego $P^{N,l}$ w różny sposób jest wyznaczane dopuszczalne rozwiązanie problemu BLM.

Z każdym stanem $P^{\eta,l}$ jest związana jego wartość $V^{\eta,l}$, która koresponduje z przyjętym kryterium optymalizacji BLM (minimalizacja liczby stanowisk pracy). Przed podaniem definicji wartości stanu wprowadzimy pewne dodatkowe określenia.

Rozważmy montaż wybranego obiektu od chwili wejścia chassis na pierwszą stację (pierwsze stanowisko pracy). Oznaczmy przez t_n chwilę zakończenia realizacji operacji ω_n na linii. A zatem t_n oznacza równocześnie czas, który upływa od wejścia montowanego obiektu na pierwszą stację do zakończenia realizacji ω_n (włącznie z nie wykorzystanym czasem na niektórych stanowiskach pracy).

Oznaczmy przez $\Omega^{\eta,l}$ zbiór operacji wykonanych w stanie $P^{\eta,l}$, a zatem:

$$\forall_n \left(p_n^{\eta,l} \neq 0\right) \Rightarrow \left(\omega_n \in \Omega^{\eta,l}\right) \tag{2.4}$$

Oznaczmy przez $T^{\eta,l}$ chwilę zakończenia wszystkich operacji ze zbioru $\Omega^{\eta,l}$ (czyli w stanie $P^{\eta,l}$). Tak więc:

$$T^{\eta,l} = \max_{\omega_n \in \Omega^{\eta,l}} t_n \tag{2.5}$$

Dla stanu $P^{0,1}$ mamy:

$$T^{0,1} = 0 \tag{2.6}$$

Definicja 2.2

Wartość stanu $V^{\eta,l}$ jest skalarem $v^{\eta,l}$:

$$v^{\eta,l} = \left[\frac{1}{c} T^{\eta,l}\right]^+ \qquad (2.7 \tag{2.7}$$

gdzie: $[\cdot]^+$ - najmniejsza liczba całkowita nie mniejsza od wartości w nawiasie kwadratowym.

Wartość stanu jest zatem liczbą stanowisk pracy wykorzystanych w tym stanie. Optymalny stan końcowy P^o wyznaczamy z warunku

$$\left(\min_{1 \leq l \leq L_N} v^{N,l} = v^{N,l^o}\right) \Rightarrow \left(P^{N,l^o} = P^o\right) \tag{2.8}$$

Jeżeli zbiór stanów końcowych nie jest dany explicite, lecz stany końcowe są generowane sekwencyjnie, to P^o jest ostatnim spośród wyznaczonych stanów aktualnie najlepszych P^a.

W przypadku gdy ze stanu $P^{\eta,l}$ można określić $T^{\eta,l}$, wartość stanu $V^{\eta,l}$ wyznaczamy z tzw. funkcji wartości stanu:

$$V^{\eta,l} = V(P^{\eta,l}) \tag{2.9}$$

W przypadku przeciwnym wartość stanu wyznaczamy z formuły rekurencyjnej, w trakcie generowania stanu:

$$V^{\eta,l} = V^{\eta-1,\lambda} + \Delta V \tag{2.10}$$

gdzie: $V^{\eta-1,\lambda}$ - wartość bezpośredniego poprzednika $P^{\eta-1,\lambda}$ stanu $P^{\eta,l}$, ΔV - przyrost wartości stanu.

Generowanie stanów polega na uzupełnieniu wybranego stanu $P^{\eta-1,\lambda}$ o operację ω_n tak, by otrzymać kolejny stan $P^{\eta,l}$.

Dla rozważanego klasycznego problemu BLM procedura generowania stanów ma postać:

$$\forall_n \forall_\eta \left(p_n^{\eta-1,\lambda} = 0\right) \wedge \left[\left(\eta_{\lambda,n} = 1\right) \Rightarrow \left(p_\eta^{\eta-1,\lambda} \neq 0\right)\right] \Rightarrow \left(P^{\eta,l} = P^{\eta-1,\lambda} + \Delta P\right) \tag{2.11}$$

Wektor ΔP ma współrzędne:

$$\Delta p_i = 0, \quad \text{dla } i \neq n \tag{2.12a}$$

oraz:

$$\Delta p_n \neq 0 \tag{2.12b}$$

Wartość współrzędnej Δp_n będzie określona odmiennie dla poszczególnych algorytmów.

Generując stan $P^{\gamma,l}$ ze stanu $P^{\gamma-1,\lambda}$ przez przydzielenie do realizacji operacji ω_n, możemy napisać:

$$T^{\gamma,l} = T^{\gamma-1,\lambda} + \Delta T \tag{2.13}$$

przy tym:

$$\Delta T = \begin{cases} \vartheta_n, & \text{jeśli } \left[\frac{1}{c} T^{\gamma-1,\lambda}\right]^+ = \left[\frac{1}{c}\,(T^{\gamma-1,\lambda} + \vartheta_n)\right]^+ \\ c\left[\frac{1}{c} T^{\gamma-1,\lambda}\right]^+ - T^{\gamma-1,\lambda} + \vartheta_n, & \text{w przypadku przeciwnym} \end{cases} \tag{2.14}$$

ponadto

$$\Delta V = \begin{cases} 0, & \text{jeśli } \left[\frac{1}{c} T^{\gamma-1,\lambda}\right]^+ = \left[\frac{1}{c}\,(T^{\gamma-1,\lambda} + \vartheta_n)\right]^+ \\ 1, & \text{w przypadku przeciwnym} \end{cases} \tag{2.15}$$

Tak więc, jeśli w obliczeniach oprócz stanów $P^{\gamma,l}$ są zapamiętywane wartości $T^{\gamma,l}$, to za pomocą formuł rekurencyjnych (2.13), (2.14) i (2.7) lub (2.13), (2.14), (2.15) i (2.10) można łatwo wyznaczyć wartość generowanego stanu.

2.3. REGUŁY ELIMINACJI STANÓW NIEPERSPEKTYWICZNYCH

Do eliminacji stanów niperspektywicznych wykorzystujemy reguły: wyczerpywania, sondowania oraz dominacji. Reguły wyczerpywania oraz sondowania wykorzystują oszacowanie dolnego ograniczenia $b^{\gamma,l}$ wartości V^{N,l^o} stanu lokalnie optymalnego P^{N,l^o}.

TWIERDZENIE 2.1.

Dla stanu $P^{\gamma,l}$ oszacowanie dolnego ograniczenia $b^{\gamma,l}$ można wyznaczyć ze wzoru:

$$b^{\gamma,l} = \left[\frac{1}{c}\left(T^{\gamma,l} + \sum_{\omega_n \notin \Omega^{\gamma,l}} \vartheta_n\right)\right]^+ \tag{2.16}$$

D o w ó d. Dla każdej trajektorii $P^{\gamma,l},\ldots,P^{N,\lambda}$ możemy napisać

$$T^{N,\lambda} = T^{\gamma,l} + \Delta T' \tag{2.17}$$

Z (2.13) i (2.14) wynika, że:

$$\Delta T' \geq \sum_{\omega_n \notin \Omega^{\gamma,l}} \vartheta_n \tag{2.18}$$

Uwzględniając (2.18) w (2.17) otrzymamy:

$$T^{N,\lambda} \geq T^{\gamma,l} + \sum_{\omega_n \notin \Omega^{\gamma,l}} \vartheta_n \tag{2.19}$$

skąd:

$$\left[\frac{1}{c} T^{N,\lambda}\right]^+ \geq \left[\frac{1}{c}\left(T^{\gamma,l} + \sum_{\omega_n \notin \Omega^{\gamma,l}} \vartheta_n\right)\right]^+ \tag{2.20}$$

Uwzględniając (2.7) i (2.16) w (2.20) dochodzimy do nierówności:

$$V^{N,\lambda} \geq b^{\gamma,l} \tag{2.21}$$

Nierówność (2.21) jest spełniona dla każdego stanu końcowego $P^{N,\lambda}$ (otrzymanego ze stanu $P^{\gamma,l}$), a zatem również dla stanu lokalnie optymalnego P^{N,l^o}. □

Reguła wyczerpywania

Ponieważ rozważana linia montażowa składa się z K stacji, zatem stan $P^{\gamma,l}$ spełniający warunek:

$$K < b^{\gamma,l} \tag{2.22}$$

jest nieperspektywiczny.

Reguła sondowania

Załóżmy, że znany jest aktualnie najlepszy stan końcowy P^a o wartości V^a. Stan $P^{\gamma,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$b^{\gamma,l} \geq V^a \tag{2.23}$$

Reguły dominacji

Załóżmy, że dane są dwa stany P^{η,l_1} oraz P^{η,l_2}.

TWIERDZENIE 2.2

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeśli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leqslant T^{\eta,l_2}) \tag{2.24}$$

D o w ó d. Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\eta,l_2},\ldots,P^{N,l_2^o}$. Każda operacja $\omega_n \notin \Omega^{\eta,l_2}$ jest wykonana zgodnie z tą trajektorią w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Jeżeli warunek (2.24) jest spełniony, to dopuszczalna jest trajektoria $P^{\eta,l_1},\ldots,P^{N,\lambda}$, w której każda operacja $\omega_n \notin \Omega^{\eta,l_1}$ jest wykonana również w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Możemy zatem napisać:

$$T^{N,\lambda} = T^{N,l_2^o} \tag{2.25}$$

lub

$$V^{N,\lambda} \leqslant V^{N,l_2^o} \tag{2.26}$$

czyli ze stanu P^{η,l_2} nie można uzyskać lepszego rozwiązania niż ze stanu P^{η,l_1}. Gdy $T^{\eta,l_1} < T^{\eta,l_2}$, to może istnieć lokalnie optymalna trajektoria $P^{\eta,l_1},\ldots,P^{N,l_1^o}$, dla której $T^{N,l_1^o} < T^{N,l_2^o}$. □

TWIERDZENIE 2.3

Stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}, $\eta_1 \geqslant \eta_2$, jeżeli jest spełniony warunek:

$$(\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}) \wedge (T^{\eta_1,l_1} \leqslant T^{\eta_2,l_2}) \tag{2.27}$$

D o w ó d. Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\eta_2,l_2},\ldots,P^{N,l_2^o}$, dla operacji $\omega_n \notin \Omega^{\eta_2,l_2}$. Operacje te są wykonywane sekwencyjnie w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Jeżeli w sekwencji tej pominiemy operacje ω_i takie, że:

$$\omega_i \in (\Omega^{\eta_1,l_1} \setminus \Omega^{\eta_2,l_2}) \tag{2.28}$$

to przy spełnieniu (2.27) otrzymamy dopuszczalną trajektorię $P^{\eta_1,l_1},\ldots,P^{N,\lambda}$. Dla stanów P^{N,l_2^o} i $P^{N,\lambda}$ możemy napisać:

$$T^{N,\lambda} \leqslant T^{N,l_2^o} \tag{2.29}$$

A zatem:

$$V^{N,l_1^o} \leqslant V^{N,\lambda} \leqslant V^{N,l_2^o} \tag{2.30}$$

czyli stan P^{η_2,l_2} jest nieperspektywiczny. □

TWIERDZENIE 2.4

Jeżeli stany P^{η_1,l_1} i P^{η_2,l_2} spełniają warunki:

$$\bigvee_n (\omega_n \notin \Omega^{\eta_1,l_1}) \Rightarrow \left\{ \left[\tfrac{1}{c}(T^{\eta_1,l_1} + \vartheta_n)\right]^+ > V^{\eta_1,l_1} \right\} \tag{2.31a}$$

oraz:

$$\bigvee_i (\omega_i \notin \Omega^{\eta_2,l_2}) \Rightarrow \left\{ \left[\tfrac{1}{c}(T^{\eta_2,l_2} + \vartheta_i)\right]^+ > V^{\eta_2,l_2} \right\} \tag{2.31b}$$

a ponadto

$$(\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}) \wedge (V^{\eta_1,l_1} = V^{\eta_2,l_2}) \tag{2.32}$$

to stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}.

D o w ó d. Jeżeli spełnione są warunki (2.31) i (2.32), to operacje $\omega_n \notin \Omega^{\eta_1,l_1}$ oraz $\omega_i \notin \Omega^{\eta_2,l_2}$ są realizowane od chwili $V^{\eta_1,l_1} \cdot c = V^{\eta_2,l_2} \cdot c$, nawet w przypadku gdy $T^{\eta_1,l_1} > T^{\eta_2,l_2}$. Zatem analogicznie jak w twierdzeniu 2.3 można wykazać, że stan P^{η_1,i_1} dominuje nad stanem P^{η_2,l_2}. □

Eliminacja stanów nieperspektywicznych polega na testowaniu wygenerowanego stanu $P^{\eta,l}$, $\eta < N$ za pomocą reguł - kolejno: wyczerpywania, sondowania i dominacji. Reguły dominacji stosujemy dla $1 < \eta < N$.

Reguły dominacji

Załóżmy, że dane są dwa stany P^{η,l_1} oraz P^{η,l_2}.

TWIERDZENIE 2.2

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeśli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \tag{2.24}$$

D o w ó d. Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\eta,l_2},\ldots,P^{N,l_2^o}$. Każda operacja $\omega_n \notin \Omega^{\eta,l_2}$ jest wykonana zgodnie z tą trajektorią w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Jeżeli warunek (2.24) jest spełniony, to dopuszczalna jest trajektoria $P^{\eta,l_1},\ldots,P^{N,\lambda}$, w której każda operacja $\omega_n \notin \Omega^{\eta,l_1}$ jest wykonana również w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Możemy zatem napisać:

$$T^{N,\lambda} = T^{N,l_2^o} \tag{2.25}$$

lub

$$V^{N,\lambda} \leq V^{N,l_2^o} \tag{2.26}$$

czyli ze stanu P^{η,l_2} nie można uzyskać lepszego rozwiązania niż ze stanu P^{η,l_1}. Gdy $T^{\eta,l_1} < T^{\eta,l_2}$, to może istnieć lokalnie optymalna trajektoria $P^{\eta,l_1},\ldots,P^{N,l_1^o}$, dla której $T^{N,l_1^o} < T^{N,l_2^o}$. □

TWIERDZENIE 2.3

Stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}, $\eta_1 \geq \eta_2$, jeżeli jest spełniony warunek:

$$(\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}) \wedge (T^{\eta_1,l_1} \leq T^{\eta_2,l_2}) \tag{2.27}$$

D o w ó d. Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\eta_2,l_2},\ldots,P^{N,l_2^o}$, dla operacji $\omega_n \notin \Omega^{\eta_2,l_2}$. Operacje te są wykonywane sekwencyjnie w przedziale $\left[t_n - \vartheta_n, t_n\right]$. Jeżeli w sekwencji tej pominiemy operacje ω_i takie, że:

$$\omega_i \in (\Omega^{\eta_1,l_1} \setminus \Omega^{\eta_2,l_2}) \tag{2.28}$$

to przy spełnieniu (2.27) otrzymamy dopuszczalną trajektorię $P^{\eta_1,l_1},\ldots,P^{N,\lambda}$. Dla stanów P^{N,l_2^o} i $P^{N,\lambda}$ możemy napisać:

$$T^{N,\lambda} \leq T^{N,l_2^o} \tag{2.29}$$

A zatem:

$$V^{N,l_1^o} \leq V^{N,\lambda} \leq V^{N,l_2^o} \tag{2.30}$$

czyli stan P^{η_2,l_2} jest nieperspektywiczny. □

TWIERDZENIE 2.4

Jeżeli stany P^{η_1,l_1} i P^{η_2,l_2} spełniają warunki:

$$\bigvee_n (\omega_n \notin \Omega^{\eta_1,l_1}) \Rightarrow \left(\left[\tfrac{1}{c}(T^{\eta_1,l_1} + \vartheta_n) \right]^+ > V^{\eta_2,l_1} \right) \tag{2.31a}$$

oraz:

$$\bigvee_i (\omega_i \notin \Omega^{\eta_2,l_2}) \Rightarrow \left(\left[\tfrac{1}{c}(T^{\eta_2,l_2} + \vartheta_i) \right]^+ > V^{\eta_2,l_2} \right) \tag{2.31b}$$

a ponadto

$$(\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}) \wedge (V^{\eta_1,l_1} = V^{\eta_2,l_2}) \tag{2.32}$$

to stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}.

D o w ó d. Jeżeli spełnione są warunki (2.31) i (2.32), to operacje $\omega_n \notin \Omega^{\eta_1,l_1}$ oraz $\omega_i \notin \Omega^{\eta_2,l_2}$ są realizowane od chwili $V^{\eta_1,l_1} \cdot c = V^{\eta_2,l_2} \cdot c$, nawet w przypadku gdy $T^{\eta_1,l_1} > T^{\eta_2,l_2}$. Zatem analogicznie jak w twierdzeniu 2.3 można wykazać, że stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}. □

Eliminacja stanów nieperspektywicznych polega na testowaniu wygenerowanego stanu $P^{\eta,l}$, $\eta < N$ za pomocą reguł - kolejno: wyczerpywania, sondowania i dominacji. Reguły dominacji stosujemy dla $1 < \eta < N$.

2.4. ALGORYTMY PROGRAMOWANIA WIELOETAPOWEGO

W oparciu o ideę metody programowania wieloetapowego rozważymy cztery algorytmy, które różnią się przede wszystkim definicją współrzędnych stanu.

2.4.1. Algorytm binarny

W algorytmie binarnym współrzędne wektora stanu $P^{\eta,l}$ przyjmują wartości 0 lub 1, czyli:

$$\Delta p_n^{\eta,l} = \begin{cases} 1, \text{ jeśli została podjęta decyzja o wykonaniu operacji } \omega_n; \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{2.33}$$

Odpowiednio dla wektora ΔP otrzymamy:

$$\Delta p_i = \begin{cases} 1, & \text{dla } i = n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.34}$$

Każdy stan końcowy $P^{N,l}$ ma wszystkie współrzędne równe jedności. A zatem ze stanu końcowego nie można określić wprost podzbiorów Ω_k, $k=1,\ldots,K$. Podzbiory te wyznaczamy następująco.

Dla każdego stanu $P^{\eta,l}$ dany jest czas $T^{\eta,l}$ (wielkości te są zapamiętywane w obliczeniach). Zgodnie z warunkiem (2.8) otrzymamy optymalny stan końcowy P^o, z którego:

$$M = v^o \tag{2.35}$$

Aby określić optymalne podzbiory $\Omega_1^o,\ldots,\Omega_m^o,\ldots,\Omega_M^o$, trzeba wyznaczyć optymalną trajektorię $P^{0,1},\ldots,P^{\eta-1,\lambda^o}$, $P^{\eta,l^o},\ldots,P^o$. Obliczenia prowadzimy od stanu P^o do stanu $P^{0,1}$. Jeżeli znany jest optymalny stan η-tego etapu P^{η,l^o}, to optymalny stan $\eta-1$-go etapu $P^{\eta-1,\lambda^o}$ wyznaczamy ze wzoru:

$$\min_{\lambda \in \alpha^{\eta,l^o}} (T^{\eta-1,\lambda} + \Delta T^{\eta-1,\lambda}) = T^{\eta-1,\lambda^o} + \Delta T^{\eta-1,\lambda^o} = T^{\eta,l^o} \tag{2.36}$$

przy tym zbiór α^{η,l^o} numerów λ wyznaczamy z warunku:

$$\exists_n \bigwedge_{\substack{1 \le i \le N \\ i \ne n}} (p_i^{\eta,l^o} = p_i^{\eta-1,\lambda}) \wedge (p_n^{\eta,l^o} > p_n^{\eta-1,\lambda}) \Rightarrow (\lambda \in \alpha^{\eta,l^o}) \tag{2.37}$$

Z optymalnej trajektorii otrzymujemy optymalną sekwencję operacji, z której wyznaczamy kolejno $\Omega_M^o,\ldots,\Omega_m^o,\ldots,\Omega_1^o$.

2.4.2. Algorytm szeregowania

W algorytmie szeregowania współrzędne stanu określają sekwencję operacji, bowiem przyjmujemy:

$$p_n^{\eta,l} = \begin{cases} s, \text{ jeśli } \omega_n \text{ ma być wykonana jako } s\text{-ta z kolei, } 1 \le s \le \eta; \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{2.38}$$

Odpowiednio dla wektora ΔP otrzymamy:

$$\Delta p_i = \begin{cases} \eta, & \text{dla } i = n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.39}$$

Każdy stan końcowy jest wektorem o dodatnich współrzędnych. Z optymalnego stanu końcowego P^o wyznaczana jest optymalna sekwencja operacji $\omega_{n_1},\ldots,\omega_{n_s},\ldots,\omega_{n_N}$, z której wyznaczamy wprost optymalne podzbiory operacji $\Omega_1^o,\ldots,\Omega_m^o,\ldots,\Omega_M^o$.

2.4.3. Algorytm przydziału

W algorytmie przydziału współrzędne stanu określają stanowiska pracy (stacje montażowe), do których zostały przydzielone operacje. Współrzędne te definiujemy następująco:

$$p_n^{\eta,l} = \begin{cases} m, \text{ jeśli operacja } \omega_n \text{ została przydzielona do podzbioru } \Omega_m; \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{2.40}$$

Odpowiednio dla wektora ΔP otrzymamy:

$$\Delta p_i = \begin{cases} \left[\frac{1}{c}(T^{\eta-1,\lambda} + t_n)\right]^*, & \text{dla } i = n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.41}$$

Z optymalnego stanu końcowego otrzymujemy wprost optymalne podzbiory operacji:

$$\bigvee_n (p_n^o = m) \Rightarrow (\omega_n \in \Omega_m^o) \tag{2.42}$$

oraz minimalną liczbę stanowisk pracy:

$$M = \max_{1 \leq n \leq N} p_n^o \tag{2.43}$$

2.4.4. Algorytm harmonogramowania

W algorytmie harmonogramowania współrzędne stanu określają chwilę zakończenia wykonywania operacji. Przyjmujemy, że:

$$p_n^{\eta,1} = \begin{cases} t_n, & \text{jeśli podjęto decyzję o wykonaniu operacji } \omega_n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.44}$$

Odpowiednio dla wektora ΔP otrzymamy:

$$\Delta p_i = \begin{cases} t_n, & \text{dla } i = n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.45}$$

Z optymalnego stanu końcowego P^o otrzymujemy wprost:

$$t_n = p_n^o \tag{2.46}$$

czyli optymalny harmonogram wykonania operacji.

Ponadto numer stanowiska pracy m_n, na którym ma być wykonana operacja ω_n, wynika z:

$$m_n = \left[\frac{1}{c} p_n^o\right]^+ \tag{2.47}$$

A zatem minimalną liczbę stanowisk pracy wyznaczamy jako:

$$M = \left[\frac{1}{c} \max_{1 \leq n \leq N} p_n^o\right]^+ \tag{2.48}$$

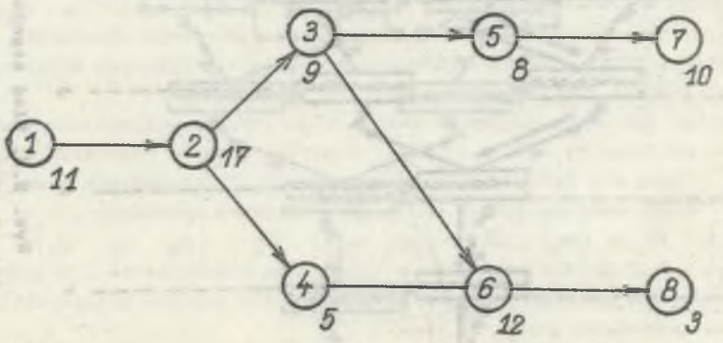
W algorytmie harmonogramowania chwilę t_n wyznaczamy ze wzoru:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \vartheta_n, & \text{gdy } \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c} (T^{\eta-1,\lambda} + \vartheta_n)\right]^+ \\ c \cdot \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ + \vartheta_n, & \text{w przypadku przeciwnym.} \end{cases} \tag{2.49}$$

Algorytm binarny wymaga więc zapamiętywania wszystkich stanów i dwukrotnego przejścia przez sieć. W algorytmach szeregowania, przydziału oraz harmonogramowania zapamiętywane są tylko stany aktywne, a rozwiązanie problemu otrzymujemy wprost ze stanów końcowych.

2.5. PRZYKŁADY

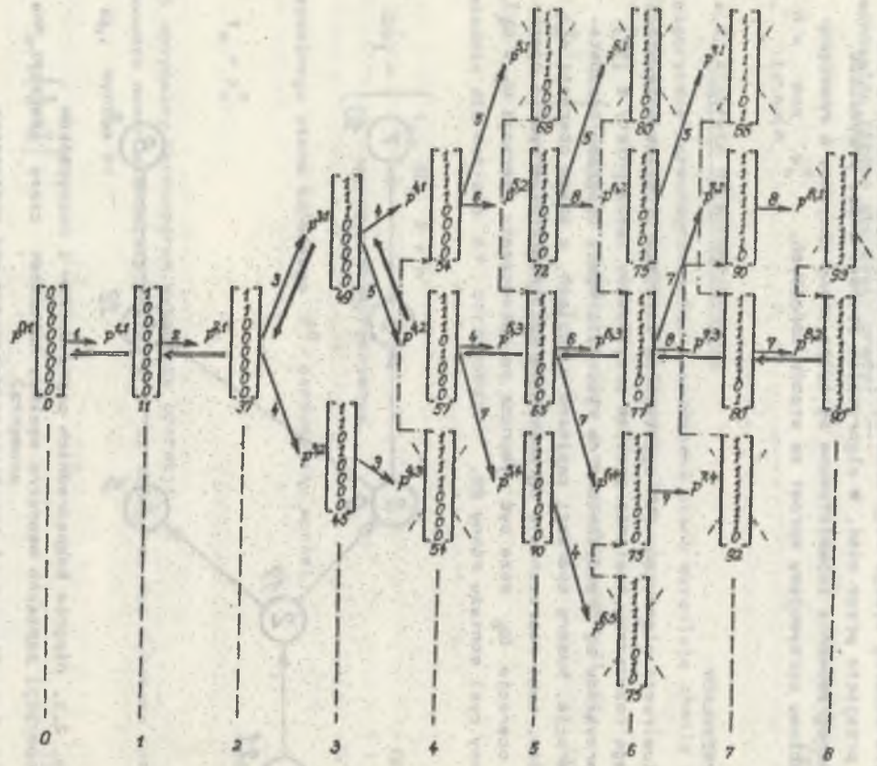
Dla ilustracji algorytmów BLM opartych na metodzie programowania wieloetapowego rozwiązany zostanie problem przykładowy zaczerpnięty z [30]. Dane do przykładu są przedstawione na rys. 2.1. Węzły diagramu przedstawiają operacje. Numery operacji zapisano w węzłach, a czasy operacji - obok węzłów. Łuki reprezentują ograniczenia kolejności wykonywania operacji (np. operacja ω_6 może być wykonana po zakończeniu operacji ω_3 i ω_4). Przyjmujemy cykl montażu równy 20.



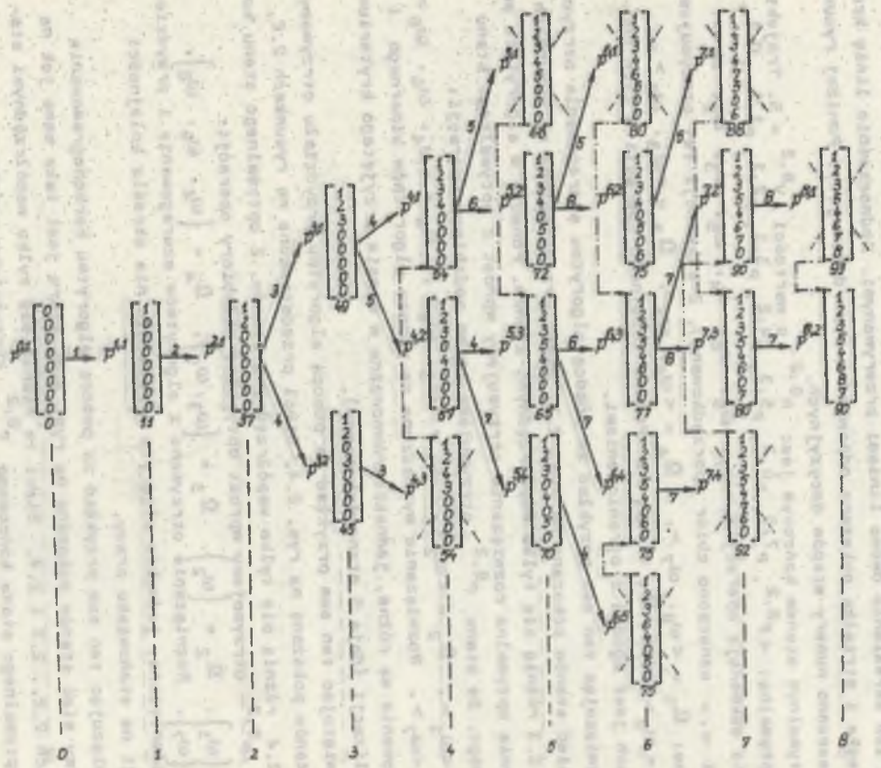
Rys. 2.1. Diagram bezpośrednich poprzedników i następników (numery operacji zapisano wewnątrz węzłów, natomiast czasy operacji na zewnątrz)

Fig. 2.1. Diagram of immediate predecessors and successors

Celem porównania kompletnej wiązki trajektorii dla poszczególnych algorytmów, w przykładach będzie stosowana reguła FIFO wyboru stanu aktywnego oraz reguła podziału zupełnego. Ponadto dla ilustracji eliminacji stanów będzie wykorzystana tylko reguła dominacji, oparta na twierdzeniu 2.2. Ponieważ w rozwiązywanym problemie występuje osiem operacji, proces decyzyjny składa się z ośmiu etapów. Rozwiązanie optymalne otrzymujemy z jednego ze stanów ostatniego etapu.



Rys. 2.2. Sieć stanów algorytmu binarnego
Fig. 2. Network of states for binary algorithm



Rys. 2.3. Sieć stanów algorytmu szeregowania
Fig. 2.3. Network of states for sequencing algorithm

Sieć stanów (wiązkę trajektorii) algorytmu binarnego pokazano na rys. 2.2. Stany są wektorami o współrzędnych 0 lub 1. Poniżej każdego wektora stanu zapisano czas montażu operacji wykonanych w danym stanie. Pojedynczymi strzałkami (z lewej na prawą stronę) zaznaczono bezpośrednie przejścia pomiędzy stanami (w trakcie generowania). Ponad tymi strzałkami podano numery przydzielonych operacji. Strzałki podwójne (z prawej na lewą stronę) wyznaczają trajektorię optymalną. Stany zdominowane oznaczono przez ich skreślenie dwoma liniami przerywanymi. Jednocześnie linią kropka-kreska i strzałką pokazano odpowiedni stan dominujący. Poniżej rysunku zamieszczono numery etapów decyzyjnych.

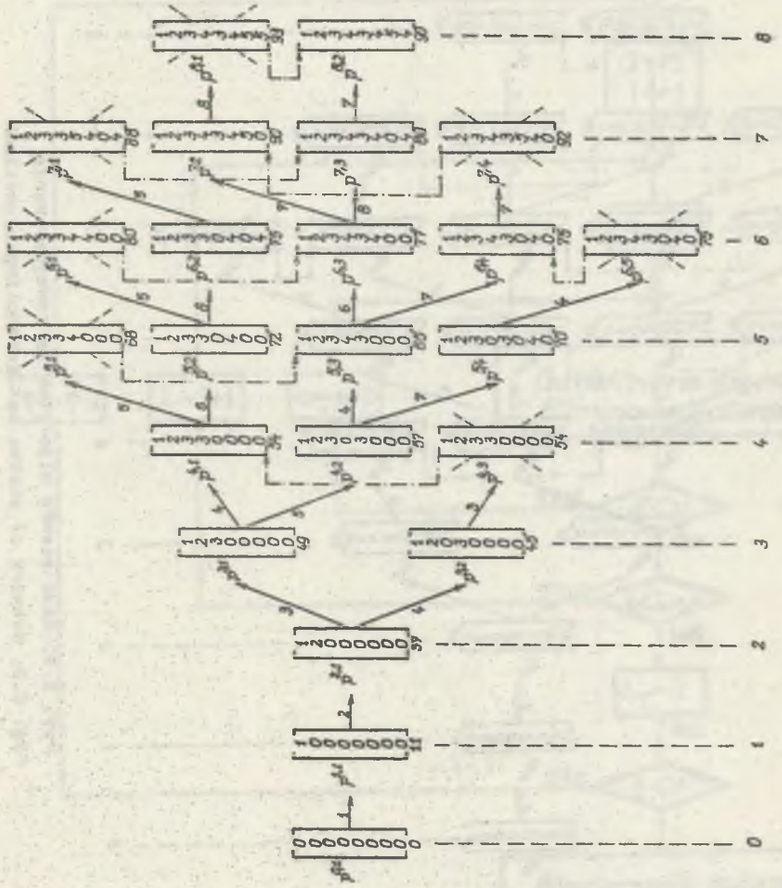
Optymalnym stanem końcowym jest $P^{8,2}$ o wartości $V^{8,2} = 5$. Trajektoria optymalna: $<P^{8,2}, P^{7,3}, P^{6,3}, P^{5,3}, P^{4,2}, P^{3,1}, P^{2,1}, P^{1,1}, P^{0,1}>$ określa sekwencję operacji: $<\omega_7, \omega_8, \omega_6, \omega_4, \omega_5, \omega_3, \omega_2, \omega_1>$ (przez $<.>$ oznaczono zbiór uporządkowany). Z sekwencji tej otrzymujemy kolejno: $\Omega_5 = <\omega_8, \omega_7>$, $\Omega_4 = <\omega_4, \omega_6>$, $\Omega_3 = <\omega_3, \omega_5>$, $\Omega_2 = <\omega_2>$ oraz $\Omega_1 = <\omega_1>$. Podana kolejność operacji w tych podzbiorach jest zgodna z ograniczeniami.

Rozwiązując ten sam przykład za pomocą algorytmu szeregowania otrzymujemy sieć stanów pokazaną na rys. 2.3. Sieci przedstawione na rysunkach 2.2 i 2.3 różnią się tylko współrzędnymi stanów. Ponadto w algorytmie szeregowania optymalne rozwiązanie otrzymujemy wprost z optymalnego stanu końcowego. Ze stanu $P^{8,2}$ otrzymujemy zatem podzbiory operacji: $\Omega_1 = <\omega_1>$, $\Omega_2 = <\omega_2>$, $\Omega_3 = <\omega_3, \omega_5>$, $\Omega_4 = <\omega_4, \omega_6, \omega_8>$, $\Omega_5 = <\omega_7>$. Rozwiązanie wyznaczone za pomocą algorytmów binarnego i szeregowania są różne, jednakże równoważne w sensie przyjętego kryterium optymalizacji (dają 5 stanowisk pracy).

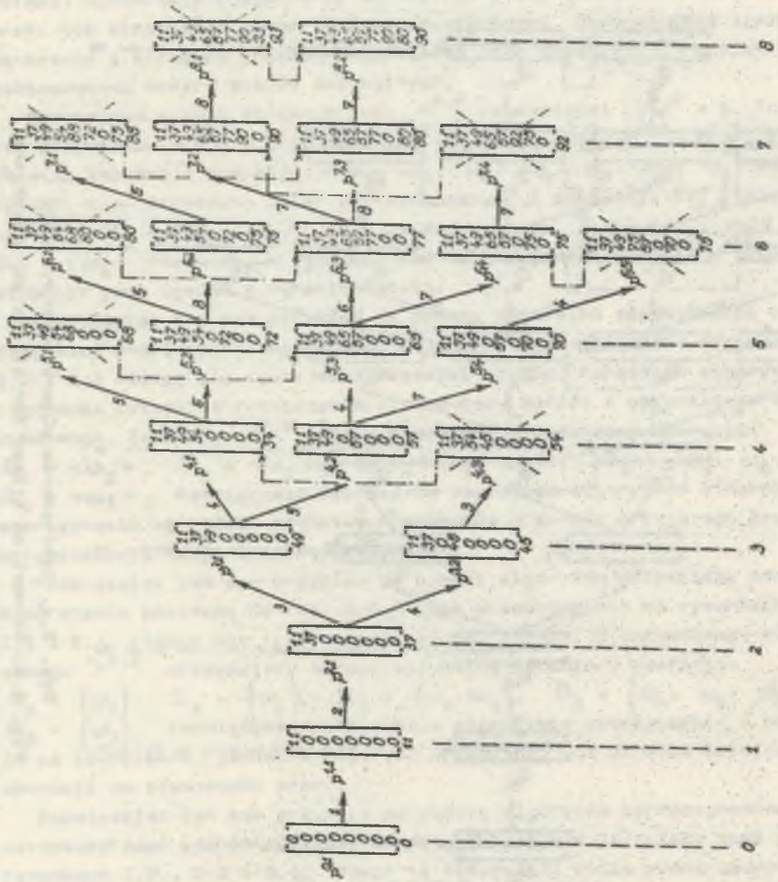
Rozwiązując ten sam przykład za pomocą algorytmu przydziału otrzymamy sieć stanów pokazaną na rys. 2.4. Sieci przedstawione na rysunkach 2.2, 2.3 i 2.4 różnią się tylko współrzędnymi stanów. Z optymalnego stanu końcowego $P^{8,2}$ otrzymujemy wprost optymalne podzbiory operacji: $\Omega_1 = \{\omega_1\}$, $\Omega_2 = \{\omega_2\}$, $\Omega_3 = \{\omega_3, \omega_5\}$, $\Omega_4 = \{\omega_4, \omega_6, \omega_8\}$, $\Omega_5 = \{\omega_7\}$. Rozwiązania otrzymane z algorytmów szeregowania i przydziału są identyczne, jednakże algorytm przydziału nie określa kolejności operacji na stanowisku pracy.

Rozwiązując ten sam przykład za pomocą algorytmu harmonogramowania otrzymamy sieć stanów pokazaną na rys. 2.5, która jest taka sama jak na rysunkach 2.2., 2.3 i 2.4. Sieci te różnią się tylko współrzędnymi stanu. Z optymalnego stanu końcowego $P^{8,2}$ otrzymujemy podzbiory operacji: $\Omega_1 = <\omega_1>$, $\Omega_2 < \omega_2>$, $\Omega_3 = <\omega_3, \omega_5>$, $\Omega_4 <\omega_4, \omega_6, \omega_8>$, $\Omega_5 = <\omega_7>$. Otrzymane rozwiązanie jest identyczne z tym, jakie uzyskano za pomocą algorytmów szeregowania i przydziału.

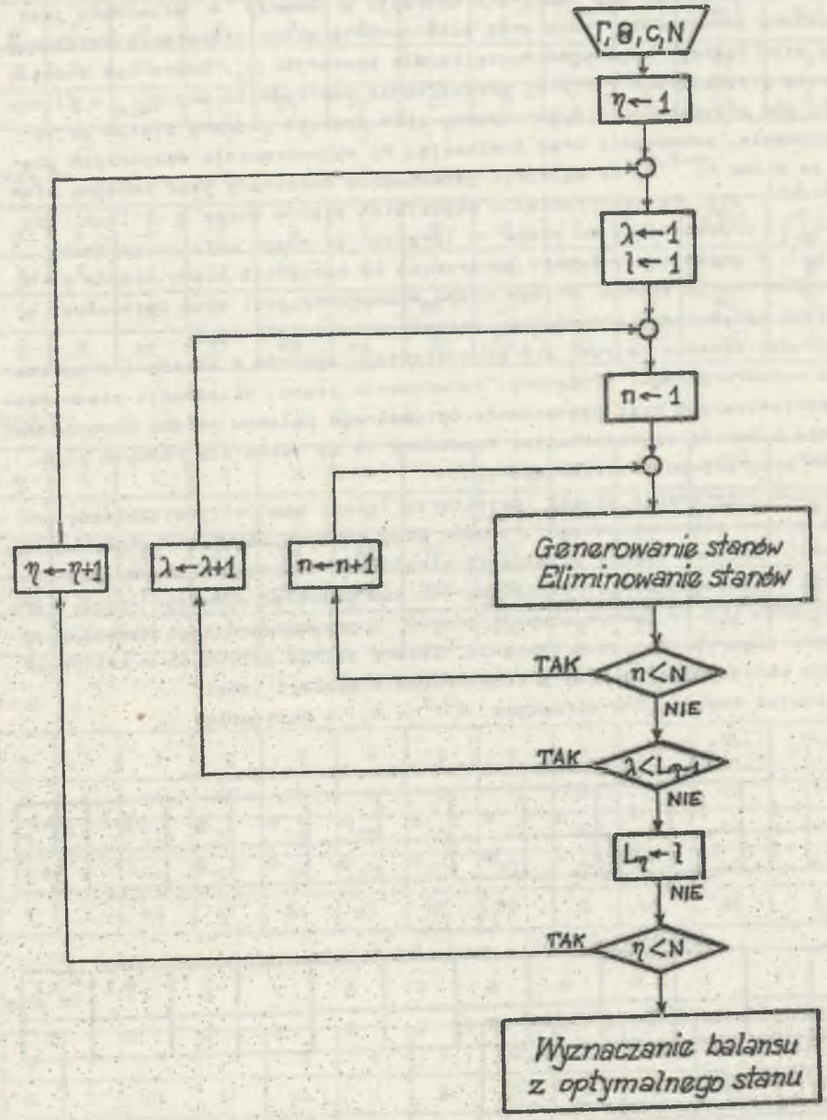
W przykładach ilustrujących algorytmy: binarny, szeregowania, przydziału oraz harmonogramowania, pokazano kompletną sieć stanów (wiązkę trajektorii). W algorytmach tych uwzględniono tylko regułę dominacji, stosując



Rys. 2.4. Sieć stanów algorytmu przydziału
Fig. 2.4. Network of states for allocation algorithm



Rys. 2.5. Sieć stanów algorytmu harmonogramowania
Fig. 2.5. Network of states for scheduling algorithm



Rys. 2.6. Schemat blokowy algorytmu BLM z regułę wyboru FIFO
Fig. 2.6. Block diagram of algorithm for ALB with selection rule FIFO

regułę FIFO wyboru stanu aktywnego. Schemat blokowy tych algorytmów pokazano na rys. 2.6. Ze schematu tego widzimy, że po wyborze na etapie $\eta-1$ stanu o numerze λ, dla kolejnych operacji o numerze n stosowana jest procedura generowania stanu oraz eliminowania stanu nieperspektywicznego. Nowy stan zostaje wygenerowany tylko dla operacji ω_n, która nie została jeszcze przydzielona i której bezpośrednie poprzedniki $\omega_s (\tau_{s,n} = 1)$ zostały już przydzielone. Wygenerowany stan zostaje poddany testom na wyczerpywanie, sondowanie oraz dominację. Po wygenerowaniu wszystkich stanów ze stanu $p^{\eta-1,\lambda}$, do dalszego generowania wybierany jest kolejny stan $p^{\eta-1,\lambda+1}$ itd. Po wykorzystaniu wszystkich stanów etapu $\eta-1$ (tzn. gdy $\lambda = L_{\eta-1}$) przechodzimy do etapu η (przyjmując numer ostatniego stanu $l = L_\eta$). W analogiczny sposób generowane są wszystkie stany etapów $\eta < N$. Po wygenerowaniu stanów N-tego etapu wyznaczany jest stan optymalny, a ze stanu optymalnego otrzymujemy optymalny balans linii.

Schemat blokowy na rys. 2.6 przedstawiono zgodnie z zasadami programowania strukturalnego. Procedury: generowania stanu, eliminacji stanu nieperspektywicznego oraz wyznaczania optymalnego balansu należy wkomponować zgodnie z zasadą zagnieżdżenia. Procedury te są różne dla różnych algorytmów programowania wieloetapowego.

Sposób generowania wiązki trajektorii (sieci stanów) jest zależny od reguł wyboru stanu aktywnego. Ponadto przy różnych regułach wyboru można stosować odmienne reguły eliminacji stanów nieperspektywicznych. Dla ilustracji tego zagadnienia rozwiążemy ten sam przykład stosując regułę LIFO, a następnie LLB (z pewnym uzupełnieniem). W przykładach tych zastosowany zostanie algorytm harmonogramowania. Zbiory stanów aktywnych w kolejnych krokach obliczeniowych będą przedstawione w postaci tabel.

Stosując regułę LIFO otrzymamy $b^{0,1} = 4$, a następnie:

Krok 1:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
1	1	11	0	0	0	0	0	0	0	11	4

Krok 2:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
2	1	11	37	0	0	0	0	0	0	37	5

Krok 3:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
3	2	11	37	0	45	0	0	0	0	45	5

Krok 4:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
4	1	11	37	54	45	0	0	0	0	54	5

Krok 5:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
5	1	11	37	54	45	68	0	0	0	68	5
5	2	11	37	54	45	0	72	0	0	72	5

Krok 6:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
5	1	11	37	54	45	68	0	0	0	68	5
6	1	11	37	54	45	80	72	0	0	80	5
6	2	11	37	54	45	0	72	0	75	75	5

Krok 7:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
5	1	11	37	54	45	68	0	0	0	68	5
6	1	11	37	54	45	80	72	0	0	80	5
7	1	11	37	54	45	88	72	0	75	88	5

Krok 8:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	$b^{\eta,1}$
3	1	11	37	49	0	0	0	0	0	49	5
5	1	11	37	54	45	68	0	0	0	68	5
6	1	11	37	54	45	80	72	0	0	80	5
8	1	11	37	54	45	88	72	90	75	90	5

W tabelach tych η - oznacza numer etapu, 1 - jest numerem stanu aktywnego w zbiorze $\mathcal{E}_\eta$. W kolumnach od 1 do 8 zapisane są kolejne współrzędne stanu $p^{\eta,1}$. Ponadto $T^{\eta,1}$ i $b^{\eta,1}$ są odpowiednimi wielkościami stanu $p^{\eta,1}$.

W przykładzie tym nie można stosować reguły dominacji. Nie stosowano również reguły sondowania - przed wyznaczeniem pierwszego ze stanów końcowych.

Jak wynika z obliczeń, rozwiązanie optymalne uzyskano w ósmym kroku, bowiem wszystkie stany aktywne okazały się nieperspektywiczne. Wygenerowano tylko 11 stanów, a w zbiorze stanów aktywnych były maksymalnie 4 stany. Dla porównania, stosując regułę FIFO wygenerowano 22 stany.

Rozważmy z kolei ten sam przykład z regułę wyboru LLB. Z rozwiązanego przykładu wynika, że dla $n > 1$ dolne ograniczenie $b^{\eta,1}$ wszystkich wygenerowanych stanów daje 5 stanowisk pracy. Dlatego stosując regułę wyboru LLB trzeba dodatkowo określić, który stan aktywny ma być wybrany. W przypadku reguły LLB_{FIFO} uzyskuje się taką samą kolejność generowania stanów jak dla reguły FIFO. Stosując regułę LLB_{LIFO} uzyskuje się przebieg obliczeń taki sam jak dla reguły LIFO. Aby zilustrować działanie reguły LLB, przyjmiemy jej modyfikację do postaci LLB/T_{LIFO}, gdzie

$$T = T^{\eta,1} + \sum_{\omega_n \notin \Omega^{\eta,1}} v_n \qquad (2.50)$$

Wybierany jest stan o najmniejszej wartości T.

Stosując tę regułę otrzymujemy przebieg obliczeń w pierwszych czterech krokach taki sam jak dla reguły LIFO. Pozostałe kroki obliczeniowe (od 5 do 11) pokazano niżej. Rozwiązanie uzyskano po 11 krokach. Oznaczenia elementów tabel są takie same jak w poprzednim przykładzie. Liniami przerywanymi ze strzałkami pokazano dominację stanów.

Krok 5:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
3	1	11	37	49	0	0	0	0	0	49	87
5	1	11	37	54	45	68	0	0	0	68	93
5	2	11	37	54	45	0	72	0	0	72	93

Krok 6:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	1	11	37	54	45	68	0	0	0	68	93
5	2	11	37	54	45	0	72	0	0	72	93
4	1	11	37	49	54	0	0	0	0	54	87

Krok 7:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	3	11	37	49	54	68	0	0	0	68	93
5	3	11	37	49	54	0	72	0	0	72	93
5	1	11	37	54	45	68	0	0	0	68	93
5	2	11	37	54	45	0	72	0	0	72	93

Krok 8:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	1	11	37	54	45	68	0	0	0	68	93
6	1	11	37	54	45	80	72	0	0	80	93
6	2	11	37	54	45	0	72	0	75	75	93

Krok 9:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	1	11	37	54	45	68	0	0	0	68	93
6	1	11	37	54	45	80	72	0	0	80	93
7	1	11	37	54	45	88	72	0	75	88	98

Krok 10:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	1	11	37	54	45	68	0	0	0	68	93
7	2	11	37	54	45	80	72	90	0	90	93
7	3	11	37	54	45	80	72	0	83	83	93
7	1	11	37	54	45	88	72	0	75	88	98

Krok 11:

η	1	1	2	3	4	5	6	7	8	$T^{\eta,1}$	T
5	1	11	37	54	45	68	0	0	0	68	93
7	1	11	37	54	45	80	72	90	0	90	93
8	1	11	37	54	45	80	72	93	93	93	93

Rozwiązane przykłady ilustrują sposób generowania trajektorii stanów dla różnych reguł wyboru stanu aktywnego. Eliminacja stanów niperspektywicznych za pomocą reguły sondowania była bardziej efektywna niż za pomocą reguły dominacji. Efektywność reguły dominacji ilustrują rezultaty testów komputerowych, przedstawione niżej.

Sformułowany problem przykładowy można rozwiązywać bez stosowania reguł eliminacji stanów niperspektywicznych. Wówczas generowane są wszystkie dopuszczalne stany, a więc wszystkie dopuszczalne rozwiązania (przegląd zupełny rozwiązań dopuszczalnych). Reguła dominacji eliminuje niektóre dopuszczalne stany niperspektywiczne. Efekt, jaki daje reguła dominacji, w porównaniu z przeglądem zupełnym rozwiązań dopuszczalnych (dla rozwiązywanego problemu przykładowego), ilustruje tabela 2.1.

Tabela 2.1

Porównanie sieci stanów w algorytmach z regułą dominacji stanów oraz bez dominacji stanów

Wariant	Liczba stanów na kolejnych etapach							
	1	2	3	4	5	6	7	8
Bez dominacji	1	1	2	3	6	11	16	16
Z dominacją	1	1	2	3	4	5	4	2

Tak więc z dominacją wygenerowano 22 stany, natomiast bez dominacji 56 stanów. Czas obliczeń dla wariantu z dominacją wynosił 0,814 [s] (Fortran, ODRA 1305).

2.5.1. Testy komputerowe

Dla ilustracji efektów, jakie daje reguła dominacji, przedstawione zostaną rezultaty 10 testów komputerowych. Obliczenia przeprowadzono w języku Fortran na komputerze ODRA 1305. Dla każdego testu zostaną podane:

- cykl c,
- liczba operacji N,
- czasy operacji (w kolejności od pierwszej do N-tej),
- relacja kolejności (w postaci uporządkowanych dwójek, "bezpośredni poprzednik - bezpośredni następnik").

Przebieg obliczeń zostanie pokazany poprzez podanie liczby stanów dla poszczególnych etapów decyzyjnych (dla programów bez dominacji oraz z dominacją stanów).

W każdym teście zostaną podane czasy obliczeń dla algorytmów bez dominacji oraz z dominacją i pewną modyfikacją dominacji. Modyfikacja ta polega na pominięciu sprawdzenia dominacji stanów, które zostały wygenerowane z tego samego bezpośredniego poprzednika (dla takich stanów dominacja nie zachodzi). Rezultatem obliczeń jest liczba stanowisk pracy na linii montażowej.

Rozwiązania suboptymalne zostaną oznaczone przez §. Rozwiązania takie otrzymuje się, gdy na pewnym etapie zostaną pominięte niektóre stany. Wynika to z ograniczonej wielkości tablicy stanów aktywnych. W tablicy tej stany etapu η-1 są zapisywane od góry, natomiast generowane steny etapu η są zapisywane od dołu. Jeżeli w tablicy takiej brak jest wolnych miejsc, to z danego stanu można wygenerować tylko jeden stan (na jego miejsce), natomiast pozostałe stany są pomijane. Pominięcie stanu powoduje utratę pewnych trajektorii, które mogą prowadzić do rozwiązania optymalnego.

TEST 1

Dane: c = 12, N = 5, czasy operacji: 7, 6, 5, 7, 8, relacja kolejności: 1-3, 1-5, 3-5.

Liczby stanów na kolejnych etapach:

- bez dominacji: 3, 7, 13, 20, 20,
- z dominację: 3, 4, 4, 3, 1.

W algorytmie bez dominacji występuje w sumie 63 stany, natomiast w algorytmie z dominacją 15 stanów, co stanowi 24%.
Czasy obliczeń:

- bez dominacji 0,840 [s]
- z dominację 0,769 [s]
- z modyfikację 0,764 [s]

Skrócenie czasu obliczeń wynosi 8% (9% - dla algorytmu z modyfikacją). Każdy algorytm daje rozwiązanie optymalne - 4 stanowiska pracy.

TEST 2

Dane: c = 12, N = 7, czasy operacji: 7, 6, 5, 7, 8, 4, 5, relacja kolejności: 1-3, 1-5, 3-5.

Liczby stanów na kolejnych etapach

- bez dominacji 5, 21, 73, 208, 480, 840, 840.
- z dominację: 5, 11, 15, 15, 11, 5, 1.

W algorytmie bez dominacji występuje w sumie 2467 stanów, natomiast w algorytmie z dominacją stanów 63 stany, co stanowi 3%.
Czasy obliczeń:

- bez dominacji 6,959 [s]
- z dominację 1,961 [s]
- z modyfikację 1,862 [s]

Skrócenie czasu obliczeń wynosi 72% (73%). Każdy algorytm daje rozwiązanie optymalne - 4 stanowiska pracy.

TEST 3

Dane: c = 12, N = 9, czasy operacji: 7, 6, 5, 7, 8, 4, 5, 5, 6, relacja kolejności: 1-3, 1-5, 3-5, 4-8.

Liczby stanów na kolejnych etapach:

- bez dominacji: 6,32, 151, 626, 530, 716, 459, 851, 851.
- z dominacją: 6, 17, 31, 41, 41, 37, 17, 6, 1.

W algorytmie bez dominacji wystąpiły 4222 stany, natomiast w algorytmie z dominacją 160 stanów. W algorytmie bez dominacji zostały pominięte niektóre stany.
Czasy obliczeń:

- bez dominacji 12,895 [s]
- z dominacją 8,156 [s]
- z modyfikacją 7,594 [s]

Skrócenie czasu obliczeń wynosi 37% (41%).
Algorytm bez dominacji stanów (suboptymalny) daje takie samo rozwiązanie jak algorytmy z dominacją, tzn. 5 stanowisk pracy.

TEST 4

Dane: c = 14, N = 9, czasy operacji: 3, 6, 4, 6, 8, 4, 5, 5, 6, relacja kolejności: 1-2, 1-3, 6-3, 2-4, 1-5, 3-5, 4-6, 5-7, 4-8.

Liczby stanów na kolejnych etapach:

- bez dominacji: 2, 3, 4, 6, 13, 22, 33, 45, 45.
- z dominacją: 2, 2, 2, 3, 4, 4, 4, 3, 1.

W algorytmie bez dominacji występuję w sumie 173 stany, natomiast w algorytmie z dominacją 25 stanów, co stanowi 14%.
Czasy obliczeń:

- bez dominacji 1,761 [s]
- z dominacją 1,379 [s]
- z modyfikacją 1,365 [s]

Skrócenie czasu obliczeń wynosi 22% (22%).
Każdy algorytm daje rozwiązanie optymalne - 4 stanowiska pracy.

TEST 5

Dane: c = 14, N = 10, czasy operacji: 3, 5, 7, 3, 8, 4, 5, 5, 6, 5, relacja kolejności 1-2, 6-3, 1-5, 4-6, 5-7, 4-8, 5-10.

Liczby stanów na kolejnych etapach:

- bez dominacji: 3, 10, 37, 139, 518, 663, 498, 818, 397, 397.
- z dominacją: 3, 7, 14, 22, 28, 30, 26, 16, 6, 1.

W algorytmie bez dominacji stanów wystąpiło 3480 stanów, natomiast w algorytmie z dominacją 153 stany.
W algorytmie bez dominacji niektóre stany zostały pominięte.
Czasy obliczeń:

- bez dominacji 11,680 [s]
- z dominacją 6,206 [s]
- z modyfikacją 5,745 [s]

Skrócenie czasu obliczeń wynosi 47% (51%).
Algorytm bez dominacji stanów (suboptymalny) daje takie samo rozwiązanie jak algorytmy z dominacją, tzn. 4 stanowiska pracy.

TEST 6

Dane: c = 14, N = 10, czasy operacji: 6, 5, 4, 7, 6, 5, 5, 6, 7, 6, relacja kolejności: 1-2, 1-3, 3-4, 3-5, 2-6, 4-7, 6-7, 5-8, 7-9, 8-9, 9-10.

Liczby stanów na kolejnych etapach:

- bez dominacji: 1, 2, 5, 12, 27, 51, 71, 71, 71, 71.
- z dominacją: 1, 2, 4, 5, 5, 4, 2, 1, 1, 1.

W algorytmie bez dominacji występuję 382 stany, natomiast w algorytmie z dominacją 26 stanów, co stanowi 7%.
Czasy obliczeń:

- bez dominacji 2,664 [s]
- z dominacją 1,458 [s]
- z modyfikacją 1,448 [s]

Skrócenie czasu obliczeń wynosi 45% (46%).
Każdy z algorytmów daje rozwiązanie optymalne - 5 stanowisk pracy.

TEST 7

Dane: c = 14, N = 11, czasy operacji: 6, 5, 5, 4, 7, 6, 5, 5, 6, 7, 6 relacja kolejności: 1-2, 1-3, 3-4, 3-5, 2-6, 4-7, 6-7, 5-8, 7-9, 8-9, 9-10, 7-11.

Liczby stanów na kolejnych etapach:

- bez dominacji: 1, 2, 5, 12, 27, 51, 77, 103, 174, 245, 245.
- z dominacją: 1, 2, 4, 5, 5, 4, 3, 2, 2, 2, 1.

W algorytmie bez dominacji występuje 930 stanów, natomiast w algorytmie z dominacją 31 stanów, co stanowi 3%.
Czasy obliczeń:

- bez dominacji 4,817 [s]
- z dominacją 1,705 [s]
- z modyfikacją 1,684 [s]

Skrócenie czasu obliczeń wynosi 65% (65%).
Każdy z algorytmów daje rozwiązanie optymalne - 6 stanowisk pracy.

TEST 8

Dane: c = 14, N = 12, czasy operacji: 6, 7, 5, 5, 4, 7, 6, 5, 5, 6, 7, 6,
relacja kolejności: 1-2, 1-3, 3-4, 3-5, 2-6, 4-7, 6-7, 5-8, 7-9, 8-9, 9-10, 7-11, 10-12.

Liczby stanów na kolejnych etapach:

- bez dominacji: 1, 2, 5, 12, 27, 51, 77, 103, 174, 245, 316, 316.
- z dominacją: 1, 2, 4, 5, 5, 4, 3, 2, 2, 2, 2, 1.

W algorytmie bez dominacji występuje 1329 stanów, natomiast w algorytmie z dominacją 33 stany, co stanowi 2%.
Czasy obliczeń:

- bez dominacji 6,677 [s]
- z dominacją 1,876 [s]
- z modyfikacją 1,865 [s]

Skrócenie czasu obliczeń wynosi 72% (72%).
Każdy z algorytmów daje rozwiązanie optymalne - 6 stanowisk pracy.

TEST 9

Dane: c = 14, N = 12, czasy operacji: 4, 6, 4, 5, 4, 7, 6, 5, 5, 6, 7, 8,
relacja kolejności: 1-2, 1-3, 2-4, 3-5, 2-6, 4-7, 6-7, 5-8, 7-9, 8-9, 9-10, 7-11, 11-12.

Liczby stanów na kolejnych etapach:

- bez dominacji: 1, 2, 5, 12, 26, 52, 84, 126, 238, 420, 602, 602.
- z dominacją: 1, 2, 4, 5, 5, 5, 4, 3, 3, 3, 2, 1.

W algorytmie bez dominacji występuje 2170 stanów, natomiast w algorytmie z dominacją 38 stanów, co stanowi 2%.
Czasy obliczeń:

- bez dominacji 9,623 [s]
- z dominacją 1,992 [s]
- z modyfikacją 1,967 [s]

Skrócenie czasu obliczeń wynosi 79% (80%).
Każdy algorytm daje rozwiązanie optymalne - 5 stanowisk pracy.

TEST 10

Dane: c = 12, N = 16, czasy operacji: 6, 8, 5, 7, 9, 3, 5, 6, 5, 4, 5, 4, 4, 5, 6, 6,
relacja kolejności: 1-2, 2-3, 1-4, 3-5, 4-6, 1-7, 5-8, 6-8, 7-8, 8-9, 8-10, 8-11, 9-12, 10-13, 12-14, 12-15, 13-15, 14-16.

Liczby stanów na kolejnych etapach:

- bez dominacji: 1, 3, 8, 19, 38, 60, 60, 60, 180, 351, 83, 172, 338, 118, 189, 189.
- z dominacją: 1, 3, 5, 6, 5, 3, 1, 1, 3, 5, 6, 6, 6, 5, 3, 1.

W algorytmie bez dominacji występuje 1869 stanów, natomiast w algorytmie z dominacją 60 stanów. W algorytmie bez dominacji niektóre stany zostały pominięte.
Czasy obliczeń:

- bez dominacji 159,653 [s]
- z dominacją 25,076 [s]
- z modyfikacją 24,113 [s]

Skrócenie czasu obliczeń wynosi 84% (85%).
Algorytmy z dominacją dają rozwiązanie optymalne (8 stanowisk pracy), natomiast algorytm bez dominacji daje rozwiązanie przybliżone (9 stanowisk pracy).

Zestawienie rezultatów przeprowadzonych testów komputerowych zamieszczono w tabeli 2.2.

W tabeli 2.2 rozwiązania suboptymalne oznaczono przez §, natomiast czasy obliczeń dla algorytmów z dominacją i modyfikacją umieszczono w nawiasach.

Jak widać z tabeli, w siedmiu testach algorytmy bez dominacji dają rozwiązania optymalne (lecz po dłuższym czasie obliczeń niż w algorytmach z dominacją). W testach 3 i 5 algorytm bez dominacji nie generuje wszystkich stanów. Pomimo to otrzymane rozwiązania są optymalne. W teście 10 algorytm bez dominacji również nie generuje wszystkich stanów. Jednakże w tym przypadku otrzymane rozwiązanie nie jest optymalne.

Tabela 2.2

Porównanie obliczeń komputerowych bez dominacji stanów oraz z dominacją stanów

Nr testu	L operacji	Bez dominacji			Z dominacją		
		L stanów	Czas [s]	L st. pracy	L stanów	Czas [s]	L st. pracy
1	5	63	0,840	4	15	0,769 (0,764)	4
2	7	2467	6,959	4	63	1,961 (1,862)	4
3	9	4222 §	12,895 §	5 §	160	8,156 (7,594)	5
4	9	173	1,761	4	25	1,379 (1,365)	4
5	10	3480 §	11,680 §	4 §	153	6,206 (5,745)	4
6	10	382	2,664	5	26	1,458 (1,448)	5
7	11	930	4,817	6	31	1,705 (1,684)	6
8	12	1329	6,677	6	33	1,876 (1,865)	6
9	12	2170	9,623	5	38	1,992 (1,967)	5
10	16	1869 §	159,653 §	9 §	60	25,076) (24,113)	8

Z przeprowadzonych testów wynika, że stosowanie dominacji zmniejsza liczbę stanów i skraca czas obliczeń. Przeciętne skrócenie czasu obliczeń wyniosło 52% (pomijając testy 3, 5, 10). Ponadto w algorytmach z dominację występuje przeciętnie 8% liczby stanów występujęcych w algorytmach bez dominacji. Z powyższych liczb można wycięgnąć wniosek, że w regułach dominacji należy dążyć do skracania czasu porównywania dwóch stanów.

2.6. ANALIZA ALGORYTMÓW

Do rozwiązania klasycznego problemu BLM wykorzystano cztery algorytmy oparte na metodzie programowania wieloetapowego (binarny, szeregowania, przydziału oraz harmonogramowania). Klasyczny problem BLM (w wersji decyzyjnej) oraz jego uogólnienia należę do klasy problemów NP-zupełnych w sensie złożoności obliczeniowej. Z tego względu algorytmy przeglądowe dla rozwiązania tego problemu mają wykładniczę złożoność obliczeniową. Analiza algorytmów programowania wieloetapowego ma na celu wykazanie pewnych elementów mających wpływ na efektywność tych algorytmów. Ponadto zostanie uzasadniony wybór algorytmu harmonogramowania dla rozwiązania - w dalszej części rozprawy - uogólnionych problemów BLM oraz problemów AO i AOiZ.

2.6.1. Efektywność algorytmów

Efektywność algorytmów programowania wieloetapowego zależy od liczby wygenerowanych oraz liczby wyeliminowanych stanów nieperspektywicznych. Z tego względu przedyskutujemy pewne elementy generowania oraz eliminowania stanów.

Załóżmy, że w stanie $P^{\eta-1,\lambda}$ podejmowana jest decyzja o realizacji operacji ω_n. Niech Ω_μ będzie ostatnim podzbiorem operacji utworzonym w stanie $P^{\eta-1,\lambda}$.

LEMAT 2.1

Jeżeli operacja ω_n może być ulokowana w Ω_μ, to przydzielenie tej operacji do Ω_m, $\mu < m$, daje stan nieperspektywiczny.

D o w ó d: Załóżmy, że przydzielajęc ω_n do Ω_μ otrzymujemy stan P^{η,l_1}, natomiast przydzielajęc ω_n do Ω_m dostajemy stan P^{η,l_2}. Zauważmy, że:

$$\Omega^{\eta-1,\lambda} \cup \omega_n = \Omega^{\eta,l_1} = \Omega^{\eta,l_2} \tag{2.51}$$

Ponadto:

$$T^{\eta,l_1} = T^{\eta-1,\lambda} + \vartheta_n \tag{2.52}$$

oraz:

$$T^{\eta,l_2} = (m-1)c + \vartheta_n \tag{2.53}$$

Z (2.52) i (2.53) wynika, że:

$$T^{\eta,l_1} - T^{\eta,l_2} = T^{\eta-1,\lambda} - (m-1)c \tag{2.54}$$

Ponieważ

$$T^{\eta-1,\lambda} < \mu \cdot c \leq (m-1)c \tag{2.55}$$

zatem z (2.54) i (2.53) otrzymujemy:

$$T^{\eta,l_1} < T^{\eta,l_2} \tag{2.56}$$

Uwzględniając (2.51) i (2.56) w (2.24) widzimy, że stan P^{η,l_1} dominuje nad stanem P^{η,l_2}. □

Zatem stosując kryterium minimalizacji liczby stanowisk pracy, w procedurze generowania stanów należy przydzielać operację ω_n do najwcześniej dopuszczalnego podzbioru Ω_k, $k \leq K$.

Eliminacja stanów przez dominację wymaga porównania stanu P^{η,l_2}, wygenerowanego jako ostatni, ze stanami P^{η,l_1}, które zostały wygenerowane wcześniej. Zwiększenie efektywności algorytmu można uzyskać przez zmniejszenie liczby tych porównań.

LEMAT 2.2

Jeżeli stan P^{η,l_1} otrzymano z $P^{\eta-1,\lambda}$ przez przydzielenie ω_ν, natomiast P^{η,l_2} otrzymano również z $P^{\eta-1,\lambda}$, lecz przez przydzielenie ω_n, gdzie $\omega_\nu \neq \omega_n$, to pomiędzy stanami P^{η,l_1} i P^{η,l_2} nie zachodzi relacja dominacji.

D o w ó d: Ponieważ:

$$\Omega^{\eta,l_1} = \Omega^{\eta-1,\lambda} \cup \omega_\nu \tag{2.57}$$

oraz

$$\Omega^{\eta,l_2} = \Omega^{\eta-1,\lambda} \cup \omega_n \tag{2.58}$$

a ponadto $\omega_\nu \neq \omega_n$, zatem:

$$\Omega^{\eta,l_1} \neq \Omega^{\eta,l_2} \tag{2.59}$$

Uwzględniając (2.59) w (2.24) widzimy, że warunek dominacji stanów nie jest spełniony. □

Liczbę porównań stanów można również zmniejszyć wprowadzając porządek leksykograficzny w zbiorze $\mathcal{L}_\eta$ stanów aktywnych η-tego etapu.

Definicja 2.3

Stan P^{η,l_1} poprzedza leksykograficznie stan P^{η,l_2} (co oznaczamy $P^{\eta,l_1} \prec P^{\eta,l_2}$), jeżeli jest spełniony warunek:

$$\exists_{\nu} \forall_{n<\nu} \left\{ \left[(p_n^{\eta,l_1} \neq 0) \Leftrightarrow (p_n^{\eta,l_2} \neq 0) \right] \vee \left[(p_n^{\eta,l_1} = 0) \Leftrightarrow (p_n^{\eta,l_2} = 0) \right] \right\} \wedge$$

$$\wedge \left[(p_\nu^{\eta,l_1} \neq 0) \Leftrightarrow (p_\nu^{\eta,l_2} = 0) \right] \tag{2.60}$$

Definicja 2.4

Stany P^{η,l_1} i P^{η,l_2} są alternatywne, jeżeli jest spełniony warunek:

$$\neg (P^{\eta,l_1} \prec P^{\eta,l_2}) \wedge \left[\neg (P^{\eta,l_2} \prec P^{\eta,l_1}) \right] \tag{2.61}$$

LEMAT 2.3

W zbiorze $\mathcal{L}_\eta$ stanów aktywnych nie ma stanów alternatywnych.

D o w ó d (nie wprost): Przypuśćmy, że w zbiorze $\mathcal{L}_\eta$ znajdują się stany alternatywne P^{η,l_1} i P^{η,l_2}. Wówczas zgodnie z (2.60) i (2.61) musi być spełniony warunek:

$$\forall_{1 \leq n \leq N} \left[(p_n^{\eta,l_1} \neq 0) \Leftrightarrow (p_n^{\eta,l_2} \neq 0) \right] \vee \left[(p_n^{\eta,l_1} = 0) \Leftrightarrow (p_n^{\eta,l_2} = 0) \right] \tag{2.62}$$

Uwzględniając (2.62) w (2.4) widzimy, że:

$$\Omega^{\eta,l_1} = \Omega^{\eta,l_2} \tag{2.63}$$

Z kolei, jeżeli zachodzi (2.63), to z (2.24) wynika, że jeden ze stanów P^{η,l_1} lub P^{η,l_2} jest zdominowany - a więc nie może należeć do zbioru $\mathcal{L}_\eta$. □

A zatem zbiór $\mathcal{L}_\eta$ jest uporządkowany liniowo. Stosowanie reguły dominacji polega na sprawdzeniu, czy w zbiorze $\mathcal{L}_\eta$ znajduje się stan P^{η,l_1} alternatywy do stanu P^{η,l_2} (który został wygenerowany jako ostatni).

Ponieważ zbiór $\mathcal{E}_\eta$ jest uporządkowany liniowo, nie trzeba sprawdzać wszystkich stanów, od $l=1$ do $l=L_\eta$. Stosując np. podział połówkowy, można w pierwszym kroku porównać stan P^{η,l_2} ze stanem o numerze $l = \left[\frac{1}{2} L_\eta\right]^+$. Jeżeli $P^{\eta,l_2} \prec P^{\eta,l}$, to pozostają do sprawdzenia stany o numerach $\lambda < l$, natomiast stanów o numerach $\lambda' > l$ sprawdzać nie trzeba (bo na pewno nie są alternatywne ze stanem P^{η,l_2}). W ten sposób stosując kolejne podziały połówkowe zmniejszamy liczbę porównań stanów, (np. jeżeli $L_\eta = 2^k$, to porównań jest k+1). Jeżeli w zbiorze $\mathcal{E}_\eta$ zostanie wyznaczony stan P^{η,l_2} alternatywny do P^{η,l_2}, to jeden z tych stanów zostanie wyeliminowany. W przypadku przeciwnym stan P^{η,l_2} wchodzi do zbioru $\mathcal{E}_\eta$.

Efektywność algorytmów programowania wieloetapowego zależy również od stosowanych reguł wyboru stanów aktywnych. Dla reguły FIFO eliminowanie stanów przez sondowanie nie ma znaczenia, ponieważ stany końcowe są znane dopiero wtedy, gdy zbiory $\mathcal{E}_\eta$ (dla $\eta < N-1$) są puste. Z kolei dla reguły LIFO nie można stosować eliminacji stanów przez dominację (zgodnie z lematem 2.2). Reguła LLB pozwala eliminować stany przez dominację oraz sondowanie.

2.6.2. Rozwiązanie przybliżone

Jeżeli w BLM nie zostaną wygenerowane wszystkie stany końcowe, to otrzymujemy jedynie rozwiązanie przybliżone. Rozwiązanie to reprezentuje aktualnie najlepszy stan końcowy P^a, o wartości V^a. Załóżmy, że δ jest błędem bezwzględnym, natomiast ε - błędem względnym. Stan P^a daje rozwiązanie przybliżone z błędem:

$$\delta = V^a - V^o \leq V^a - b^o \tag{2.64}$$

lub:

$$\varepsilon = \frac{V^a - V^o}{V^o} \leq \frac{V^a}{b^o} - 1 \tag{2.65}$$

gdzie: b^o - najmniejsze dolne ograniczenie wartości V^o stanu optymalnego P^o.

LEMAT 2.4

Najmniejsze dolne ograniczenie b^o otrzymujemy ze stanu $P^{0,1}$, tzn.:

$$b^o = \left[\frac{1}{c} \sum_{n=1}^{n=N} \nu_n^{\eta}\right]^+ \tag{2.66}$$

D o w ó d: Dla każdego stanu $P^{\eta,l} \neq P^{0,1}$ otrzymamy na podstawie (2.16):

$$b^{\eta,l} = \left[\frac{1}{c}\left(T^{\eta,l} + \sum_{\omega_n \notin \Omega^{\eta,l}} \nu_n^{\eta}\right)\right]^+ \geq \left[\frac{1}{c}\left(\sum_{\omega_n \in \Omega^{\eta,l}} \nu_n^{\eta} + \sum_{\omega_n \notin \Omega^{\eta,l}} \nu_n^{\eta}\right)\right]^+ = b^o \tag{2.67}$$

ponieważ $T^{\eta,l}$ uwzględnia luzy czasu na stanowiskach pracy. □

LEMAT 2.5

Jeżeli $\Omega^{\eta_1,l_1} \subset \Omega^{\eta_2,l_2}$, to $b^{\eta_1,l_1} \leq b^{\eta_2,l_2}$

D o w ó d: Dla stanu P^{η_1,l_1} napiszemy:

$$b^{\eta_1,l_1} = \left[\frac{1}{c}\left(T^{\eta_1,l_1} + \sum_{\omega_n \notin \Omega^{\eta_1,l_1}} \nu_n^{\eta}\right)\right]^+ \tag{2.68}$$

Oznaczmy przez Ω' zbiór operacji zdefiniowany następująco:

$$\Omega' = \Omega^{\eta_2,l_2} \setminus \Omega^{\eta_1,l_1} \tag{2.69}$$

Wyrażenie (2.68) można przekształcić do postaci:

$$b^{\eta_1,l_1} = \left[\frac{1}{c}\left(T^{\eta_1,l_1} + \sum_{\omega_n \in \Omega'} \nu_n^{\eta} + \sum_{\omega_n \notin \Omega^{\eta_2,l_2}} \nu_n^{\eta}\right)\right]^+ \tag{2.70}$$

Zauważmy, że:

$$T^{\eta_1,l_1} + \sum_{\omega_n \in \Omega'} \nu_n^{\eta} \leq T^{\eta_2,l_2} \tag{2.71}$$

A zatem podstawiając (2.71) do (2.70) otrzymamy ostatecznie:

$$b^{\eta_1,l_1} \leq \left[\frac{1}{c}\left(T^{\eta_2,l_2} + \sum_{\omega_n \notin \Omega^{\eta_2,l_2}} \nu_n^{\eta}\right)\right]^+ = b^{\eta_2,l_2} \tag{2.72}$$

□

Ponieważ dobre oszacowanie b^o powinno być bliskie v^o, zatem w obliczeniach przyjmujemy jako b^o najmniejsze dolne ograniczenie wyznaczone dla stanów aktywnych.

Ograniczenie pamięci komputerowej nie jest istotne, ponieważ można stosować regułę podziału częściowego. Wówczas z każdym stanem aktywnym $P^{\eta,l}$ związany jest jego indeks $I^{\eta,l}$. Indeks ten jest największym numerem operacji, którą wykorzystano w generowaniu stanów ze stanu $P^{\eta,l}$. Jeżeli $I^{\eta,l} = 0$, to ze stanu $P^{\eta,l}$ nie wygenerowano żadnego stanu. W przypadku gdy $I^{\eta,l} = N$, stan $P^{\eta,l}$ przestaje być aktywny. A zatem $P^{\eta,l}$ może być wielokrotnie wybrany do generowania stanów. W regule podziału częściowego należy uwzględnić dodatkowe warunki:

$$I^{\eta-1,\alpha} < n \leq N \tag{2.73a}$$

Ponadto ze stanu $P^{\eta-1,\alpha}$ można wygenerować najwyżej δl stanów, gdzie:

$$\delta l = L_\eta - |\mathcal{L}_\eta| \tag{2.73b}$$

przy tym L_η jest liczbą dopuszczalnę stanów w zbiorze $\mathcal{L}_\eta$.

2.6.3. Wybór algorytmu

Przedstawione algorytmy programowania wieloetapowego można przeanalizować z punktu widzenia możliwości rozwiązania uogólnionych problemów BLM, AO oraz AOiZ. W ogólnym przypadku w rozwiązywaniu tych problemów potrzebna jest znajomość historii podejmowanych decyzji. Z tego względu algorytm binarny nie może być wykorzystany. Algorytm przydziału nie pozwala rozwiązać problemów AO w liniach z równoległymi stanowiskami pracy oraz pewnych uogólnionych modeli BLM (np. montaż przestrzenny). Algorytm szeregowania pozwala rozwiązać problemy, które będą sformułowane w dalszej części rozprawy. Jednakże w przypadku ograniczeń chwil zakończenia operacji, w algorytmie tym są potrzebne pewne dodatkowe obliczenia. Wszystkie z wymienionych problemów można rozwiązać bez dodatkowych obliczeń za pomocą algorytmu harmonogramowania. Z tego względu w dalszej części rozprawy będzie wykorzystany algorytm harmonogramowania, który w skrócie będzie nazywany algorytmem programowania wieloetapowego.

3. UOGÓLNIONE MODELE BLM

3.1. WSTĘP

W rozdziale tym zostanę przedstawione pewne uogólnione modele BLM. Uogólnienia będą dotyczyć dodatkowych ograniczeń oraz kryterium optymalizacji. Wyróżnione zostaną ograniczenia wynikające z:

- wykluczania pewnych podzbiorów operacji na jednym stanowisku pracy,
- przynależności operacji do pewnego podzbioru zwanego enklawą,
- lokalizacji operacji w montowanym obiekcie,
- kosztów wykonywania operacji.

W modelach BLM zakłada się, że podzbiory operacji tworzące stanowiska pracy są zlokalizowane na kolejnych stacjach montażowych.

Rozważmy linię montażową składającę się z K stacji. Tak jak w modelu klasycznym założymy, że dane są: zbiór Ω operacji ω_n, $n = 1,\ldots,N$, wektor Θ czasów operacji ϑ_n, macierz Γ relacji kolejności operacji oraz cykl c. Przez t_n oznaczymy chwilę zakończenia montażu operacji ω_n na linii.

Do rozwiązania uogólnionych problemów BLM wykorzystamy algorytm programowania wieloetapowego (harmonogramowania). W algorytmie tym przyjmiemy stan procesu decyzyjnego taki sam jak dla klasycznego modelu BLM. Pozostałe elementy algorytmu będą bardziej ogólne niż dla modelu klasycznego.

3.2. DODATKOWE PARAMETRY OPERACJI

Rozważmy uogólniony model BLM, w którym oprócz czasu istnieją inne parametry operacji. Dla wykonania operacji potrzebne są detale i narzędzia. Detale znajdują się w kontenerach o odpowiednich gabarytach. Z uwagi na ograniczone wymiary stacji montażowej oraz gabaryty kontenerów z detalami, pewne podzbiory operacji nie mogą tworzyć jednego stanowiska pracy. Ten sam efekt powstaje, gdy detale z uwagi na swój ciężar wymagają od montera znacznej pracy dla ich przeniesienia z kontenera do montowanego obiektu. Wprowadzając ograniczenie wielkości tej pracy eliminujemy pewne podzbiory operacji jako stanowiska pracy. Na stanowisku pracy może być ograniczona liczba narzędzi lub płaca montera może być zależna od ceny operacji (ograniczona od dołu). Również w tych (i innych tego typu) przypadkach pewne podzbiory operacji nie mogą tworzyć stanowisk pracy.

3.2.1. Model matematyczny

Cechę charakterystyczną przedstawionych wyżej ograniczeń jest ograniczenie sumy parametrów operacji na jednym stanowisku pracy. Załóżmy, że dodatkowe parametry operacji dane są macierzą:

$$\Delta = \left[\delta_{n,j}\right]_{\substack{n=1,\dots,N \\ j=1,\dots,J}} \tag{3.1}$$

gdzie: $\delta_{n,j}$ - j-ty parametr operacji ω_n, J - liczba parametrów operacji.

Ponadto przyjmiemy, że dany jest wektor D dodatkowych ograniczeń liniowych:

$$D = \left[d_{k,j}\right]_{\substack{k=1,\dots,K \\ j=1,\dots,J}} \tag{3.2}$$

gdzie: $d_{k,j}$ - dopuszczalna suma parametrów j-tego typu dla operacji na k-tym stanowisku pracy.

Tak więc dodatkowe ograniczenia można zapisać w postaci:

$$\sum_{\omega_n \in \Omega_k} \delta_{n,j} \leq d_{k,j} \quad \begin{matrix} k=1,\dots,K \\ j=1,\dots,J \end{matrix} \tag{3.3}$$

Z (3.3) widzimy, że są to ograniczenia liniowe.

Dla przypomnienia, w modelu klasycznym BLM występują jedynie ograniczenia montażu kompletnego oraz:

- sumy czasów

$$\sum_{\omega_n \in \Omega_k} \vartheta_n \leq c \tag{3.4}$$

- kolejności

$$\bigwedge_n \bigwedge_\nu \bigvee_{\mathcal{H} \leq k} (\omega_n \in \Omega_k) \wedge (\tau_{\nu,n}=1) \Rightarrow (\omega_\nu \in \Omega_\mathcal{H}) \tag{3.5}$$

Natomiast kryterium optymalizacji ma postać:

$$Q = \max_{1 \leq n \leq N} \left[\frac{1}{c} t_n\right]^+ \longrightarrow \min \tag{3.6}$$

Warto zauważyć, że ograniczenie sumy czasów (3.4) jest liniowe. A zatem rozpatrywany model zawiera jedynie dodatkowe ograniczenia liniowe. Ograniczenie (3.3), (3.4) i (3.5) przedstawiamy w postaci wyrażeń, w których występuje niewiadoma t_n, bowiem:

$$\bigwedge_n \left\{\left[\frac{1}{c} t_n\right]^+ = k\right\} \Rightarrow (\omega_n \in \Omega_k) \tag{3.7}$$

3.2.2. Algorytm

W algorytmie programowania wielostopowego dla rozważanego uogólnionego modelu stan i wartość stanu definiowane są analogicznie jak dla modelu klasycznego.

G e n e r o w a n i e s t a n ó w

Procedura generowania stanów jest taka sama jak dla modelu klasycznego BLM, z tym że:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \vartheta_n, \text{ jeśli } \bigwedge_{1 \leq j \leq J} \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c}(T^{\eta-1,\lambda} + \vartheta_n)\right]^+ \wedge \\ \qquad \wedge (D_j^{\eta-1,\lambda} + \delta_{n,j} \leq d_{k,j}) \\ \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ \cdot c - T^{\eta-1,\lambda} + \vartheta_n, \text{ w przypadku przeciwnym} \end{cases} \tag{3.8}$$

przy tym

$$\sum_{\omega_i \in \Omega_k} \delta_{i,j} = D_j^{\eta-1,\lambda} \tag{3.9}$$

oraz

$$\bigwedge_i \left\{\left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ = k\right\} \wedge \left\{\left[\frac{1}{c} p_i^{\eta-1,\lambda}\right]^+ = k\right\} \Rightarrow (\omega_i \in \Omega_k) \tag{3.9a}$$

Sumy parametrów operacji na ostatnim stanowisku pracy można również wyznaczyć równocześnie z generowaniem stanu, na podstawie wzoru rekurencyjnego:

$$D_j = \begin{cases} D_j^{\eta-1,\lambda} + \delta_{n,j}, \text{ jeśli } \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c} t_n\right]^+ \\ \delta_{n,j}, \text{ w przypadku przeciwnym} \end{cases} \tag{3.10}$$

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Dla uogólnionego modelu BLM z dodatkowymi parametrami operacji stany nieperspektywiczne eliminuje się za pomocą reguł: dominacji, wyczerpywania oraz sondowania.

TWIERDZENIE 3.1

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeśli jest spełniony warunek:

$$\bigvee_{1\leq j\leq J} (\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge (D_j^{\eta,l_1} \leq D_j^{\eta,l_2}) \quad (3.11)$$

Dowód tego twierdzenia można przeprowadzić analogicznie jak w modelu klasycznym BLM.

TWIERDZENIE 3.2

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$\bigvee_{1\leq j\leq J} (D_j^{\eta,l} + \sum_{\omega_n \notin \Omega^{\eta,l}} \delta_{n,j} > \sum_{\varkappa=K}^{\varkappa=K} {}_{\eta,l} d_{\varkappa,j}) \vee (b^{\eta,l} \geq V^a) \quad (3.12)$$

Jeżeli jest spełniony pierwszy składnik alternatywy (3.12), to stan $P^{\eta,l}$ jest wyczerpany (nie daje rozwiązania dopuszczalnego). W przypadku gdy spełniony jest drugi składnik alternatywy, każdy stan końcowy otrzymany z $P^{\eta,l}$ jest gorszy niż stan aktualnie najlepszy P^a. A zatem warunek (3.12) jest połączeniem reguły wyczerpywania i sondowania.

3.3. ENKLAWY OPERACJI

Rozważmy przypadek występowania w procesie montażu pewnych podzbiorów operacji, zwanych enklawami. Operacje należące do enklawy muszą być wykonane w sekwencji, która nie zawiera innych operacji. Wyróżnimy dwa typy enklaw: podzielne oraz niepodzielne. Operacje należące do enklawy podzielnej mogą być wykonane na sąsiednich stanowiskach pracy przez różnych monterów. W przypadku enklawy niepodzielnej, wszystkie należące do niej operacje są wykonywane przez jednego montera, to znaczy nie wyróżniamy podzbiorów operacji w ramach enklawy.

W balansowaniu linii montażowej enklawa niepodzielna jest traktowana jako jedna operacja, której czas wykonania może być większy od cyklu.

Jeżeli czas wykonania enklawy niepodzielnej jest większy niż cykl, to monter pracuje na kilku kolejnych stacjach. Powstaje wówczas tzw. wielokrotne stanowisko pracy, czyli wielokrotność stacji montażowej. Liczba monterów pracujących na wielokrotnym stanowisku pracy jest równa wielokrotności tego stanowiska (liczbie zajmowanych stacji montażowych). Również cykl na takim stanowisku jest wielokrotnością cyklu podstawowego.

Na stanowisku wielokrotnym monter obsługuje tylko niektóre (krotne) obiekty. Wyróżnienie operacji w enklawie niepodzielnej wynika z odmiennych typów detali i narzędzi potrzebnych do wykonania tych operacji. W przypadku enklawy podzielnej operacje są wykonywane na kolejnych stacjach przez różnych monterów. Inaczej mówiąc, na każdym stanowisku znajduje się jeden monter, który obsługuje każdy kolejny obiekt.

Wprowadzenie enklaw w procesie montażu jest uzasadnione względami technologicznymi (konstrukcję linii dostosowaną do wykonania pewnych operacji, kontrolą wykonania montażu układu funkcjonalnego itp.).

Z uwagi na ograniczenia grupowania operacji wyróżnimy następujące przypadki:

- na pewnych stacjach są realizowane wyłącznie enklawy,
- enklawy mogą być poprzedzane innymi operacjami na swych stacjach,
- po enklawach (na ich stacjach) mogą następować inne operacje.

W dalszym ciągu uwzględnimy wyszczególnione wyżej przypadki dla enklaw podzielnych oraz niepodzielnych.

3.3.1. Model matematyczny

Uogólniony model BLM z enklawami operacji zawiera wszystkie elementy modelu klasycznego. Ponadto w zbiorze Ω wyróżniamy podzbiory E_r, $r = 1,\ldots,R$, zwane enklawami. Zakładamy, że enklawy spełniają następujące warunki:

$$E_p \cap E_r = 0 \qquad p \neq r \quad (3.13a)$$

$$\bigcup_{r=1}^{r=R} E_r \subset \Omega \quad (3.13b)$$

gdzie: R - liczba enklaw.
Przynależność operacji ω_n do enklawy E_r jest opisana wektorem:

$$A = [a_n]_{n=1,\ldots,N} \quad (3.14)$$

Elementy tego wektora mają następujące znaczenie:

$$a_n = \begin{cases} r, & \text{jeśli } \omega_n \text{ należy do } E_r \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{3.14a}$$

W dalszym ciągu przez E_r oznaczymy enklawę niepodzielną, a przez E_ρ - podzielną.

Dopuszczalny balans linii musi spełniać następujące ograniczenia:

- kolejności

$$\bigwedge_{\nu} \bigwedge_{n} (g_{\nu,n} = 1) \Rightarrow (t_\nu \leq t_n - \vartheta_n) \tag{3.15}$$

- niepodzielności enklawy E_r

$$\bigvee_{n} (a_n = r) \Rightarrow \left\{ k_r^1 = \left[\frac{1}{c}(t_n - \vartheta_n)\right]^+ \right\} \wedge \left\{ k_r^2 = \left[\frac{1}{c} t_n\right]^+ \right\} \tag{3.16}$$

gdzie: k_r^1, (k_r^2) - numer początkowej (końcowej) stacji, na której jest wykonywana niepodzielna enklawa E_r;

- niepodzielności operacji ω_n

$$\bigwedge_{n} (a_n \neq r) \Rightarrow \left\{ k_n = \left[\frac{1}{c} t_n\right]^+ = \left[\frac{1}{c}(t_n - \vartheta_n)\right]^+ \right\} \tag{3.17}$$

- sekwencji operacji podzielnej enklawy E_ρ

$$\bigwedge_{\nu} (a_\nu \neq \rho) \Rightarrow \left\{ \left\{ \left[\frac{1}{c} t_\nu\right]^+ \leq k_\rho^1 \right\} \vee \left\{ \left[\frac{1}{c} t_\nu\right]^+ \geq k_\rho^2 \right\} \right\} \tag{3.18}$$

przy tym:

$$k_\rho^1 = \min_{\omega_n \in E_\rho} \left[\frac{1}{c} t_n\right]^+ \tag{3.19a}$$

$$k_\rho^2 = \max_{\omega_n \in E_\rho} \left[\frac{1}{c} t_n\right]^+ \tag{3.19b}$$

Jeżeli na stacjach o numerach spełniających warunek:

$$k_\rho^1 \leq k \leq k_\rho^2 \tag{3.20}$$

są wykonywane wyłącznie operacje $\omega_n \in E_\rho$, to w (3.18) odpowiednie nierówności są ostre.

Analogiczny warunek można przytoczyć dla niepodzielnej enklawy E_r:

$$\bigwedge_{\nu} (a_\nu \neq r) = \left\{ \left\{ \left[\frac{1}{c} t_\nu\right]^+ \leq k_r^1 \right\} \vee \left\{ \left[\frac{1}{c} t_\nu\right]^+ \geq k_r^2 \right\} \right\} \tag{3.21}$$

Jednakże w tym przypadku nie chodzi o sekwencję operacji, ponieważ enklawa E_r jest traktowana jako jedna operacja. Jeżeli żadna operacja nie może poprzedzać enklawy E_r i następować po E_r na jej stacjach, to odpowiednie nierówności w (3.21) są ostre.

Jako kryterium optymalizacji BLM z enklawami operacji przyjmujemy minimalizację liczby stanowisk pracy. W sformułowanym modelu matematycznym występuje zatem N niewiadomych, którymi są chwile t_n.

3.3.2. Algorytm dla enklaw podzielnych

W procesie z enklawami podzielnymi cykl spełnia warunek:

$$\max_{1 \leq n \leq N} \vartheta_n \leq c < \sum_{n=1}^{n=N} \vartheta_n \tag{3.22}$$

Załóżmy, że na stacjach wyodrębnionych dla enklawy podzielnej nie mogą znajdować się operacje nie należące do enklawy.

Do rozwiązania problemu BLM z podzielnymi enklawami operacji wykorzystamy algorytm programowanie wieloetapowego (harmonogramowania). Stan i wartość stanu zdefiniujemy tak jak dla klasycznego problemu BLM.

G e n e r o w a n i e s t a n ó w

Generowanie stanów prowadzi się wg następujących procedur:

- dla stanu $P^{0,1}$

$$\bigwedge_{n} \bigvee_{1 \leq \nu \leq N} (g_{\nu,n} = 0) \Rightarrow (P = P^{0,1} + \Delta P) \tag{3.23}$$

przy tym

$$\Delta p_i = \begin{cases} t_n, & \text{dla } i = n \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{3.23a}$$

oraz

$$t_n = \vartheta_n \tag{3.23b}$$

- jeżeli ostatnią operację przydzieloną w wybranym stanie $P^{\eta-1,\lambda}$ jest ω_μ, która nie należy do żadnej enklawy, to jako następną można przydzielić operację, która należy lub nie należy do enklawy:

$$\forall_n \forall_{\nu} \Big(\max_{1\le i\le N} p_i^{\eta-1,\lambda} = p_{\mu}^{\eta-1,\lambda} \Big) \wedge (a_\mu=0) \wedge (p_n^{\eta-1,\lambda}=0) \wedge$$

$$\wedge \Big[(\vartheta_{\nu,n}=1) \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0) \Big] \Rightarrow (P=P^{\eta-1,\lambda} + \Delta P) \qquad (3.24)$$

Elementy wektora ΔP są określone przez (3.23a).
W tym przypadku dla ω_n otrzymamy:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + v_n^{\eta}, \text{ jeśli } (a_n=0) \wedge \left\{ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ = \left[\frac{1}{c} (T^{\eta-1,\lambda} + v_n^{\eta}) \right]^+ \right\} \\ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ \cdot c + v_n^{\eta}, \text{ w przypadku przeciwnym} \end{cases} \qquad (3.25)$$

- jeżeli ostatnią operację przydzieloną w wybranym stanie $P^{\eta-1,\lambda}$ jest ω_μ która należy do enklawy E_ρ, to jako następną przydzielamy operację ω_n, która również należy do enklawy E_ρ (jeżeli taka operacja istnieje).
W przypadku przeciwnym przydzielamy operację, która nie należy do enklawy E_ρ

$$\forall_n \forall_{\nu} \forall_j \Big(\max_{1\le i\le N} p_i^{\eta-1,\lambda} = p_{\mu}^{\eta-1,\lambda} \Big) \wedge (a_\mu=\rho) \wedge (p_n^{\eta-1,\lambda}=0) \wedge$$

$$\wedge \Big\{ (a_n=\rho) \vee \Big[(a_j=\rho) \Rightarrow (p_j^{\eta-1,\lambda} > 0) \Big] \Big\} \wedge \Big[(\vartheta_{\nu,n}=1) \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0) \Big] \Rightarrow$$

$$\Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \qquad (3.26)$$

Elementy wektora ΔP są określone przez (3.23a), przy czym:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + v_n^{\eta}, \text{ jeśli } (a_n=\rho) \wedge \left\{ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ = \left[\frac{1}{c} (T^{\eta-1,\lambda} + v_n^{\eta}) \right]^+ \right\} \\ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ \cdot c + v_n^{\eta}, \text{ w przypadku przeciwnym.} \end{cases} \qquad (3.27)$$

Gdy $a_n=\rho$, to procedurę (3.26) stosujemy tylko do operacji należących do enklawy E_ρ.

Jeżeli na stacjach wyodrębnionych dla enklaw mogą się znajdować operacje nie należące do enklawy, to zmieniają się jedynie wzory (3.25) i (3.27). Pomijając w (3.25) warunek $a_n=0$, otrzymujemy przypadek poprzedzania enklawy E_ρ przez operacje, które do niej nie należą. Pomijając z kolei

w (3.27) $a_n=\rho$ otrzymujemy możliwość grupowania operacji należących do enklawy E_ρ z operacjami nie należącymi do niej - na ostatniej stacji tej enklawy.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Dla problemu BLM z enklawami podzielnymi stany nieperspektywiczne można eliminować za pomocą reguł sondowania lub dominacji. Reguła sondowania jest taka sama jak dla klasycznego problemu BLM. Reguła dominacji jest oparta na następującym twierdzeniu:

TWIERDZENIE 3.3

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \le T^{\eta,l_2}) \wedge (a_{\mu_1}^{\eta,l_1} = a_{\mu_2}^{\eta,l_2}) \qquad (3.28)$$

gdzie: $a_\mu^{\eta,l}$ - kod przyporządkowania operacji ω_μ do enklawy, w stanie $P^{\eta,l}$. Symbolem ω_μ oznaczono ostatnią operację przydzieloną w stanie $P^{\eta,l}$.

Dowód tego twierdzenia pominiemy, gdyż jest analogiczny jak dowód twierdzenia o dominacji stanów dla modelu klasycznego BLM. Warto jedynie zaznaczyć, że ostatnie operacje w stanach P^{η,l_1} i P^{η,l_2} mogą być różne, lecz muszą należeć do tej samej enklawy.

3.3.3. Algorytm dla enklaw niepodzielnych

W problemie BLM enklawę niepodzielną E_r traktujemy jak jedną operację, a zatem czas wykonania tej operacji może przekraczać cykl c.

Załóżmy, że na stacjach montażowych wyodrębnionych dla enklawy E_r nie mogą być wykonywane inne opracje. Dla rozwiązania problemu BLM z enklawami niepodzielnymi wykorzystamy algorytm programowania wieloetapowego (harmonogramowania). W algorytmie tym - w stosunku do problemu z enklawami podzielnymi - inne są jedynie procedura generowania oraz reguła dominacji stanów.

G e n e r o w a n i e s t a n ó w

Procedura generowania stanów ma postać:

$$\forall_n \forall_{\nu} (p_n^{\eta-1,\lambda}=0) \wedge \Big[(\vartheta_{\nu,n}=1) \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0) \Big] \Rightarrow (P=P^{\eta-1,\lambda} + \Delta P) \qquad (3.29)$$

Elementy wektora ΔP są określone przez (3.23a), przy czym:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \vartheta_n^{\eta}, \text{ jeśli } (a_{\mu}^{\eta} \neq r) \wedge (a_n^{\eta} \neq r) \wedge \left\{ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ = \left[\frac{1}{c} (T^{\eta-1,\lambda} + \vartheta_n^{\eta}) \right]^+ \right\} \\ \left[\frac{1}{c} T^{\eta-1,\lambda} \right]^+ \cdot c + \vartheta_n^{\eta}, \text{ w przypadku przeciwnym.} \end{cases} \quad (3.30)$$

Stacje, na których są wykonywane enklawy, wyznaczamy ze wzoru (3.16).

Jeżeli w (3.30) pominiemy warunek $a_{\mu}^{\eta} \neq r$, to otrzymamy przypadek, w którym po enklawie E_r mogą być wykonywane inne operacje - ne tej samej stacji. W przypadku gdy w (3.30) pominiemy warunek $a_n^{\eta} \neq r$, to otrzymamy sytuację, gdy przed enklawę można przydzielić inne operacje, na tę samą stację.

E l i m i n o w a n i e e t a n ó w n i e p e r s p e k t y w i c z n y c h

Reguła dominacji etanów jest oparta na następującym twierdzeniu:

TWIERDZENIE 3.4

Sten P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge \left[(a_{\mu_1}^{\eta,l_1} \neq 0) \Leftrightarrow (a_{\mu_2}^{\eta,l_2} \neq 0) \right] \quad (3.31)$$

Dowód tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia o dominacji dla modelu klasycznego BLM. Warto zaznaczyć, że jeżeli ostatnie operacja przydzielona w stanie P^{η,l_1} jest enklawą, to ostatnia operacja przydzielone w etanie P^{η,l_2} też musi być enklawą (niekoniecznie tę samą). Warunek ten jest istotny tylko w przypadku, gdy po enklawie nie można przydzielić innych operecji do tej samej stacji, w przypadku przeciwnym ostatni człon koniunkcji (3.31) pomijamy.

3.4. MONTAŻ PRZESTRZENNY - FRONTY PRACY

W rozważanych dotychczas modelach BLM nie uwzględniono lokalizacji monterów, narzędzi i detali. Jeżeli czasy przemieszczeń pomiędzy wyróżnionymi w montażu punktami nie są pomijalnie małe, to proces montażu ma charakter przestrzenny. Typowym przypadkiem montażu przestrzennego jest

proces z wyróżnionymi frontami pracy. Fronty pracy są określone przez pozycje, jakie zajmuje monter i montowany obiekt względem siebie.

Wyróżnimy dwa modele montażu z frontami pracy. W pierwszym modelu monter przed wykonaniem operacji odwraca obiekt (jeśli jest to konieczne) do najbardziej dogodnej pozycji. W drugim modelu monter przed wykonaniem operacji przesuwa się względem obiektu w celu zajęcia najbardziej dogodnej pozycji dla wykonania operacji. Model ten jest zwięzany z gabarytami obiektu.

W pierwszym modelu wyróżnimy podstawowy front linii, określony przez ustawienie obiektu przed rozpoczęciem i po zakończeniu wszystkich operacji. Monter przed wykonaniem operacji ustawia obiekt na odpowiednim froncie pracy. Front pracy obiektu, który wchodzi na pewne stanowisko pracy, jest ostatnim frontem ustawionym na poprzednim stanowisku. Front pracy obiektu wchodzącego na stanowisko pracy nazwiemy podstawowym frontem stanowiska. Monter podejmuje wykonanie pewnej operacji, jeśli zdąży przed zakończeniem cyklu zmienić front pracy (jeśli to konieczne) i wykonać operację. Na ostatnim stanowisku, po zakończeniu wszystkich operacji, należy ustawić obiekt na podstawowym froncie linii montażowej.

W rozpatrywanym modelu BLM istotne znaczenie ma znajomość podstawowych frontów na każdym stanowisku pracy. Dla pierwszego stanowiska front taki jest dany jako podstawowy front linii. Dla pozostałych stanowisk odpowiednie fronty podstawowe wynikają z kolejności wykonywania operacji.

W drugim modelu monter zmienia swoje położenie względem montowanego obiektu, który znajduje się na jego stanowisku pracy. Monter przed wykonaniem operacji przechodzi na odpowiedni front pracy. Front pracy, na którym znajduje się monter przed rozpoczęciem montażu obiektu na swoim stanowisku pracy, nazwiemy podstawowym frontem stanowiska. Monter podejmuje wykonanie pewnej operacji, jeśli zdąży przed zakończeniem cyklu przejść na odpowiedni front pracy i wykonać operację, a następnie powrócić na front podstawowy.

W rozpatrywanym modelu BLM fronty podstawowe są określone przez pierwsze operacje wykonywane na stanowiskach pracy. A zatem wynikają one - tak jak w pierwszym modelu - z kolejności wykonywania operacji.

3.4.1. Model matematyczny

Uogólniony model BLM z frontami pracy zawiera wszystkie elementy klasycznego modelu BLM. Ponadto przyjmujemy, że w montowanym obiekcie wyróżniono J frontów pracy ($J < N$). Lokalizacja operacji we frontach pracy opisana jest wektorem:

$$F = [f_n] \qquad n = 1,\ldots,N \quad (3.32)$$

gdzie: f_n - numer frontu pracy operacji ω_n.

Niech czasy przejścia pomiędzy frontami pracy będą dane macierzą:

$$\tau = \left[\tau_{i,j}\right]_{\substack{i=1,\ldots,J \\ j=1,\ldots,J}} \tag{3.33}$$

gdzie: $\tau_{i,j}$ - czas przejścia od i-tego od j-tego frontu pracy. Macierz (3.33) może być niesymetryczna.

W modelu matematycznym BLM z frontami pracy modyfikacji ulegają (w stosunku do klasycznego modelu BLM) ograniczenia kolejnościowe, które przyjmują postać:

$$\bigwedge_{\omega} \bigwedge_{n} (\eta_{\omega,n} = 1) \Rightarrow (t_{\omega} < t_n - \nu_n^{\eta} - \tau_{i,j}) \tag{3.34}$$

przy tym:

$$i = f_{\omega} \tag{3.35}$$

$$j = f_n \tag{3.36}$$

3.4.2. Algorytm

Do rozwiązania problemu BLM z frontami pracy wykorzystamy algorytm programowania wieloetapowego (harmonogramowania). Stan i wartość stanu zdefiniujemy tak jak dla klasycznego modelu BLM.

G e n e r o w a n i e s t a n ó w

Oznaczmy przez j_n - front pracy operacji ω_n; j^1 (j^2) - front pracy pierwszej (ostatniej) operacji na ostatniej stacji w stanie $p^{\eta-1,\lambda}$, j^o - podstawowy front linii.

Procedura generowania stanów ma postać:

$$\bigwedge_{n} \bigwedge_{\omega} (p_n^{\eta-1,\lambda} = 0) \wedge \left[(\eta_{\omega,n} = 1) \Rightarrow (p_{\omega}^{\eta-1,\lambda} > 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{3.37}$$

Chwilę t_n wyznaczamy wg różnych formuł, zależnych od przyjętego modelu montażu z frontami pracy.

W modelu pierwszym (ruch obiektu względem montera) otrzymamy dla $\eta < N$:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \tau_{j^2,j_n} + \nu_n^{\eta}, \text{ jeśli } \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c - T^{\eta-1,\lambda} \geq \nu_n^{\eta} + \tau_{j^2,j_n} \\ \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c + \nu_n^{\eta} + \tau_{j^2,j_n}, \text{ w przypadku przeciwnym,} \end{cases} \tag{3.38}$$

natomiast dla $\eta = N$

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \nu_n^{\eta} + \tau_{j^2,j_n} + \tau_{j_n,j^o}, \text{ jeśli } \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c - T^{\eta-1,\lambda} \geq \nu_n^{\eta} + \\ \qquad + \tau_{j^2,j_n} + \tau_{j_n,j^o} \\ \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c + \tau_{j^2,j_n} + \nu_n^{\eta} + \tau_{j_n,j^o}, \text{ w przypadku przeciwnym,} \end{cases} \tag{3.39}$$

przy czym

$$j^2 = \begin{cases} f_{\mu}, \text{ jeśli } T^{\eta-1,\lambda} = p_{\mu}^{\eta-1,\lambda} > 0 \\ j^o, \text{ w przypadku przeciwnym.} \end{cases} \tag{3.40}$$

Dla drugiego modelu (ruch montera względem obiektu) otrzymamy:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \tau_{j^2,j_n} + \nu_n^{\eta}, \text{ jeśli } \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c - T^{\eta-1,\lambda} \geq \tau_{j^2,j_n} + \\ \qquad + \nu_n^{\eta} + \tau_{j_n,j^1} \\ \left[\frac{1}{c} T^{\eta-1,\lambda}\right]^{+} \cdot c + \nu_n^{\eta}, \text{ w przypadku przeciwnym,} \end{cases} \tag{3.41}$$

przy czym

$$j^1 = \begin{cases} f_{\omega}, \text{ jeśli } \min_{\omega_i \in \Omega_{K^{\eta-1,\lambda}}} p_i^{\eta-1,\lambda} = p_{\omega}^{\eta-1,\lambda} > 0 \\ j_n, \text{ w przypadku przeciwnym.} \end{cases} \tag{3.42}$$

oraz

$$\bigwedge_{i} \left\{\left[\frac{1}{c} p_i^{\eta-1,\lambda}\right]^{+} = K^{\eta-1,\lambda}\right\} \Rightarrow (\omega_i \in \Omega_{K^{\eta-1,\lambda}}) \tag{3.43}$$

Ponadto w stanie $P^{0,1}$ przyjmujemy $j^2 = j_n$.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Stany nieperspektywiczne eliminujemy za pomocą reguł sondowania oraz dominacji. Jako regułę sondowania można wykorzystać formułę przyjętą

dla klasycznego modelu BLM. Jednakże jest ona mało dokładna, ponieważ nie uwzględnia czasów przemieszczeń pomiędzy frontami pracy. Z kolei formuły uwzględniające przemieszczenia pomiędzy frontami pracy są obliczeniowo czasochłonne. Czas przejścia między frontami pracy można oszacować korzystając z dolnego ograniczenia dla problemu komiwojażera.

Dla pierwszego modelu BLM z frontami pracy dominacja stanów jest oparta na następującym twierdzeniu.

TWIERDZENIE 3.5

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeśli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge (f_{\mu_1}^{\eta,l_1} = f_{\mu_2}^{\eta,l_2}) \qquad (3.44)$$

gdzie: $f_{\mu}^{\eta,l}$ - front pracy ostatniej operacji wykonanej w stanie $P^{\eta,l}$.

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia o dominacji w klasycznym modelu BLM. Należy zaznaczyć, że ostatnie operacje wykonane w stanach P^{η,l_1} i P^{η,l_2} mogą być różne, lecz odpowiednie fronty muszą być jednakowe.

Dla drugiego modelu BLM z frontami pracy dominacja stanów jest oparta na następującym twierdzeniu.

TWIERDZENIE 3.6

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge (f_{\mu_1}^{\eta,l_1} = f_{\mu_2}^{\eta,l_2}) \wedge$$

$$\wedge (f_{\omega_1}^{\eta,l_1} = f_{\omega_2}^{\eta,l_2}) \qquad (3.45)$$

gdzie: $f_{\omega}^{\eta,l}$ - front pracy pierwszej operacji na ostatniej stacji w stanie $P^{\eta,l}$.

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia o dominacji w klasycznym modelu BLM.

3.5. KOSZT MONTAŻU

Uogólnienia klasycznego modelu BLM mogą dotyczyć ograniczeń lub kryterium optymalizacji. Obecnie rozważymy model, w którym zamiast minimalizacji nie wykorzystanego czasu pracy (liczby stanowisk) należy minimalizo-

wać koszt montażu. Model taki jest bardziej złożony, lecz w niektórych przypadkach dobrze opisuje sytuację występującą w praktyce przemysłowej. Istota problemu polega na tym, że dany jest koszt wykonania każdej operacji, a koszt wykonania podzbioru operacji jest kosztem najdroższej operacji. Płaca montera na stanowisku pracy jest zależna od kosztu najdroższej operacji, którą wykonuje. Zatem należy minimalizować sumę kosztów montażu na wszystkich stanowiskach pracy. W ogólnym przypadku kryterium takie nie jest równoważne minimalizacji liczby stanowisk pracy.

3.5.1. Model matematyczny

W uogólnionym modelu BLM założymy, że oprócz elementów modelu klasycznego dane są koszty wykonywania operacji:

$$W = [w_n]_{n=1,\ldots,N} \qquad (3.46)$$

gdzie: w_n - koszt wykonania operacji ω_n.

Jako kryterium optymalizacji w tym przypadku przyjmiemy minimalizację kosztu montażu, co zapiszemy w postaci:

$$G = \sum_{k=1}^{k=K} g_k \longrightarrow \min \qquad (3.47)$$

przy tym g_k określamy następująco:

$$g_k = \max_{\omega_n \in \Omega_k} w_n \qquad (3.48)$$

Warto zauważyć, że jedynie w przypadku jednakowych kosztów montażu podzbiorów operacji, kryterium (3.47) sprowadza się do minimalizacji liczby stanowisk pracy.

3.5.2. Algorytm

Problem minimalizacji kosztów montażu rozwiążemy za pomocą algorytmu programowania wieloetapowego (harmonogramowania). Przyjmiemy definicję stanu tak jak dla modelu klasycznego. Ponieważ wartość stanu powinna być związana z kryterium optymalizacji, zatem dla modelu uogólnionego wprowadzimy inną definicję wartości stanu.

Definicja 3.1

Wartość stanu jest wektorem:

$$v^{\eta,l} = [v_i^{\eta,l}]_{i=1,2} \qquad (3.49)$$

Elementy tego wektora określamy następująco:

$$v_1^{\eta,l} = \sum_{k=1}^{k=K^{\eta,l}-1} g_k^{\eta,l} \tag{3.50a}$$

$$v_2^{\eta,l} = g_{K^{\eta,l}} \tag{3.50b}$$

przy czym:

$$K^{\eta,l} = \left[\frac{1}{c}\, T^{\eta,l}\right]^+ \tag{3.51}$$

oraz:

$$g_k^{\eta,l} = \max_{\omega_n \in \Omega_k^{\eta,l}} w_n \tag{3.52}$$

Ponadto do zbioru $\Omega_k^{\eta,l}$ należą tylko te operacje ω_n, które spełniają warunek:

$$\forall_n \left(\left[\frac{1}{c}\, p_n^{\eta,l}\right]^+ = k \right) \Rightarrow (\omega_n \in \Omega_k^{\eta,l}) \tag{3.53}$$

Wartość stanu $P^{0,1}$ jest wektorem zerowym, natomiast dla każdego stanu końcowego $P^{N,l}$ można wyznaczyć koszty montażu $G^{N,l}$ jako:

$$G^{N,l} = v_1^{N,l} + v_2^{N,l} \tag{3.54}$$

Stan końcowy o najmniejszych kosztach jest stanem optymalnym. Z optymalnego stanu wyznaczamy, zgodnie z (3.53), optymalne podzbiory operacji.

G e n e r o w a n i e s t a n ó w

W rozważanym uogólnionym modelu BLM generowanie stanów jest bardziej skomplikowane niż w modelu klasycznym. Istota problemu polega na tym, by wygenerować wszystkie rozwiązania dopuszczalne, wśród których znajduje się rozwiązanie optymalne. W modelu klasycznym operacja ω_n była przydzielona na jak najwcześniejsze stanowisko pracy. Postulat ten wynikał z przyjętego kryterium minimalizacji liczby stanowisk pracy. W modelu uogólnionym należy minimalizować koszt montażu. Przydzielając operację ω_n w stanie $P^{\eta-1,\lambda}$ do jak najwcześniejszego stanowiska pracy można nie uzyskać rozwiązania optymalnego. Istotne znaczenie ma koszt w_n przydzielonej

operacji ω_n, w odniesieniu do $v_2^{\eta-1,\lambda}$. W związku z tym wyróżnimy cztery procedury generowania stanów. Dowód poprawności konstrukcji procedur generowania stanów zostanie przedstawiony w następnym punkcie.

Jeżeli:

$$v_2^{\eta-1,\lambda} \geq w_n \tag{3.55}$$

oraz

$$\left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c - T^{\eta-1,\lambda} \geq \vartheta_n^{\eta} \tag{3.56}$$

to operacja ω_n może być przydzielona na $K^{\eta-1,\lambda}$-te stanowisko pracy (stację montażową) bez podwyższania kosztu montażu na tym stanowisku.

W przypadku gdy warunek (3.55) nie jest spełniony i zachodzi nierówność (3.56) dla $\eta > 1$, wówczas operacja ω_n musi być przydzielona do $K^{\eta-1,\lambda}$-go i $K^{\eta-1,\lambda}+1$-go stanowiska pracy. Uwzględnienie tych dwóch wariantów pozwala wygenerować wszystkie rozwiązania dopuszczalne, a zatem również rozwiązanie optymalne.

Jedynie w stanie $P^{0,1}$ operacje są przydzielone tylko do pierwszego stanowiska pracy. Wprawdzie każda operacja ω_n nie spełnia wówczas warunku (3.55) i spełnia (3.56), lecz jej przydzielanie do drugiego stanowiska pracy nie miałoby sensu.

W przypadku gdy nierówność (3.56) nie jest spełniona, operacja ω_n jest przydzielana tylko do $K^{\eta-1,\lambda}+1$-go stanowiska pracy.

Tak więc stan P jest wygenerowany na podstawie jednej z czterech procedur. Procedura startowa (dla $P^{0,1}$) jest taka sama jak dla modelu klasycznego. Jeżeli warunek (3.55) jest spełniony, to procedura generowania stanów ma postać:

$$\forall_n \forall_\vartheta (w_n \leq v_2^{\eta-1,\lambda}) \wedge (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\vartheta,n}=1) \Rightarrow (p_\vartheta^{\eta-1,\lambda} > 0)\right] \Rightarrow$$

$$\Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{3.57}$$

Elementy wektora ΔP są określone tak jak w (3.23a).
Chwilę t_n wyznaczamy z formuły:

$$t_n = \begin{cases} T^{\eta-1,\lambda} + \vartheta_n^{\eta}, \text{ jeśli } \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c}(T^{\eta-1,\lambda} + \vartheta_n^{\eta})\right]^+ \\ \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c + \vartheta_n^{\eta}, \text{ w przypadku przeciwnym.} \end{cases} \tag{3.58}$$

Równocześnie z generowaniem stanu P wyznaczamy jego wartość V, za pomocą wzorów:

$$v_1 = \begin{cases} v_1^{\eta-1,\lambda}, \text{ jeśli } \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c}(T^{\eta-1,\lambda} + \varphi_n^{\eta})\right]^+ \\ v_1^{\eta-1,\lambda} + v_2^{\eta-1,\lambda}, \text{ w przypadku przeciwnym} \end{cases} \tag{3.59}$$

oraz

$$v_2 = \begin{cases} v_2^{\eta-1,\lambda}, \text{ jeśli } \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ = \left[\frac{1}{c}(T^{\eta-1,\lambda} + \varphi_n^{\eta})\right]^+ \\ w_n, \text{ w przypadku przeciwnym.} \end{cases} \tag{3.60}$$

W przypadku gdy warunki (3.55) i (3.56) nie są spełnione, to procedura generowania stanów ma postać:

$$\bigwedge_{\varphi}\bigvee_{n}(v_2^{\eta-1,\lambda} < w_n) \wedge \left(\left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c - T^{\eta-1,\lambda} < \varphi_n^{\eta}\right) \wedge (p_n^{\eta-1,\lambda} = 0) \wedge$$

$$\wedge\left[(\varphi_{\varphi,n}=1) \Rightarrow (p_{\varphi}^{\eta-1,\lambda} > 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{3.61}$$

Elementy wektora ΔP są zdefiniowane tak jak w (3.23a), przy tym:

$$t_n = \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c + \varphi_n^{\eta} \tag{3.62}$$

Elementy wektora wartości V otrzymujemy z zależności:

$$v_1 = v_1^{\eta-1,\lambda} + v_2^{\eta-1,\lambda} \tag{3.63}$$

oraz:

$$v_2 = w_n \tag{3.64}$$

W przypadku gdy warunek (3.55) nie jest spełniony, a warunek (3.56) jest spełniony, procedura generowania stanów ma postać:

$$\bigvee_{n}\bigwedge_{\varphi}(v_2^{\eta-1,\lambda} < w_n) \wedge \left(\left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c - T^{\eta-1,\lambda} \geq \varphi_n^{\eta}\right) \wedge (p_n^{\eta-1,\lambda}=0) \wedge \tag{3.65}$$

$$\wedge\left[(\varphi_{\varphi,n}=1) \Rightarrow (p_{\varphi}^{\eta-1,\lambda} > 0)\right] \Rightarrow (P^1 = P^{\eta-1,\lambda} + \Delta P^1) \wedge (P^2 = P^{\eta-1,\lambda} + \Delta P^2)$$

Elementy wektorów ΔP^1 i ΔP^2 są określone tak jak w (3.23a), przy czym:

$$t_n^1 = T^{\eta-1,\lambda} + \varphi_n^{\eta} \tag{3.66a}$$

$$t_n^2 = \left[\frac{1}{c}\, T^{\eta-1,\lambda}\right]^+ \cdot c + \varphi_n^{\eta} \tag{3.66b}$$

Ponadto:

$$v_1^1 = v_1^{\eta-1,\lambda} \tag{3.67a}$$

$$v_1^2 = v_1^{\eta-1,\lambda} + v_2^{\eta-1,\lambda} \tag{3.67b}$$

oraz

$$v_2^1 = w_n \tag{3.68a}$$

$$v_2^2 = w_n \tag{3.68b}$$

A zatem za pomocą jednej z przedstawionych procedur, ze stanu $P^{\eta-1,\lambda}$ generowane są kolejne stany.

3.5.3. Eliminowanie stanów nieperspektywicznych

Dla zwiększenia efektywności algorytmu, w trakcie obliczeń eliminowane są stany nieperspektywiczne. Efektywność algorytmu zależy od liczby stanów wygenerowanych oraz liczby stanów wyeliminowanych - spośród wygenerowanych. Dla uzyskania rozwiązania optymalnego muszę być generowane tylko stany perspektywiczne. Z tego też względu w rozważanym problemie uwzględniono cztery procedury generowania stanów. W modelu klasycznym nie przydziela się operacji do dwóch kolejnych stanowisk pracy (ponieważ jeden z wygenerowanych stanów byłby zdominowany przez drugi). Podobna sytuacja występuje w rozważanym modelu uogólnionym dla operacji ω_n, która spełnia warunek (3.55). Wynika to z następującego twierdzenia.

TWIERDZENIE 3.7

Jeżeli operacja ω_n spełnia warunki (3.55) i (3.56) dla $\eta > 1$, to przydzielając ją do $K^{\eta-1,\lambda}$+1-go stanowiska pracy otrzymujemy stan nieperspektywiczny.

D o w ó d: Załóżmy dla dowodu, że ze stanu $P^{\eta-1,\lambda}$ generowane są dwa stany P^1 i P^2. Stany te otrzymujemy przez przydzielenie operacji ω_n spełniającej warunki (3.55) i (3.56), odpowiednio do $K^{\eta-1,\lambda}$-go oraz

$K^{\eta-1,\lambda}{}_{+1}$-go stanowiska pracy. Przyjmijmy dla uproszczenia zapisu $\mu = K^{\eta-1,\lambda}$.

Załóżmy, że określona jest trajektoria lokalnie optymalna $P^2,\ldots,P^{N,l_2}$. Trajektorii tej odpowiadają kolejne podzbiory operacji $\Omega^2_{\mu+1},\ldots,\Omega^2_K$, przy czym

$$\omega_n \in \Omega^2_{\mu+1} \tag{3.69}$$

Koszt montażu w tym przypadku wyniesie:

$$G^2 = v_1^{\eta-1,\lambda} + v_2^{\eta-1,\lambda} + \sum_{k=\mu+1}^{k=K} g_k^2 \tag{3.70}$$

Zauważmy, że od stanu P^1 można zrealizować trajektorię (niekoniecznie optymalną), której odpowiadają kolejne podzbiory:

$$\Omega^1_{\mu+1},\ldots,\Omega^1_K$$

przy czym:

$$\Omega^1_{\mu+1} = \Omega^2_{\mu+1} \setminus \omega_n \tag{3.70a}$$

oraz

$$\Omega^1_k = \Omega^2_k \qquad k = \mu+2,\ldots,K \tag{3.70b}$$

Koszt montażu w tym przypadku wyniesie:

$$G^1 = v_1^{\eta-1,\lambda} + \max(v_2^{\eta-1,\lambda}, w_n) + \sum_{k=\mu+1}^{k=K} g_k^1 \tag{3.71}$$

Wyznaczmy z kolei różnicę kosztów:

$$\Delta G = G^2 - G^1 \tag{3.72}$$

Podstawiając (3.70) i (3.71) do (3.72) otrzymamy:

$$\Delta G = v_2^{\eta-1,\lambda} + \sum_{k=\mu+1}^{k=K} g_k^2 - \max(v_2^{\eta-1,\lambda}, w_n) - \sum_{k=\mu+1}^{k=K} g_k^1 \tag{3.73}$$

Uwzględniając (3.70a) i (3.70b) w (3.73) dochodzimy do wzoru:

$$\Delta G = v_2^{\eta-1,\lambda} + g^2_{\mu+1} - \max(v_2^{\eta-1,\lambda}, w_n) - g^1_{\mu+1} \tag{3.74}$$

Ponieważ zgodnie z (3.70a):

$$g^1_{\mu+1} \le g^2_{\mu+1} \tag{3.75}$$

zatem:

$$\Delta G \ge v_2^{\eta-1,\lambda} - \max(v_2^{\eta-1,\lambda}, w_n) \tag{3.76}$$

Uwzględniając (3.55) w (3.76) otrzymamy ostatecznie:

$$\Delta G \ge 0 \tag{3.77}$$

co kończy dowód, bowiem koszt montażu dla trajektorii $P^{0,1},\ldots,P^{\eta-1,\lambda}, P^1,\ldots,P^{N,l_1}$ nie jest większy niż koszt montażu dla trajektorii $P^{0,1},\ldots,P^{\eta-1,\lambda}, P^2,\ldots,P^{N,l_2}$. □

WNIOSEK 3

Jeżeli operacja ω_n nie spełnia warunku (3.55) i spełnia warunek (3.56) dla $\eta > 1$, to trzeba ją przydzielać do $K^{\eta-1,\lambda}$-go i $K^{\eta-1,\lambda}{}_{+1}$-go stanowiska pracy.

Słuszność tego wniosku wynika wprost z (3.76) uwzględniając nie spełniony warunek (3.55), bowiem:

$$\Delta G \ge v_2^{\eta-1,\lambda} - w_n \tag{3.78}$$

Tak więc ΔG może być dodatnie lub ujemne. □

Spośród wygenerowanych stanów, niektóre mogą okazać się nieperspektywiczne. Do eliminacji tych stanów można wykorzystać reguły dominacji, wyczerpywania oraz sondowania.

Reguła dominacji ma postać twierdzenia.

TWIERDZENIE 3.8

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (K^{\eta,l_1} \cdot c\text{-}T^{\eta,l_1} \ge K^{\eta,l_2} \cdot c\text{-}T^{\eta,l_2}) \wedge$$

$$\wedge (v_1^{\eta,l_1} \leq v_1^{\eta,l_2}) \wedge (v_2^{\eta,l_1} \leq v_2^{\eta,l_2}) \qquad (3.79)$$

D o w ó d: Załóżmy, że liczby stanowisk pracy K^{η,l_1} i K^{η,l_2} mogą być różne. Jeżeli spełnione są pierwsze trzy czynniki warunku (3.79), to każda trajektoria wychodząca ze stanu P^{η,l_2} daje rozwiązanie, które można uzyskać również od stanu P^{η,l_1}.

Załóżmy, że określona jest trajektoria lokalnie optymalna $P^{\eta,l_2},\ldots,P^{N,l_2^o}$, której odpowiadają podzbiory operacji:

$$\Omega_0^2,\ldots,\Omega_M^2$$

przy czym podzbiór Ω_0^2 odpowiada K^{η,l_2}-tej stacji, a kolejne podzbiory przypadają na kolejne stacje $(K^{\eta,l_2} + M \leq K)$. A zatem koszt montażu dla stanu P^{N,l_2^o} można przedstawić w postaci:

$$G^2 = v_1^{\eta,l_2} + \sum_{m=0}^{m=M} g_m^2 \qquad (3.80)$$

Oznaczmy przez $P^{N,2}$ końcowy stan trajektorii wychodzącej ze stanu P^{η,l_1}, z której otrzymamy podzbiory operacji:

$$\Omega_0^1,\ldots,\Omega_M^1$$

przy czym podzbiór Ω_0^1 odpowiada K^{η,l_1}-tej stacji, a kolejne podzbiory przypadają na kolejne stacje $(K^{\eta,l_1} + M \leq K)$.
A zatem koszt montażu dla stanu $P^{N,2}$ można napisać w postaci:

$$G^1 = v_1^{\eta,l_1} + \sum_{a=0}^{m=M} g_a^1 \qquad (3.81)$$

Załóżmy, że:

$$\Omega_0^1 \neq \Omega_0^2 \qquad (3.82a)$$

oraz

$$\Omega_a^1 = \Omega_a^2 \qquad a = 1,\ldots,M \qquad (3.82b)$$

Porównując (3.80) z (3.81) widzimy, że:

$$G^2 - G^1 = v_1^{\eta,l_2} - v_1^{\eta,l_1} + g_0^2 - g_0^1 \qquad (3.83)$$

Ponieważ $v_2^{\eta,l_1} \leq v_2^{\eta,l_2}$, stąd wnioskujemy, że $g_0^1 \leq g_0^2$. Poza tym $v_1^{\eta,l_1} \leq v_1^{\eta,l_1}$, a zatem z (3.83) wynika, że:

$$G^2 \geq G^1 \qquad (3.84)$$

co kończy dowód. □

W rozważanym problemie zakłada się, że linia montażowa ma K stacji (K < N). Warto zauważyć, że gdyby przydzielane w stanach $P^{\eta-1,2}$ operacje ω_n były uporządkowane wg rosnących kosztów w_n, to dla pewnej trajektorii już w etapie $\eta = K$ można osięgnąć K-tę stację.
Z tego względu istotne znaczenie ma reguła wyczerpywania oparta na twierdzeniu

TWIERDZENIE 3.9

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$b^{\eta,l} > K \qquad (3.85)$$

D o w ó d tego twierdzenia jest analogiczny jak dla reguły sondowania w klasycznym modelu BLM.

Reguła sondowania dla rozważanego uogólnionego modelu BLM ma postać twierdzenia.

TWIERDZENIE 3.10

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$b_G^{\eta,l} > v_1^a + v_2^a \qquad (3.86)$$

gdzie: $b_G^{\eta,l}$ - dolne ograniczenie kosztu montażu stanu lokalnie optymalnego otrzymanego ze stanu $P^{\eta,l}$.

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla reguły sondowania w klasycznym modelu BLM. W twierdzeniu tym istotne znaczenie ma oszacowanie $b_G^{\eta,l}$.

W najprostszym przypadku można przyjąć:

$$b_G^{\eta,l} = v_1 + \max(v_2, \max_{\omega_n \notin \Omega^{\eta,l}} w_n) \tag{3.86a}$$

W (3.86a) przyjęto, że wszystkie operacje $\omega_n \notin \Omega^{\eta,l}$ zostaną przydzielone do ostatniego (μ-tego) stanowiska pracy w stanie $P^{\eta,l}$.

Jeżeli:

$$\mu < b^{\eta,l} \tag{3.86b}$$

to oszacowanie $b_G^{\eta,l}$ można wyznaczyć na podstawie następującego algorytmu.

Operacje $\omega_n \in \Omega'$, gdzie:

$$\bigvee_n [(\omega_n \notin \Omega^{\eta,l}) \vee (\omega_n \in \Omega_{[\mu]})] \Rightarrow (\omega_n \in \Omega') \tag{3.86c}$$

powinny być przydzielone do podzbiorów $\Omega^{\mu},\ldots,\Omega^{m},\ldots,\Omega^{M}$, gdzie $M = b^{\eta,l}$. Załóżmy, że operacje $\omega_n \in \Omega'$ są uporządkowane wg malejącego kosztu. Wówczas można utworzyć jednoelementowe podzbiory: $\Omega^{M},\ldots\Omega^{m},\ldots,\Omega^{\mu+1}$ oraz podzbiór Ω^{μ} zawierający pozostałe operacje ze zbioru Ω'. Przyjmujemy, że do Ω^{M} przydzielona jest operacja o najniższym koszcie itd. Tak więc:

$$g^{m-1} > g^{m} \qquad m = \mu+1,\ldots,M \tag{3.86d}$$

gdzie: g^{m} - koszt podzbioru Ω^{m}.

A zatem $b_G^{\eta,l}$ można oszacować jako:

$$b_G^{\eta,l} = v_1 + \sum_{m=\mu}^{m=M} g^{m} \tag{3.86e}$$

Przesunięcie operacji z Ω^{m-1} do Ω^{m} (z uwagi na ograniczenia czasu lub kolejności) może jedynie zwiększyć $b_G^{\eta,l}$.

4. ALOKACJA OPERACJI. SZEREGOWE STANOWISKA PRACY

4.1. WSTĘP

W modelach BLM stanowisko pracy utożsamiano ze stacją montażową. Rezultatem BLM były podzbiory operacji tworzące stanowiska pracy, lokowane do kolejnych stacji montażowych. A zatem lokalizacja operacji do stacji montażowych nie stanowiła istotnego problemu.

W tym rozdziale rozważymy problemy alokacji operacji (AO) do szeregowych stanowisk pracy. Przyjmujemy zatem podstawowe założenie, że operacje montażu obiektu są wykonywane sekwencyjnie. Tym samym na stacji montażowej może znajdować się co najwyżej jedno stanowisko pracy.

W pierwszym modelu AO wyróżnimy pozycje, na których w ramach stacji montażowej może znajdować się stanowisko pracy. W typowym przypadku stanowisko pracy może znajdować się wyłącznie po lewej lub prawej stronie osi linii. Wyróżnienie pozycji jest związane z konstrukcją linii lub gabarytem obiektu. Problem AO pojawia się, gdy montaż przestrzenny nie jest dopuszczalny. Wówczas podzbiór operacji tworzący stanowisko pracy musi być możliwy do wykonania z jednej pozycji. Ponadto czasy wykonywania operacji z różnych pozycji mogą być różne.

Istotny problem stanowi zatem wyznaczenie takich pozycji wykonania operacji w linii, by liczba stanowisk pracy była minimalna.

W drugim z rozważanych modeli AO uwzględnimy ograniczenia czasowe wykonywania operacji, które wynikają z technologii montażu (np. klejenie lub malowanie). W takich przypadkach można określić najkrótszy i najdłuższy czas, jaki musi (może) upłynąć pomiędzy zakończeniem pewnej operacji i rozpoczęciem innej operacji. Z tego względu przydzielanie operacji do kolejnych stacji montażowych nie zawsze jest dopuszczalne. Problem AO polega w tym modelu na wyznaczeniu stacji dla wykonania operacji tak, by liczba stanowisk pracy była minimalna.

W trzecim modelu AO, który stanowi przedmiot rozważań w tym rozdziale, wyróżnimy zakresy wykonania operacji. Zakres operacji określa stacje montażowe, na których można wykonać operację. Ograniczenie to wynika z oprzyrządowania linii montażowej oraz lokalizacji detali. Czas wykonania operacji może być zależny od stacji, na której jest ona wykonywana. Zatem problem AO polega na wyznaczeniu - dla każdej operacji - stacji montażowej tak, by liczba stanowisk pracy była minimalna.

Na zakończenie tego rozdziału rozważymy linię asymetryczną, w której stanowisko pracy może być zlokalizowane po lewej lub po prawej stronie osi

linii. Przy założeniu szeregowych stanowisk pracy problem AO w linii asymetrycznej jest szczególnym przypadkiem modelu z ograniczeniami pozycyjnymi. Jednakże sposób rozwiązania tego przypadku stanowi dobre wprowadzenie do problemów AO w liniach z równoległymi stanowiskami pracy.

4.2. OGRANICZENIA POZYCYJNE

Rozważymy proces montażu, w którym na każdej stacji operacje mogę być wykonywane przez jednego montera zajmującego jedną z wyróżnionych pozycji. Ponadto dla każdej operacji określone są pozycje, z których operacja może być wykonana. Istota ograniczeń pozycyjnych polega na tym, że podzbiór operacji tworzący stanowisko pracy musi posiadać wspólną pozycję, z której mogę być wykonane wszystkie operacje.

4.2.1. Model matematyczny

Załóżmy, że dana jest linia montażowa składająca się z K stacji. Na każdej stacji może znajdować się co najwyżej jedno stanowisko pracy. Dla określenia lokalizacji stanowiska pracy na stacji montażowej wyróżnia się J pozycji. Kompletny montaż obiektu wymaga wykonania wszystkich operacji ω_n $(n = 1,\dots,N)$ ze zbioru Ω. Czasy operacji ϑ_n dane są wektorem Θ, natomiast relację kolejności określa macierz Γ. Dany jest cykl c procesu montażu.

Alternatywne przyporządkowanie operacji do pozycji stanowiska pracy dane jest macierzą:

$$A = \left[a_{n,j}\right]_{\substack{n=1,\dots,N \\ j=1,\dots,J}} \tag{4.1}$$

Elementy tej macierzy określamy następująco:

$$a_{n,j} = \begin{cases} 1, \text{ jeśli operację } \omega_n \text{ można wykonać z j-tej pozycji} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.1a}$$

Rezultatem alokacji operacji są dopuszczalne podzbiory operacji Ω_k, $k = 1,\dots,K$. Podzbiór Ω_k jest dopuszczalny, jeżeli spełnia następujące ograniczenia:

- sum czasów

$$\bigwedge_{1\le k\le K} \sum_{\omega_n \in \Omega_k} \vartheta_n \le c \tag{4.2a}$$

- kolejnościowe

$$\bigwedge_{\nu} \bigwedge_{n} \bigvee_{\mu\le k} (\gamma_{\nu,n}=1) \wedge (\omega_n \in \Omega_k) \Rightarrow (\omega_\nu \in \Omega_\mu) \tag{4.2b}$$

- pozycyjne

$$\bigwedge_{1\le k\le K} \bigvee_{j_k} \bigwedge_{n} (\omega_n \in \Omega_k) \Rightarrow (a_{n,j_k} = 1) \tag{4.2c}$$

Ograniczenie (4.2c) interpretujemy jako warunek możliwości wykonania z j_k-tej pozycji każdej operacji przydzielonej do k-tej stacji.

Jako kryterium optymalnej alokacji operacji przyjmujemy minimalizację liczby stanowisk pracy. Funkcję celu można zapisać w postaci:

$$Q = \sum_{k=1}^{k=K} q_k \longrightarrow \min \tag{4.3}$$

przy czym:

$$q_k = \begin{cases} 1, \text{ jeśli } \Omega_k \neq 0 \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.3a}$$

Wprowadzając zmienną t_n (dla oznaczenia chwili zakończenia operacji ω_n), otrzymamy model matematyczny alokacji operacji w postaci:

- ograniczenia

$$\bigwedge_{1\le n\le N} \left[\tfrac{1}{c}(t_n-\vartheta_n)\right]^+ = \left[\tfrac{1}{c}\, t_n\right]^+ \tag{4.4a}$$

$$\bigwedge_{\nu} \bigwedge_{n} (\gamma_{\nu,n} = 1) \Rightarrow (t_\nu \le t_n-\vartheta_n) \tag{4.4b}$$

$$\bigwedge_{1\le k\le K} \bigvee_{j_k} \bigwedge_{n} \left\{\left[\tfrac{1}{c}\, t_n\right]^+ = k\right\} \Rightarrow (a_{n,j_k} = 1) \tag{4.4c}$$

- kryterium optymalizacji określa (4.3), przy tym:

$$q_k = \begin{cases} 1, \text{ jeśli } \bigvee_{n} \left[\tfrac{1}{c}\, t_n\right]^+ = k \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.5}$$

4.2.2. Algorytm

Do rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. Stan $P^{\eta,l}$ jest wektorem o współrzędnych $p_n^{\eta,l} = t_n$, jeżeli operacja ω_n została przydzielona do stanowiska pracy ($p_n^{\eta,l} = 0$, w przypadku przeciwnym).

Ze stanu końcowego $P^{N,l}$ można wprost otrzymać dopuszczalne rozwiązanie problemu, bowiem:

$$k_n = \left[\frac{1}{c} p_n^{\eta,l}\right]^+ \tag{4.6}$$

gdzie: k_n - numer stacji, do której przydzielono operację ω_n.

Ponadto pozycję j_k, na której zlokalizowane jest stanowisko pracy k-tej stacji, wyznaczamy z warunku:

$$\exists_{1 \leq j_k \leq J} \; \forall_n \left\{\left[\frac{1}{c} p_n^{\eta,l}\right]^+ = k\right\} \Rightarrow (a_{n,j_k} = 1) \tag{4.7}$$

Chwilę $T^{\eta,l}$ zakończenia montażu wszystkich operacji należących do stanu $P^{\eta,l}$ wyznaczamy ze wzoru:

$$T^{\eta,l} = \max_{1 \leq i \leq N} p_i^{\eta,l} \tag{4.8}$$

A zatem numer stacji $K^{\eta,l}$, na której wykonano ostatnią operację w stanie $P^{\eta,l}$, wyznaczamy jako:

$$K^{\eta,l} = \left[\frac{1}{c} T^{\eta,l}\right]^+ \tag{4.9}$$

Wartość stanu zdefiniujemy następująco:

Definicja 4.1

Wartość stanu jest skalarem:

$$V^{\eta,l} = \sum_{k=1}^{k=K^{\eta,l}} q_k^{\eta,l} \tag{4.10}$$

przy czym:

$$q_k^{\eta,l} = \begin{cases} 1, \text{ jeśli } \exists_n \left[\frac{1}{c} p_n^{\eta,l}\right]^+ = k \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.11}$$

Jeżeli na wszystkich stacjach od 1 do $K^{\eta,l}$ znajdują się stanowiska pracy, to:

$$V^{\eta,l} = K^{\eta,l} \tag{4.12}$$

Dalej udowodnimy, że dla trajektorii perspektywicznych warunek ten musi być spełniony.

Optymalny stan końcowy P^o wyznaczamy z warunku:

$$\left(\min_{1 \leq l \leq L_N} V^{N,l} = V^{N,l^o}\right) \Rightarrow (P^{N,l^o} = P^o) \tag{4.13}$$

G e n e r o w a n i e s t a n ó w

Rozważmy wyznaczanie bezpośrednich następników P wybranego stanu $P^{\eta-1,\lambda}$. W tym celu respektując ograniczenia (4.4) należy przydzielić operację ω_n do stanu $P^{\eta-1,\lambda}$.

Oznaczmy przez $A^{\eta-1,\lambda}$ wektor (o wymiarze J) pozycji, na których mogą być wykonywane operacje przydzielone do $K^{\eta-1,\lambda}$-tej stacji w stanie $P^{\eta-1,\lambda}$. Elementy tego wektora określamy następująco:

$$a_j^{\eta-1,\lambda} = \begin{cases} 1, \text{ jeśli } \forall_i \left\{\left[\frac{1}{c} p_i^{\eta-1,\lambda}\right]^+ = K^{\eta-1,\lambda}\right\} \Rightarrow (a_{i,j} = 1) \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.14}$$

Ponadto przyjmiemy, że dla stanu $P^{0,1}$ wektor $A^{0,1}$ ma wszystkie współrzędne równe jedności.

Operacja ω_n, która ma być przydzielona na stanowisko pracy $K^{\eta-1,\lambda}$-tej stacji, musi spełniać warunek:

$$\exists_j \; (a_j^{\eta-1,\lambda} = 1) \wedge (a_{n,j} = 1) \wedge (K^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} \geq \vartheta_n) \tag{4.15}$$

W przypadku przeciwnym operację tę przydzielamy do następnej stacji.

Ogólna procedura generowania stanów ma postać:

$$\forall_n \forall_\nu \; (p_n^{\eta-1,\lambda} = 0) \wedge \left[(\gamma_{\nu,n} = 1) \Rightarrow (p_\nu^{\eta-1,\lambda} > 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{4.16}$$

Elementy wektora ΔP mają postać:

$$\Delta p_i = \begin{cases} t_n, \text{ jeśli } i=n \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.17}$$

Chwilę t_n wyznaczamy ze wzoru:

$$t_n = \begin{cases} T^{\nu-1,\lambda} + v_n^{\nu}, \text{ jeśli } \exists_j (a_j^{\nu-1,\lambda}=1) \wedge (a_{n,j}=1) \wedge (K^{\nu-1,\lambda} \cdot c + \\ \qquad - T^{\nu-1,\lambda} > v_n^{\nu}) \\ K^{\nu-1,\lambda} \cdot c + v_n^{\nu}, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.18}$$

Dla wygenerowanego stanu $P^{\nu,l}$ można rekurencyjnie otrzymać:

- Jeżeli spełniony jest warunek:

$$t_n = T^{\nu-1,\lambda} + v_n^{\nu} \tag{4.19}$$

to współrzędne wektora $A^{\nu,l}$ wyznaczamy z formuły

$$a_j^{\nu,l} = \begin{cases} 1, \text{ jeśli } (a_j^{\nu-1,\lambda}=1) \wedge (a_{n,j}=1) \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.20}$$

- Jeżeli warunek (4.19) nie jest spełniony, tzn. gdy:

$$t_n = K^{\nu-1,\lambda} \cdot c + v_n^{\nu} \tag{4.21}$$

to:

$$a_j^{\nu,l} = a_{n,j} \qquad j = 1,\ldots,J \tag{4.22}$$

Ponadto bezpośrednio otrzymujemy:

$$T^{\nu,l} = t_n \tag{4.23}$$

oraz

$$V^{\nu,l} = \left[\frac{1}{c} t_n\right]^+ \tag{4.24}$$

W procedurze generowania stanów przydzielamy operację ω_n do jak najwcześniejszej stacji montażowej (co wynika z (4.18)). Przydzielenie operacji ω_n do dalszych stacji daje stan nieperspektywiczny. Tezę tę udowodnimy w oparciu o regułę dominacji stanów.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Do eliminacji stanów wykorzystamy reguły wyczerpywania, sondowania oraz dominacji. Reguła sondowania ma postać twierdzenia.

TWIERDZENIE 4.1

Stan $P^{\nu,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$b^{\nu,l} \geq V^{a} \tag{4.25}$$

przy czym:

$$b^{\nu,l} = \left[\frac{1}{c}(T^{\nu,l} + \sum_{\omega_i \notin \Omega^{\nu,l}} v_i^{\nu})\right]^+ \tag{4.26}$$

oraz

$$\bigwedge_i (p_i^{\nu,l}=0) \Rightarrow (\omega_i \notin \Omega^{\nu,l}) \tag{4.27}$$

D o w ó d: Stan lokalnie optymalny P^{N,l^0} spełnia warunek:

$$b^{\nu,l} \leq V^{N,l^0} \tag{4.28}$$

Tym samym z (4.25) i (4.28) wynika, że:

$$V^{N,l^0} \geq V^{a} \tag{4.29}$$

Czyli stan $P^{\nu,l}$ jest nieperspektywiczny, bo nie można uzyskać z niego lepszego stanu końcowego niż aktualnie najlepszy stan końcowy P^a. □

Jeżeli stan P^a nie został jeszcze wygenerowany, to przyjmując w (4.25) $V^a = K$, otrzymujemy regułę wyczerpywania stanów.

Reguła dominacji stanów ma postać twierdzenia.

TWIERDZENIE 4.2

Stan P^{ν,l_1} dominuje nad stanem P^{ν,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{1 \leq j \leq J} (\Omega^{\nu,l_1} = \Omega^{\nu,l_2}) \wedge \left\{(V^{\nu,l_1} < V^{\nu,l_2}) \vee \left\{(V^{\nu,l_1} = V^{\nu,l_2}) \Rightarrow \right.\right.$$
$$\left.\left. \Rightarrow \left[(T^{\nu,l_1} \leq T^{\nu,l_2}) \wedge (a_j^{\nu,l_1} \geq a_j^{\nu,l_2})\right]\right\}\right\} \tag{4.30}$$

D o w ó d: Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\eta,l_2}, \ldots, P^{N,l_2^o}$. Trajektoria ta reprezentuje określony przydział operacji do stanowisk pracy. Jeżeli jest spełniony warunek (4.30), to przydział ten jest również realizowalny od stanu P^{η,l_1}. Ponieważ $T^{\eta,l_1} \leq T^{\eta,l_2}$, zatem otrzymamy trajektorię $P^{\eta,l_1}, \ldots, P^{N,\lambda}$. Stan $P^{N,\lambda}$ na ogół nie jest stanem lokalnie optymalnym P^{N,l_1^o}, zatem możemy napisać:

$$V^{N,l_1^o} \leq V^{N,\lambda} \tag{4.31}$$

Ponadto:

$$V^{N,\lambda} \leq V^{N,l_2^o} \tag{4.32}$$

Na podstawie (4.31) i (4.32) wnioskujemy, że:

$$V^{N,l_1^o} \leq V^{N,l_2^o} \tag{4.33}$$

e zatem stan P^{η,l_2} jest nieperspektywiczny. □

Załóżmy, że dany jest stan $P^{\eta-1,\lambda}$, do którego przydzielamy operację ω_n. Niech będzie spełniony warunek (4.15). A zatem operację ω_n można przydzielić do stanowiska pracy $K^{\eta-1,\lambda}$-tej stacji.

LEMAT 4.1

Jeżeli dla stanu $P^{\eta-1,\lambda}$ i operacji ω_n spełniony jest warunek (4.15), to przydzielenie tej operacji do stacji k, gdzie:

$$K^{\eta-1,\lambda} < k \tag{4.34}$$

daje stan nieperspektywiczny.

D o w ó d: Załóżmy, że stan P^{η,l_1} otrzymujemy ze stanu $P^{\eta-1,\lambda}$ przydzielając operację ω_n do stacji $K^{\eta-1,\lambda}$. Z kolei przez P^{η,l_2} oznaczymy stan otrzymany z $P^{\eta-1,\lambda}$ przez przydzielenie operacji ω_n do stacji k. Można zauważyć, że stany P^{η,l_1} i P^{η,l_2} spełniają warunek (4.30), a stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, czyli stan P^{η,l_2} jest nieperspektywiczny. □

Zastosowana w algorytmie procedura generowania stanów spełnia zatem warunek konieczny perspektywiczności generowanych stanów (trajektorii).

Jednakże nie jest to warunek wystarczający. Z tego względu niektóre stany są eliminowane za pomocą reguł wyczerpywania, dominacji i sondowania.

4.3. OGRANICZENIA CZASOWE

Rozważmy problem alokacji operacji z uwzględnieniem ograniczeń czasowych. Ograniczenia tego typu występują pomiędzy parami operacji. Polegają one na określeniu przedziału czasu, w którym musi być wykonana operacja realizowana po pewnej operacji. Biorąc pod uwagę wszystkie pary operacji, można dla wybranej operacji wyznaczyć najwcześniejszy i najpóźniejszy termin jej rozpoczęcia. Terminy te są zależne od chwil zakończenia operacji poprzedzających daną operację. A zatem rozważane ograniczenia czasowe mają charakter względny. W punkcie tym założymy, że na każdej stacji znajduje się jedna pozycja dla ewentualnego stanowiska pracy.

4.3.1. Model matematyczny

Załóżmy, że dana jest linia montażowa składająca się z K stacji. Na każdej stacji znajduje się jedna pozycja, w której może być zlokalizowane co najwyżej jedno stanowisko pracy. Dany jest zbiór Ω operacji montażowych ω_n, $n = 1,\ldots,N$. Czasy operacji ϑ_n dane są wektorem Θ. Ponadto relację kolejności operacji określa macierz Γ. Dany jest cykl c procesu montażu.

Załóżmy, że dane są minimalne czasy, które muszą upłynąć pomiędzy zakończeniem i rozpoczęciem odpowiednich operacji (poprzednika i następnika):

$$\Phi = \left[\varphi_{i,n}\right]_{\substack{i=1,\ldots,N \\ n=1,\ldots,N}} \tag{4.35}$$

gdzie: $\varphi_{i,n}$ - minimalny czas, jaki musi upłynąć pomiędzy zakończeniem operacji ω_i (poprzednika) i rozpoczęciem operacji ω_n (następnika). Analogicznie przyjmiemy maksymalne czasy, jakie mogą upłynąć pomiędzy zakończeniem i rozpoczęciem odpowiednich operacji:

$$\Psi = \left[\psi_{i,n}\right]_{\substack{i=1,\ldots,N \\ n=1,\ldots,N}} \tag{4.36}$$

gdzie: $\psi_{i,n}$ - maksymalny czas jaki może upłynąć pomiędzy zakończeniem operacji ω_i i rozpoczęciem operacji ω_n.

Jako kryterium optymalizacji alokacji operacji przyjmiemy minimalizację liczby stanowisk pracy. Funkcja celu ma postać (4.3). Wprowadzając oznaczenie t_n dla chwili zakończenia operacji ω_n, można model matematyczny przedstawić w postaci:

- kryterium

$$Q = \sum_{k=1}^{k=K} q_k \longrightarrow \min \tag{4.37}$$

przy czym

$$q_k = \begin{cases} 1, \text{ jeśli } \exists_n \left[\frac{1}{c} t_n\right]^+ = k \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{4.37a}$$

- ograniczeń

$$\bigwedge_{1 \le n \le N} \left[\frac{1}{c} t_n\right]^+ = \left[\frac{1}{c}(t_n - \psi_n)\right]^+ \tag{4.38a}$$

$$\bigwedge_n \bigwedge_\eta (\varphi_{\eta,n}=1) \Longrightarrow (t_\eta \le t_n - \psi_n) \tag{4.38b}$$

$$\bigwedge_n \bigwedge_i t_i + \varphi_{i,n} \le t_n - \psi_n \le t_i + \psi_{i,n} \tag{4.38c}$$

Wyrażenie (4.38a) jest warunkiem niepodzielności operacji, (4.38b) - określa dopuszczalną kolejność operacji, natomiast (4.38c) - wyraża ograniczenia czasowe operacji.

4.3.2. Algorytm

Do rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. Stan $P^{\eta,1}$ jest wektorem o współrzędnych $p_n^{\eta,1} = t_n$, jeśli operacja ω_n została przydzielona do stanowiska pracy, natomiast $p_n^{\eta,1} = 0$, w przypadku przeciwnym. Analogicznie jak w punkcie 4.2.2 wyznaczyć można: numer stacji k_n, do której przydzielono operację ω_n, chwilę $T^{\eta,1}$ zakończenia montażu wszystkich operacji należących do stanu $P^{\eta,1}$ oraz numer stacji $K^{\eta,1}$, na której wykonano ostatnią operację w stanie $P^{\eta,1}$. Przyjmiemy definicję (4.10) wartości stanu. Jednakże z uwagi na ograniczenia czasowe operacji dla trajektorii perspektywicznych spełnienie warunku (4.12) nie jest konieczne. Optymalny stan końcowy wynika z (4.13).

G e n e r o w a n i e s t a n ó w

Oznaczmy przez δt krok dyskretyzacji czasu a przez ρ_n^r $(r=1,\ldots,R)$ prognozowaną chwilę rozpoczęcia operacji ω_n. Procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\eta \bigwedge_r (p_n^{\eta-1,\lambda} = 0) \wedge \left[(\varphi_{\eta,n} = 1) \Longrightarrow (p_\eta^{\eta-1,\lambda} > 0)\right] \wedge$$

$$\wedge (t_n^r - \psi_n \le T_{n,\psi}^{\eta-1,\lambda}) \Longrightarrow (P^r = P^{\eta-1,\lambda} + \Delta P^r) \tag{4.39}$$

Elementy wektora ΔP^r określamy tak jak w (4.14). Chwilę t_n^r wyznaczamy ze wzoru:

$$t_n^r = \begin{cases} \rho_n^r + \psi_n, \text{ jeśli } \left[\frac{1}{c}\rho_n^r\right]^+ = \left[\frac{1}{c}(\rho_n^r + \psi_n)\right]^+ \\ \left[\frac{1}{c}\rho_n^r\right]^+ c + \psi_n, \text{ w przypadku przeciwnym} \end{cases} \tag{4.40}$$

przy czym

$$\rho_n^r = \max (T^{\eta-1,\lambda}, T_{n,\varphi}^{\eta-1,\lambda}) + (r-1)\delta t \tag{4.40a}$$

Przez $T_{n,\varphi}^{\eta-1,\lambda}$ oznaczono chwilę najwcześniejszego (a przez $T_{n,\psi}^{\eta-1,\lambda}$ chwilę najpóźniejszego) rozpoczęcia operacji ω_n, gdzie:

$$T_{n,\varphi}^{\eta-1,\lambda} = \max_{\omega_i \in \Omega^{\eta-1,\lambda}} (p_i^{\eta-1,\lambda} + \varphi_{i,n}) \tag{4.41}$$

oraz

$$T_{n,\psi}^{\eta-1,\lambda} = \min_{\omega_i \in \Omega^{\eta-1,\lambda}} (p_i^{\eta-1,\lambda} + \psi_{i,n}) \tag{4.42}$$

Zbiór $\Omega^{\eta-1,\lambda}$ jest określony przez (4.27).

Równocześnie z generowaniem stanów wyznaczamy wartość stanu stosując wzór rekurencyjny:

$$v^{\eta,1} = \begin{cases} v^{\eta-1,\lambda}, \text{ jeśli } \left[\frac{1}{c} t_n\right]^+ = K^{\eta-1,\lambda} \\ v^{\eta-1,\lambda} + 1, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.43}$$

Dla stanu początkowego przyjmujemy $v^{0,1} = 0$.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

W rozważanym problemie stany nieperspektywiczne są eliminowane za pomocą reguł wyczerpywania, sondowania oraz dominacji.

Reguła wyczerpywania ma postać twierdzenia.

TWIERDZENIE 4.3

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$\exists_n \exists_i \; (p_n^{\eta,l}=0) \wedge (p_i^{\eta,l} > 0) \wedge (p_i^{\eta,l} + \psi_{i,n} < t_n^1 - \vartheta_n) \vee (T^{\eta,l} + \sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_\nu > K.c) \tag{4.44}$$

D o w ó d: Zgodnie z procedurę (4.39) operacja ω_n spełniająca warunek (4.44) nie pozwala wygenerować kolejnego stanu, bowiem np. chwila t_n^1 wyznaczona z (4.40) nie spełnia warunku $t_n^1 - \vartheta_n \leq T_{n,\psi}^{\eta,l}$. Generowanie stanów z $P^{\eta,l}$ przez przydzielanie innych operacji, które spełniają (4.39), nie prowadzi do dopuszczalnego stanu końcowego. Wynika to z faktu, że dla wyznaczenia rozwiązania dopuszczalnego trzeba wygenerować stan N-tego etapu, a zatem przydzielić do realizacji wszystkie operacje. Skoro w stanie $P^{\eta,l}$ nie można przydzielić operacji ω_n, to nie można jej przydzielić również w stanach następujących po $P^{\eta,l}$. A zatem stan $P^{\eta,l}$ jest nieperspektywiczny, bo wyczerpane zostały możliwości generowania rozwiązań dopuszczalnych.

Reguła sondowania ma postać twierdzenia.

TWIERDZENIE 4.4

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$V^{\eta,l} - 1 + \left[\frac{1}{c}\left\{T^{\eta,l} - (K^{\eta,l}-1).c + \sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_n\right\}\right]^+ \geq V^a \tag{4.45}$$

D o w ó d: Zauważmy, że lewa strona nierówności (4.45) jest oszacowaniem $b^{\eta,l}$ liczby stanowisk pracy dla stanu lokalnie optymalnego P^{N,l^0}. Wzór ten nie pokrywa się z (4.26), ponieważ w przypadku ograniczeń czasowych niektóre stacje mogę nie zawierać stanowisk pracy.
A zatem:

$$V^{\eta,l} \leq \left[\frac{1}{c} T^{\eta,l}\right]^+ \tag{4.45a}$$

Dalsza część dowodu przebiega tak jak dla twierdzenia 4.1.

Reguła dominacji jest oparta na twierdzeniu.

TWIERDZENIE 4.5

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli spełniony jest warunek:

$$\bigwedge_{\omega_n \notin \Omega^{\eta,l_1}} (\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T_{n,\varphi}^{\eta,l_1} \leq T_{n,\varphi}^{\eta,l_2}) \wedge (T_{n,\psi}^{\eta,l_2} \leq T_{n,\psi}^{\eta,l_1}) \wedge \tag{4.46}$$

$$\wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge \left[(V^{\eta,l_1} < V^{\eta,l_2}) \vee (V^{\eta,l_1} \leq V^{\eta,l_2}) \wedge (K^{\eta-1,l_1} = K^{\eta,l_2})\right]$$

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia 4.2.

W oparciu o regułę dominacji stanów można wykazać, że procedura generowania stanów spełnia warunek konieczny generowania stanów perspektywicznych. Zgodnie ze wzorem (4.40) operacja ω_n jest realizowana w dopuszczalnym przedziale czasu.

4.4. OGRANICZENIA ZAKRESOWE

Rozważmy procee montażu, w którym stanowisko pracy na każdej stacji montażowej może być zlokalizowane tylko w jednej pozycji. Z uwagi na lokalizację detali oraz zasięg narzędzi każda operacja może być wykonywana tylko na pewnych (kolejnych) stacjach montażowych. Stacje te wyznaczają zakres linii, w którym może być realizowana dana operacja. W ogólnym przypadku czas operacji jest zależny od stacji montażowej, w której operacja ta jest wykonywana.

4.4.1. Model matematyczny

Załóżmy, że dana jest linia montażowa składająca się z K stacji. Na każdej stacji może być zlokalizowane najwyżej jedno stanowisko pracy. Dla kompletnego montażu każdego obiektu trzeba wykonać wszystkie operacje ω_n, $n = 1,\ldots,N$, ze zbioru Ω. Relacja kolejności operacji dana jest macierzą Γ, a cykl montażu wynosi c.

Czasy wykonania operacji dane są macierzą:

$$\Theta = \left[\vartheta_{n,k}\right]_{\substack{n=1,\ldots,N \\ k=1,\ldots,K}} \tag{4.47}$$

gdzie: $\vartheta_{n,k}$ - czas wykonania operacji ω_n w k-tej stacji montażowej.
Załóżmy, że cykl c spełnia warunek:

$$\bigwedge_{1 \leq n \leq N} \exists_{1 \leq k \leq K} \vartheta_{n,k} < c \tag{4.48}$$

A zatem jeśli pewne elementy macierzy (4.47) nie spełniają warunku (4.48), to znaczy, że dana operacja nie może być wykonana na odpowiedniej stacji.

Załóżmy, że dla każdej operacji ω_n istnieje stacja $\mathcal{K}_n$, w której operację tę można wykonać w najkrótszym czasie $\vartheta_{n,\mathcal{K}_n}$, tzn.:

$$\bigwedge_{1\le n\le N}\ \bigexists_{1\le \mathcal{K}_n\le K}\ \bigwedge_{1\le i<\mathcal{K}_n}\ \bigwedge_{\mathcal{K}_n<j<K} (\vartheta_{n,\mathcal{K}_n} < \vartheta_{n,i} \le \vartheta_{n,i-1}) \wedge$$

$$\wedge (\vartheta_{n,\mathcal{K}_n} < \vartheta_{n,j} \le \vartheta_{n,j+1}) \tag{4.49}$$

Ponadto dla każdej operacji ω_n można określić przedział stacji montażowych $\left[k_n^1, k_n^2\right]$ - zwany <u>zakresem</u>, w których operacja ta może być wykonana, czyli:

$$\bigwedge_{k_n^1\le k\le k_n^2} \vartheta_{n,k} \le c \tag{4.50}$$

Wartości k_n^1 oraz k_n^2 wyznaczamy ze wzorów:

$$k_n^1 = \begin{cases} 1, \text{ jeśli } \vartheta_{n,1} < c \\ i_n, \text{ jeśli } (\vartheta_{n,i_n-1} > c) \wedge (\vartheta_{n,i_n} \le c) \end{cases} \tag{4.51a}$$

oraz

$$k_n^2 = \begin{cases} K, \text{ jeśli } \vartheta_{n,K} \le c \\ j_n, \text{ jeśli } (\vartheta_{n,j_n} \le c) \wedge (\vartheta_{n,j_n+1} > c) \end{cases} \tag{4.51b}$$

Problem alokacji operacji z ograniczeniami zakresowymi polega na wyznaczeniu dopuszczalnych podzbiorów operacji Ω_k, $k = 1,\ldots,K$. Jako cel optymalizacji przyjmiemy minimalizację liczby stanowisk pracy.

Wprowadzając zmienną t_n (dla chwili zakończenia operacji ω_n) przedstawimy model matematyczny alokacji operacji z ograniczeniami zakresowymi w następującej postaci:

- kryterium

$$Q = \sum_{k=1}^{k=K} q_k \longrightarrow \min \tag{4.52}$$

przy czym

$$q_k = \begin{cases} 1, \text{ jeśli } \bigexists_n \left[\frac{1}{c} t_n\right]^+ = k \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{4.52a}$$

- ograniczenia

$$\bigwedge_{1\le n\le N} \left\{\left[\frac{1}{c} t_n\right]^+ = k\right\} \Rightarrow \left\{\left[\frac{1}{c}(t_n - \vartheta_{n,k})\right]^+ = k\right\} \tag{4.53a}$$

$$\bigwedge_n \bigwedge_\nu \left\{(\vartheta_{\nu,n-1}) \wedge \left\{\left[\frac{1}{c} t_n\right]^+ = k\right\}\right\} \Rightarrow (t_\nu < t_n - \vartheta_{n,k}) \tag{4.53b}$$

$$\bigwedge_{1\le n\le N} k_n^1 \le \left[\frac{1}{c} t_n\right]^+ \le k_n^2 \tag{4.53c}$$

Wyrażenie (4.53a) jest warunkiem niepodzielności, a (4.53b) warunkiem kolejności operacji. Ograniczenie zakresowe przedstawia wyrażenie (4.53c).

4.4.2. <u>Algorytm</u>

Do rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. Stan $P^{\eta,l}$ zdefiniujemy tak jak dla klasycznego problemu BLM. Ze stanu $P^{\eta,l}$ można wyznaczyć: lokalizację operacji ω_n do stacji k_n, chwilę $T^{\eta,l}$ zakończenia wszystkich operacji oraz numer $K^{\eta,l}$ ostatniej wykorzystanej stacji w stanie $P^{\eta,l}$. Przyjmiemy definicję (4.10) wartości stanu. Z uwagi na ograniczenia zakresowe operacji, spełnienie warunku (4.12) dla trajektorii perspektywicznych nie jest konieczne. Optymalny stan końcowy wynika z (4.13).

G e n e r o w a n i e s t a n ó w

Załóżmy, że $P^{\eta-1,\lambda}$ jest wybranym stanem aktywnym, z którego za pomocą operacji ω_n należy wygenerować stan P. Wyróżnimy trzy wzajemnie wykluczające się przypadki generowania stanów.

I. Jeżeli jest spełniony warunek:

$$K^{\eta-1,\lambda} < k_n^1 \tag{4.54}$$

to operację ω_n należy przydzielić do stacji k z przedziału:

$$k_n^1 \le k \le k_n^2 \tag{4.54a}$$

II. Jeżeli jest spełniony warunek:

$$k_n^1 \leq K^{\eta-1,\lambda} < k_n^2 \tag{4.55}$$

to operację ω_n należy przydzielać do stacji k z przedziału:

$$K^{\eta-1,\lambda} < k \leq k_n^2 \tag{4.55a}$$

Ponadto w przypadku gdy:

$$c \cdot K^{\eta-1,\lambda} - T^{\eta-1,\lambda} \geq \vartheta^{\eta}_{n,K^{\eta-1,\lambda}} \tag{4.55b}$$

to operację ω_n przydziela się jedynie do stacji $K^{\eta-1,\lambda}$.

III. Jeżeli jest spełniony warunek:

$$(K^{\eta-1,\lambda} = k_n^2) \wedge (c \cdot K^{\eta-1,\lambda} - T^{\eta-1,\lambda} \geq \vartheta^{\eta}_{n,K^{\eta-1,\lambda}}) \tag{4.56}$$

to operację ω_n należy przydzielić do stacji $K^{\eta-1,\lambda}$.

Pierwsza procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_{\nu} \bigwedge_{k_n^1 \leq k \leq k_n^2} (K^{\eta-1,\lambda} < k_n^1) \wedge (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0)\right] \Rightarrow$$
$$\Rightarrow (P^k = P^{\eta-1,\lambda} + \Delta P^k) \tag{4.57}$$

Elementy wektora ΔP^k określone są następująco:

$$\Delta p_i^k = \begin{cases} t_n^k, & \text{dla } i=n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{4.58}$$

Chwilę t_n^k wyznaczamy ze wzoru:

$$t_n^k = (k-1) \cdot c + \vartheta^{\eta}_{n,k} \quad k=k_n^1,\ldots,k_n^2 \tag{4.59}$$

Druga procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_{\nu} \bigwedge_{K^{\eta-1,\lambda} \leq k \leq k_n^2} (k_n^1 \leq K^{\eta-1,\lambda} < k_n^2) \wedge (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow \right.$$
$$\left. \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0)\right] \Rightarrow (P^k = P^{\eta-1,\lambda} + \Delta P^k) \tag{4.60}$$

Elementy wektora ΔP^k są określone przez (4.58), natomiast chwilę t_n^k wyznaczamy ze wzoru:

$$t_n^k = \begin{cases} T^{\eta-1,\lambda} + \vartheta^{\eta}_{n,K^{\eta-1,\lambda}}, & \text{dla } (k = K^{\eta-1,\lambda}) \wedge (cK^{\eta-1,\lambda} + \\ & \quad - T^{\eta-1,\lambda} \geq \vartheta^{\eta}_{n,K^{\eta-1,\lambda}}) \\ (k-1)c + \vartheta^{\eta}_{n,k}, & \text{dla } K^{\eta-1,\lambda} < k \leq k_n^2 \end{cases} \tag{4.61}$$

Trzecia procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_{\nu} (K^{\eta-1,\lambda} = k_n^2) \wedge (p_n^{\eta-1,\lambda} = 0) \wedge (cK^{\eta-1,\lambda} - T^{\eta-1,\lambda} \geq \vartheta^{\eta}_{n,K^{\eta-1,\lambda}}) \wedge$$
$$\wedge \left[(\vartheta_{\nu,n} = 1) \Rightarrow (p_{\nu}^{\eta-1,\lambda} > 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{4.62}$$

Elementy wektora ΔP są określone analogicznie jak w (4.58), natomiast chwilę t_n wyznaczamy ze wzoru:

$$t_n = T^{\eta-1,\lambda} + \vartheta^{\eta}_{n,K^{\eta-1,\lambda}} \tag{4.63}$$

Procedury (4.57), (4.60) i (4.61) pozwalają wygenerować wszystkie bezpośrednie następniki P stanu $P^{\eta-1,\lambda}$. Dla operacji ω_n wykorzystywana jest jedna z tych procedur. Operacja ω_n jest przydzielana kolejno do każdej dopuszczalnej stacji k.

Eliminowanie stanów nieperspektywicznych

Do eliminacji stanów nieperspektywicznych wykorzystuje się reguły wyczerpywania, sondowania oraz dominacji.

Reguła wyczerpywania jest oparta na następującym twierdzeniu.

TWIERDZENIE 4.6

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełniony jest warunek:

$$\exists_n (\rho_n^{\gamma,l}=0) \wedge \left[(K^{\gamma,l} > k_n^2) \vee (K^{\gamma,l}=k_n^2) \wedge (K^{\gamma,l}_{.c} - T^{\gamma,l} < \vartheta_{n,K^{\gamma,l}})\right] \quad (4.64)$$

D o w ó d tego twierdzenia wynika wprost z procedury generowania stanów (4.62). Jeżeli operacja ω_n spełnia warunek (4.64), to nie można jej przydzielić do realizacji. A zatem ze stanu $P^{\gamma,l}$ nie można otrzymać dopuszczalnego rozwiązania problemu.

Reguła sondowania ma postać twierdzenia.

TWIERDZENIE 4.7

Stan $P^{\gamma,l}$ jest nieperspektywiczny, jeżeli spełniony jest warunek:

$$V^{\gamma,l} - 1 + \left[\frac{1}{c}(T^{\gamma,l} - (K^{\gamma,l}-1).c + \sum_{\omega_n \notin \Omega^{\gamma,l}} \vartheta_{n,\mathcal{R}_n})\right]^+ > V^a \quad (4.65)$$

Dla d o w o d u wystarczy zauważyć, że lewa strona nierówności (4.65) jest dolnym ograniczeniem wartości V^{N,l^o} stanu lokalnie optymalnego P^{N,l^o}.

W oszacowaniu tego ograniczenia przyjęto optymistycznie, że każda operacja $\omega_n \notin \Omega^{\gamma,l}$ jest wykonana w najkrótszym czasie $\vartheta_{n,\mathcal{R}_n}$, a ponadto na odpowiednich stanowiskach pracy nie występują luzy czasu. Jeżeli spełniony jest warunek (4.65), to ze stanu $P^{\gamma,l}$ nie można otrzymać lepszego stanu końcowego niż P^a. □

Reguła dominacji jest oparta na następującym twierdzeniu.

TWIERDZENIE 4.8

Stan P^{γ,l_1} dominuje nad stanem P^{γ,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\gamma,l_1} = \Omega^{\gamma,l_2}) \wedge (T^{\gamma,l_1} \leq T^{\gamma,l_2}) \wedge \left[(V^{\gamma,l_1} < V^{\gamma,l_2}) \vee (K^{\gamma,l_1} = K^{\gamma,l_2}) \wedge (V^{\gamma,l_1} \leq V^{\gamma,l_2})\right] \quad (4.66)$$

D o w ó d: Załóżmy, że dana jest trajektoria lokalnie optymalna $P^{\gamma,l_2}, \ldots, P^{N,l_2^o}$. Trajektorii tej odpowiada pewien przydział operacji do stacji montażowych. Jeżeli $\Omega^{\gamma,l_1} = \Omega^{\gamma,l_2}$ i $T^{\gamma,l_1} \leq T^{\gamma,l_2}$, to przydział ten jest realizowalny od stanu P^{γ,l_1}. W ten sposób można otrzymać trajektorię $P^{\gamma,l_1}, \ldots, P^{N,\lambda}$, gdzie stan $P^{N,\lambda}$ na ogół nie jest stanem lokalnie optymalnym P^{N,l_1^o}. A zatem możemy napisać:

$$V^{N,l_2^o} = V^{\gamma,l_2} + \Delta V^{\gamma,l_2} \quad (4.67a)$$

oraz

$$V^{N,\lambda} = V^{\gamma,l_1} + \Delta V^{\gamma,l_1} \quad (4.67b)$$

Ponieważ:

$$\Delta V^{\gamma,l_1} = \begin{cases} \Delta V^{\gamma,l_2}, \text{ jeśli } K^{\gamma,l_1} = K^{\gamma,l_2} \\ \Delta V^{\gamma,l_2}+1, \text{ jeśli } K^{\gamma,l_1} < K^{\gamma,l_2} \end{cases} \quad (4.68a)$$

a ponadto w (4.66) żądamy, by:

$$V^{\gamma,l_1} \leq V^{\gamma,l_2} \text{ dla } K^{\gamma,l_1} = K^{\gamma,l_2} \quad (4.68b)$$

lub

$$V^{\gamma,l_1} < V^{\gamma,l_2} \text{ dla } K^{\gamma,l_1} < K^{\gamma,l_2} \quad (4.68c)$$

Zatem z (4.67) i (4.68) wynika, że:

- dla $K^{\gamma,l_1} = K^{\gamma,l_2}$

$$V^{N,\lambda} = V^{\gamma,l_1} + \Delta V^{\gamma,l_1} \leq V^{\gamma,l_2} + \Delta V^{\gamma,l_2} = V^{N,l_2^o} \quad (4.69a)$$

- dla $K^{\gamma,l_1} < K^{\gamma,l_2}$

$$V^{N,\lambda} = V^{\gamma,l_1} + \Delta V^{\gamma,l_1} < V^{\gamma,l_2} + \Delta V^{\gamma,l_2} + 1 = V^{N,l_2^o} + 1 \quad (4.69b)$$

czyli:

$$V^{N,\lambda} < V^{N,l_2^o} \quad (4.69c)$$

Ponadto:

$$V^{N,l_1^o} \leq V^{N,\lambda} \quad (4.70)$$

A zatem z (4.69) i (4.70) otrzymujemy ostatecznie:

$$V^{N,l_1^o} \leq V^{N,l_2^o} \tag{4.71}$$

stąd wynika, że stan P^{φ,l_1} dominuje nad stanem P^{φ,l_2}. □

4.5. LINIA ASYMETRYCZNA

Linia asymetryczna składa się ze stacji, na których wyróżnia się dwie pozycje - po lewej oraz po prawej stronie osi linii. Stanowiska pracy są szeregowe, jeżeli na każdej stacji wykorzystywana jest co najwyżej jedna pozycja. W przypadku przeciwnym stanowiska pracy są równoległe. Rozważymy linię o stanowiskach szeregowych.

4.5.1. Model matematyczny

Załóżmy, że linia montażowa składa się z K stacji. Na każdej stacji wyróżniamy J=2 pozycje. Niech j=1 dla lewej oraz j=2 dla prawej strony linii. Przyjmiemy, że proces montażu jest sekwencyjny. Dany jest zbiór Ω operacji montażowych ω_n, n=1,...,N. Relację kolejności operacji określa macierz Γ. Czasy operacji dane są macierzą:

$$\Theta = \left[\vartheta_{n,j}\right]_{\substack{n=1,\dots,N \\ j=1,2}} \tag{4.72}$$

gdzie: $\vartheta_{n,j}$ - czas wykonania operacji ω_n po j-tej stronie linii. Cykl c procesu montażu spełnia warunek:

$$\max_{1\leq n\leq N}\ (\min_{1\leq j\leq 2}\ \vartheta_{n,j}) \leq c \tag{4.73}$$

A zatem z (4.73) wynika, że jeżeli:

$$c < \vartheta_{n,j} \tag{4.74}$$

to operacja ω_n nie może być wykonana po j-tej stronie linii. Przez $\Omega_{k,j}$ oznaczymy podzbiór operacji tworzący stanowisko pracy po j-tej stronie linii.

Jako cel optymalizacji alokacji operacji przyjmiemy minimalizację liczby stanowisk pracy.

Oznaczmy przez t_n chwilę zakończenia operacji. Aby wyróżnić stronę linii, po której jest wykonywana operacja ω_n, przyjmiemy kodowanie chwili t_n. Dla j=1 (lewa strona linii) będziemy zapisywać $t_n < 0$, natomiast

dla j=2 (prawa strona linii) zapiszemy $t_n > 0$. A zatem lokalizację operacji ω_n wyznaczamy ze wzoru:

$$k_n = \left[\tfrac{1}{c}|t_n|\right]^+ \operatorname{sign} t_n \tag{4.75}$$

gdzie: k_n - numer stanowiska pracy, do którego przydzielono operację ω_n. Jeżeli $k_n < 0$, to operacja ω_n jest wykonywana po lewej stronie linii, natomiast dla $k_n > 0$, operacja ω_n jest wykonywana po prawej stronie linii.

Przy powyższych założeniach model matematyczny alokacji operacji można przedstawić w postaci:

- kryterium

$$Q = \sum_{k=1}^{k=K} q_k \tag{4.76}$$

przy czym

$$q_k = \begin{cases} 1, \text{ jeśli } \exists_n \left[\tfrac{1}{c}|t_n|\right]^+ = k \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{4.76a}$$

- ograniczenia

$$\bigvee_{1\leq n\leq N} \left[\tfrac{1}{c}|t_n|\right]^+ = \left[\tfrac{1}{c}(|t_n| - \vartheta_{n,j})\right]^+ \tag{4.77a}$$

przy czym

$$j = \begin{cases} 1, \text{ jeśli } t_n < 0 \\ 2, \text{ w przypadku przeciwnym} \end{cases} \tag{4.77b}$$

$$\bigvee_n \bigvee_\nu (\vartheta_{\nu,n}=1) \Rightarrow (|t_\nu| \leq |t_n| - \vartheta_{n,j}) \tag{4.77c}$$

$$\bigvee_n \bigvee_{i\neq n} \left\{\left[\tfrac{1}{c}|t_n|\right]^+ \operatorname{sign} t_n = k_n\right\} \Rightarrow \left\{\left[\tfrac{1}{c}|t_i|\right]^+ \operatorname{sing} t_i \neq -k_n\right\} \tag{4.77d}$$

Wyrażenie (4.77a) interpretuje warunek niepodzielności, natomiast wyrażenie (4.77c) - kolejności operacji. Ograniczenie lokalizacji operacji przedstawia warunek (4.77d).

4.5.2. Algorytm

Do rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. Z uwagi na kodowanie chwil t_n niektóre określenia ulegną modyfikacji.

Definicja 4.2

Stan jest wektorem:

$$P^{\eta,l} = \left[p_n^{\eta,l}\right]_{n=1,\ldots,N} \tag{4.78}$$

Elementy tego wektora określamy następująco:

$$p_n^{\eta,l} = \begin{cases} t_n, \text{ jeśli operacja } \omega_n \text{ została przydzielona do stanowiska pracy} \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{4.78a}$$

A zatem jeśli $p_n^{\eta,l} < 0$, to operację ω_n przydzielono do stanowiska pracy po lewej stronie linii, natomiast jeśli $p_n^{\eta,l} > 0$, to po prawej.
Ze stanu $P^{\eta,l}$ otrzymujemy:

$$k_n = \left[\frac{1}{c}\left|p_n^{\eta,l}\right|\right]^+ \operatorname{sign} p_n^{\eta,l} \tag{4.79}$$

Chwilę $T^{\eta,l}$, w której kończy się wykonywanie wszystkich operacji przydzielonych do stanowisk pracy w stanie $P^{\eta,l}$ wyznaczamy ze wzoru:

$$T^{\eta,l} = \max_{1\le i\le N}\left|p_i^{\eta,l}\right| \tag{4.80}$$

Tak więc z (4.9) otrzymujemy $K^{\eta,l}$.
Wartość stanu określamy tak jak w (4.10), z tym że:

$$q_k^{\eta,l} = \begin{cases} 1, \text{ jeśli } \exists_n \left[\frac{1}{c}\left|p_n^{\eta,l}\right|\right]^+ = k \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{4.81}$$

Dla trajektorii perspektywicznych musi być spełniony warunek (4.12), a optymalny stan końcowy P^o wyznaczamy z warunku (4.13).

G e n e r o w a n i e s t a n ó w

Rozważmy algorytm jednokrokowy, w którym przydzielając do wybranego stanu $P^{\eta-1,\lambda}$ operację ω_n, otrzymujemy stan P. Oznaczając przez ω_μ ostatnią operację wykonaną w stanie $P^{\eta-1,\lambda}$ mamy:

$$T^{\eta-1,\lambda} = \left|p_\mu^{\eta-1,\lambda}\right| \tag{4.82}$$

W celu wygenerowania wszystkich dopuszczalnych i perspektywicznych trajektorii stosowane są dwie procedury.

1) Jeżeli operacja ω_n nie mieści się na ostatnim stanowisku pracy, tzn. spełniony jest warunek:

$$(p_\mu^{\eta-1,\lambda} < 0) \wedge (K^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} < \vartheta_{n,1}) \vee (p_\mu^{\eta-1,\lambda} > 0) \wedge$$
$$\wedge (K^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} < \vartheta_{n,2}) \tag{4.83}$$

to przydzielamy ją na lewe oraz prawe stanowisko następnej stacji (o ile przydział taki jest dopuszczalny).
Procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\nu \bigwedge_{1\le j\le 2} (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p_\nu^{\eta-1,\lambda}\neq 0)\right] \wedge (\vartheta_{n,j} \le c) \Rightarrow$$
$$\Rightarrow (P^j = P^{\eta-1,\lambda} + \Delta P^j) \tag{4.84}$$

Elementy wektora ΔP^j określamy tak jak w (4.17). Chwilę t_n^j (zakodowaną) wyznaczamy z formuły:

$$t_n^j = \begin{cases} -(K^{\eta-1,\lambda} \cdot c + \vartheta_{n,1}), & \text{dla } j=1 \\ K^{\eta-1,\lambda} \cdot c + \vartheta_{n,2}, & \text{dla } j=2 \end{cases} \tag{4.85}$$

Jeżeli operacja ω_n może być wykonana po obydwu stronach linii, to ze stanu $P^{\eta-1,\lambda}$ generujemy dwa stany P, wykorzystując obie wartości z (4.85).

2) Jeżeli operacja ω_n mieści się na ostatnim stanowisku pracy, tzn. warunek (4.83) nie jest spełniony, to przydzielamy ją tylko na to stanowisko.
Procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\nu (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p_\nu^{\eta-1,\lambda}\neq 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{4.86}$$

Elementy wektora ΔP są określone przez (4.17). Chwilę t_n (zakodowaną) wyznaczamy ze wzoru:

$$t_n = \begin{cases} -(T^{\eta-1,\cdot} + \vartheta_{n,1}), & \text{gdy } p_{\mu}^{\eta-1,\lambda} < 0 \\ T^{\eta-1,\lambda} + \vartheta_{n,2}, & \text{w przypadku przeciwnym.} \end{cases} \tag{4.87}$$

Celowość przydzielania operacji na jak najwcześniejsze stanowisko pracy wynika z reguły dominacji stanów, która zostanie podana niżej. Równocześnie z generowaniem stanu P wyznaczamy jego wartość

$$V = \left[\frac{1}{c}\,|t_n|\right]^+ \tag{4.88a}$$

a ponadto:

$$T = |t_n| \tag{4.88b}$$

Wygenerowany stan P jest testowany na perspektywiczność.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Stany nieperspektywiczne eliminujemy za pomocą reguł dominacji oraz sondowania. Reguła dominacji jest oparta na następującym twierdzeniu:

TWIERDZENIE 4.9

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge \left[(K^{\eta,l_1} < K^{\eta,l_2}) \vee (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge \right.$$

$$\left. \wedge\, (K^{\eta,l_1} = K^{\eta,l_2}) \wedge (p_{\mu_1^{\eta}}^{\eta,l_1} \cdot p_{\mu_2^{\eta}}^{\eta,l_2} > 0)\right] \tag{4.89}$$

gdzie: ω_{μ_1} (ω_{μ_2}) - ostatnia operacja wykonana w stanie P^{η,l_1} (P^{η,l_2}).

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia 4.2.

TWIERDZENIE 4.10

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$\left[\frac{1}{c}(T^{\eta,l} + \sum_{\omega_i \notin \Omega^{\eta,l}} \min_{1 \leq j \leq 2} \vartheta_{i,j})\right]^+ \geq V^a \tag{4.90}$$

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia 4.1.

Anelogicznie do lematu 4.1 można udowodnić, że przedstawione procedury (4.84) i (4.86) spełniają warunek perspektywiczności generowanych stanów.

5. ALOKACJA OPERACJI. RÓWNOLEGŁE STANOWISKA PRACY

5.1. WSTĘP

Linie montażowa składa się z transportera, na którym w równych odstępach znajduję się montowane obiekty. Wzdłuż linii rozmieszczone są stacje montażowe tak, by na każdej stacji znajdował się tylko jeden obiekt. W liniach z szeregowymi stanowiskami pracy na stacji montażowej może znajdować się co najwyżej jedno stanowisko. W liniach z równoległymi stanowiskami pracy warunek ten nie jest spełniony.

W rozdziale tym rozważymy linie montażowe z równoległymi stanowiskami pracy przyjmując, że na stacji montażowej mogą znajdować się co najwyżej dwa stanowiska - zlokalizowane po obu stronach osi linii. Wyróżnimy model podstawowy, w którym występuje tylko relacja kolejności operacji. Ponadto rozważymy modele z dodatkowymi relacjami: wykluczania oraz synchronizacji operacji. Relacje te są charakterystyczne dla linii z równoległymi stanowiskami pracy. Z tego względu skupimy na nich uwagę, pomijając elementy przedstawione w uogólnionych modelach BLM oraz modelach AO w liniach z szeregowymi stanowiskami pracy.

Linie z równoległymi stanowiskami pracy mogą wynikać z postulatu koncentracji procesu montażu (minimalizacja długości linii). W liniach o określonej długości (liczbie stacji montażowych) o organizacji stanowisk szeregowych lub równoległych decyduje cykl montażu.

Rozwiązanie problemu BLM lub AO w linii z szeregowymi stanowiskami pracy może - przy zbyt krótkim cyklu - nie być możliwe. W takim przypadku powstaje problem AO w linii z równoległymi stanowiskami pracy.

5.2. MODEL PODSTAWOWY

W podstawowym modelu AO w linii z równoległymi stanowiskami pracy uwzględnimy jedynie założenia przyjęte dla klasycznego modelu BLM. Jednakże operacje mogą być przydzielane do stanowisk pracy znajdujęcych się po lewej lub po prawej stronie osi linii. Jako cel optymalizacji AO przyjmiemy minimalizację liczby stanowisk pracy.

5.2.1. Model matematyczny

Załóżmy, że dany jest zbiór Ω operacji montażowych ω_n, $n = 1,\ldots,N$. Dla kompletnego zmontowania jednego obiektu muszą być wykonane wszystkie

operacje ze zbioru Ω. Relacja kolejności operacji dana jest macierzę Γ, a czasy operacji - wektorem Φ. Dany jest cykl c montażu, spełniający warunek:

$$\max_{1\leq n\leq N} \vartheta_n \leq c < \sum_{n=1}^{n=N} \vartheta_n \tag{5.1}$$

A zatem każda operacja może być w całości wykonana na jednym stanowisku pracy.

Linia montażowa składa się z K stacji. Na każdej stacji mogą się znajdować co najwyżej dwa stanowiska pracy (po lewej oraz po prawej stronie osi linii). Każdy obiekt przebywa w czasie cyklu c na jednej stacji, a w czasie $K \cdot c$ - na linii. Tym samym czas montażu obiektu nie może przekroczyć $2K \cdot c$.

Jeżeli spełniony jest warunek:

$$K < \left[\frac{1}{c}\sum_{n=1}^{n=N} \vartheta_n\right]^{+} \tag{5.2}$$

to w linii montażowej muszą być zorganizowane równoległe stanowiska pracy. Gdy warunek (5.2) nie jest spełniony, można rozwiązywać problem AO (lub BLM) dla szeregowych stanowisk pracy. Jednakże w przypadku braku rozwiązania dopuszczalnego - powstaje do rozwiązania problem AO w linii z równoległymi stanowiskami pracy.

Kryterium optymalizacji AO w linii z równoległymi stanowiskami pracy można zapisać w postaci:

$$Q = \sum_{k=1}^{k=K}\sum_{j=1}^{j=J} q_{k,j} \longrightarrow \min \tag{5.3}$$

przy tym

$$q_{k,j} = \begin{cases} 1, \text{ jeśli } \Omega_{k,j} \neq 0 \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.4}$$

gdzie: $\Omega_{k,j}$ - podzbiór operacji przydzielonych do j-tego stanowiska k-tej stacji ($j=1$ - lewa strona linii, $j=2$ - prawa strona linii).

Oznaczmy przez t_n chwilę zakończenia operacji ω_n. Dla rozróżnienia strony linii, po której została wykonana operacja ω_n, przyjmiemy, że t_n jest liczbę ujemnę dla $j=1$ oraz dodatnię dla $j=2$. Stosując takie kodowa-

nie możemy wyznaczyć numer stacji k_n, do której przydzielono operację ω_n:

$$k_n = \left[\frac{1}{c}\,|t_n|\right]^+ \tag{5.5}$$

Ponadto lokalizację stanowiska pracy dla operacji ω_n otrzymujemy z:

$$\bigwedge_{1\le k\le K}\ \bigwedge_{n}\ \left\{(t_n < 0) \wedge \left\{\left[\frac{1}{c}\,|t_n|\right]^+ = k\right\}\right\} \Rightarrow (\omega_n \in \Omega_{k,1}) \tag{5.6a}$$

$$\bigwedge_{1\le k\le K}\ \bigwedge_{n}\ \left\{(t_n > 0) \wedge \left\{\left[\frac{1}{c}\,t_n\right]^+ = k\right\}\right\} \Rightarrow (\omega_n \in \Omega_{k,2}) \tag{5.6b}$$

Przy powyższych założeniach możemy przedstawić model matematyczny AO w linii z równoległymi stanowiskami pracy w następującej postaci:

- kryterium określa (5.3), przy czym

$$q_{k,1} = \begin{cases} 1, \text{ jeśli } \bigvee_{\nu}\left\{\left[\frac{1}{c}\,|t_\nu|\right]^+ = k\right\} \wedge (t_\nu < 0) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.7a}$$

$$q_{k,2} = \begin{cases} 1, \text{ jeśli } \bigvee_{n}\left\{\left[\frac{1}{c}\,|t_n|\right]^+ = k\right\} \wedge (t_n > 0) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.7b}$$

- ograniczenia

$$\bigwedge_{1\le n\le N}\left\{(t_n < 0) \Rightarrow \left\{\left[\frac{1}{c}\,|t_n|\right]^+ = \left[\frac{1}{c}\,|t_n + \nu_n|\right]^+\right\} \vee \left\{(t_n > 0) \Rightarrow \right.\right.$$
$$\left.\left. \Rightarrow \left\{\left[\frac{1}{c}\,t_n\right]^+ = \left[\frac{1}{c}\,(t_n - \nu_n)\right]^+\right\}\right\} \tag{5.8a}$$

$$\bigwedge_{1\le k\le K}\ \bigwedge_{j}\ \bigwedge_{n}(\omega_n \in \Omega_{k,j}) \Rightarrow \Big(\sum_{\omega_n \in \Omega_{k,j}} \nu_n \le c\Big) \tag{5.8b}$$

$$\bigwedge_{\nu}\ \bigwedge_{n}\ (\gamma_{\nu,n}=1) \Rightarrow (|t_\nu| \le |t_n| - \nu_n) \tag{5.8c}$$

Z warunku (5.8a) wynika, że każda operacja ω_n musi być wykonana w całości na jednym stanowisku pracy. A zatem wykluczone jest wykonanie operacji ω_n równocześnie na lewym i prawym stanowisku pracy tej samej stacji montażowej. Wyrażenie (5.8b) przedstawia ograniczenie sumy czasów

operacji na stanowisku pracy, natomiast (5.8c) reprezentuje ograniczenia kolejnościowe.

Warto podkreślić, że wyróżnienie w kryterium (5.3) stanowisk pracy po lewej i po prawej stronie osi linii pozwala analizować bardziej złożone modele. Można dla przykładu ograniczyć lub minimalizować liczbę stanowisk pracy po jednej stronie linii.

5.2.2. Algorytm

Do rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. W algorytmie tym wykorzystamy następujące definicje i określenia.

Definicja 5.1

Stan jest wektorem:

$$P^{\eta,l} = \left[p_n^{\eta,l}\right]_{\substack{\eta=0,\dots,N\\ n=1,\dots,N\\ l=1,\dots,L_\eta}} \tag{5.9}$$

Elementy tego wektora określamy następująco:

$$p_n^{\eta,l} = \begin{cases} t_n, \text{ jeśli operacja } \omega_n \text{ została przydzielona do stanowiska pracy} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{5.9a}$$

A zatem jeśli $p_n^{\eta,l} < 0$, to operację ω_n przydzielono do stanowiska pracy po lewej stronie osi linii, natomiast gdy $p_n^{\eta,l} > 0$, to operacja ω_n została przydzielona do stanowiska pracy po prawej stronie osi linii. Stan początkowy $P^{0,1}$ ma wszystkie współrzędne zerowe, natomiast każdy stan końcowy $P^{N,l}$ ma wszystkie współrzędne różne od zera.

Numer stacji montażowej k_n, do której przydzielono operację ω_n, można wyznaczyć znając stan $P^{\eta,l}$, bowiem:

$$k_n = \left[\frac{1}{c}\,\left|p_n^{\eta,l}\right|\right]^+ \tag{5.10}$$

Dla oceny stanu $P^{\eta,l}$ zdefiniujemy funkcję wartości stanu, którą oznaczymy przez V.

Definicja 5.2

Funkcja wartości stanu ma postać:

$$V^{\eta,l} = V(P^{\eta,l}) \tag{5.11}$$

przy tym:

$$v^{\eta,l} = \left[v_j^{\eta,l}\right]_{j=1,2} \tag{5.12}$$

oraz

$$v_j^{\eta,l} = \sum_{k=1}^{k=K_j^{\eta,l}} q_{k,j}^{\eta,l} \tag{5.13}$$

gdzie: $K_j^{\eta,l}$ - numer ostatniej stacji, na której po j-tej stronie linii znajduje się ostatnie stanowisko pracy w stanie $P^{\eta,l}$.

Numer $K_j^{\eta,l}$ wyznaczamy ze wzoru:

$$K_j^{\eta,l} = \left[\frac{1}{c} T_j^{\eta,l}\right]^+ \tag{5.14}$$

gdzie:

$$T_1^{\eta,l} = \left| \min_{1 \le i \le N} p_i^{\eta,l} \right| \tag{5.15}$$

oraz

$$T_2^{\eta,l} = \max_{1 \le i \le N} p_i^{\eta,l} \tag{5.16}$$

Jak widać, z (5.15) i (5.16) wyznaczane są chwile zakończenia montażu operacji odpowiednio po lewej i po prawej stronie osi linii.

Oznaczmy przez $\Omega_{\eta,n}$ podzbiór operacji, które są bezpośrednimi poprzednikami operacji ω_n. Mamy zatem:

$$\bigwedge_{\nu} (\gamma_{\nu,n}=1) \Longrightarrow (\omega_\nu \in \Omega_{\eta,n}) \tag{5.17}$$

W stanie $P^{\eta,l}$ dla każdej operacji ω_n, $(p_n^{\eta,l} \neq 0)$, można wyznaczyć chwilę $T_{\eta,n}^{\eta,l}$ zakończenia jej bezpośrednich poprzedników. Zatem:

$$T_{\eta,n}^{\eta,l} = \max_{\omega_\nu \in \Omega_{\eta,n}} \left| p_\nu^{\eta,l} \right| \tag{5.18}$$

Z przyjętego kryterium optymalizacji (5.3) oraz definicji funkcji wartości stanu (5.11) wynika, że globalnie optymalny stan końcowy P^o wyznaczamy z warunku:

$$\left[\min_{1 \le l \le L_N} (v_1^{N,l} + v_2^{N,l}) = v_1^{N,l^o} + v_2^{N,l^o}\right] \Rightarrow (P^{N,l^o} = P^o) \tag{5.19}$$

Ze stanu P^o wyznaczamy optymalny przydział operacji do stanowisk pracy zgodnie z (5.10) i (5.6). W trakcie obliczeń wygenerowany stan końcowy $P^{N,l}$ jest porównywany ze stanem aktualnie najlepszym P^a, a lepszy z nich jest zapamiętywany do dalszych obliczeń. Ostatni ze stanów aktualnie najlepszych jest stanem globalnie optymalnym.

G e n e r o w a n i e s t a n ó w

Oznaczmy przez $P^{\eta-1,\lambda}$ wybrany (za pomocą dowolnej reguły) stan aktywny. Załóżmy, że reguła podziału jest jednokrokowa i zupełna. Generowanie stanów polega na uzupełnieniu stanu $P^{\eta-1,\lambda}$ o dopuszczalną operację ω_n. W rezultacie otrzymamy dwa stany: P^1 - przydzielając operację ω_n na lewą stronę linii oraz P^2 - przydzielając operację ω_n na prewę stronę linii. Stany P^1 i P^2 otrzymają odpowiednie indeksy (η,l) stanów aktywnych, jeśli okażę się perspektywiczne.

Procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\nu (p_n^{\eta-1,\lambda}=0) \wedge \left[(\gamma_{\nu,n}=1) \Longrightarrow (p_\nu^{\eta-1,\lambda} \neq 0)\right] \Longrightarrow \left[(P^1 = P^{\eta-1,\lambda} + \Delta P^1) \wedge (P^2 = P^{\eta-1,\lambda} + \Delta P^2)\right] \tag{5.20}$$

Wektory ΔP^1 i ΔP^2 są określone następująco:

$$\Delta p_i^1 = \begin{cases} t_n^1, & \text{dla } i=n \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{5.21}$$

oraz

$$\Delta p_i^2 = \begin{cases} t_n^2, & \text{dla } i=n \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{5.22}$$

Chwilę t_n^i wyznaczamy z następujących wzorów:

- Jeżeli spełniony jest warunek:

$$T_2^{\eta-1,\lambda} \le T_1^{\eta-1,\lambda} \tag{5.23}$$

to wówczas

$$t_n^1 = \begin{cases} -(T_1^{\eta-1,\lambda} + \vartheta_n), \text{ jeżeli } K_1^{\eta-1,\lambda} \cdot c - T_1^{\eta-1,\lambda} \geq \vartheta_n \\ -(K_1^{\eta-1,\lambda} \cdot c + \vartheta_n), \text{ w przypadku przeciwnym.} \end{cases} \tag{5.24}$$

- Jeżeli warunek (5.23) nie jest spełniony, otrzymujemy:

$$t_n^1 = \begin{cases} -\left[\max(T_1^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda}) + \vartheta_n\right], \text{ jeżeli } \left[\frac{1}{c}\max(T_1^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda})\right]^+ = \\ = \left[\frac{1}{c}\left\{\max(T_1^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda}) + \vartheta_n\right\}\right]^+ \\ -\left\{\left[\frac{1}{c}\max(T_1^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda})\right]^+ \cdot c + \vartheta_n\right\}, \text{ w przypadku przeciwnym} \end{cases} \tag{5.25}$$

Analogicznie otrzymamy dla chwili t_n^2:

- Jeżeli

$$T_1^{\eta-1,\lambda} \leq T_2^{\eta-1,\lambda} \tag{5.26}$$

to

$$t_n^2 = \begin{cases} T_2^{\eta-1,\lambda} + \vartheta_n, \text{ jeśli } K_2^{\eta-1,\lambda} \cdot c - T_2^{\eta-1,\lambda} \geq \vartheta_n \\ K_2^{\eta-1,\lambda} \cdot c + \vartheta_n, \text{ w przypadku przeciwnym.} \end{cases} \tag{5.27}$$

- Jeżeli warunek (5.26) nie jest epełniony, to otrzymamy:

$$t_n^2 = \begin{cases} \max(T_2^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda}) + \vartheta_n, \text{jeśli } \left[\frac{1}{c}\max(T_2^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda})\right]^+ = \\ = \left[\frac{1}{c}\left\{\max(T_2^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda}) + \vartheta_n\right\}\right]^+ \\ \left[\frac{1}{c}\max(T_2^{\eta-1,\lambda}, T_{\eta,n}^{\eta-1,\lambda})\right]^+ \cdot c + \vartheta_n, \text{ w przypadku przeciwnym.} \end{cases} \tag{5.28}$$

A zatem wychodząc ze stanu $P^{0,1}$ można za pomocą przedstawionej procedury wygenerować wszystkie stany. Stan P^1 jest dopuszczalny, jeśli $|t_n^1| \leq K \cdot c$; analogicznie dla stanu P^2 sprawdzamy warunek dopuszczalności $t_n^2 \leq K \cdot c$.

W trakcie generowania stanów P^1 i P^2 można obliczać współrzędne wektorów V^1 i V^2. A zatem zamiast definicji 5.2 stosujemy wzór rekurencyjny:

$$V^{\eta,1} = V^{\eta-1,\lambda} + \Delta V \tag{5.29}$$

Dla stanu P^1 otrzymujemy:

$$\Delta v_1^1 = \begin{cases} 1, \text{ jeśli } \left[\frac{1}{c}\left|t_n^1\right|\right]^+ > K_1^{\eta-1,\lambda} \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.30a}$$

oraz

$$\Delta v_2^1 = 0 \tag{5.30b}$$

Analogicznie dla stanu P^2 wyznaczamy:

$$\Delta v_1^2 = 0 \tag{5.30c}$$

oraz

$$\Delta v_2^2 = \begin{cases} 1, \text{ jeśli } \left[\frac{1}{c}\, t_n^2\right]^+ > K_2^{\eta-1,\lambda} \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.30d}$$

przy tym

$$T_1^{\eta,1} = \left|t_n^1\right| \tag{5.31a}$$

$$T_2^{\eta,1} = t_n^2 \tag{5.31b}$$

Tak więc obliczenia wartości generowanych stanów upraszczają się.

5.2.3. Eliminowanie stanów nieperspektywicznych

Do eliminacji stanów nieperspektywicznych wykorzystamy reguły sondowania, dominacji oraz wyczerpywania. Reguła sondowania ma postać twierdzenia.

TWIERDZENIE 5.1

Stan $P^{\eta,1}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$v_1^a + v_2^a \leq v_1^{\eta,l} + v_2^{\eta,l} - 2 + \left[\frac{1}{c}\left\{T_1^{\eta,l} + T_2^{\eta,l} + \right.\right.$$

$$\left.\left. - (K_1^{\eta,l} + K_2^{\eta,l} - 2) \cdot c + \sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_n \right\}\right]^+ \qquad (5.32)$$

D o w ó d tego twierdzenia polega na wykazaniu, że prawa strona nierówności (5.32) jest dolnym ograniczeniem liczby stanowisk pracy dla stanu P^{N,l^o} (otrzymanego z $P^{\eta,l}$). Jeżeli w stanie $P^{\eta,l}$ po lewej stronie linii montaż kończy się w chwili $T_1^{\eta,l}$ na stacji $K_1^{\eta,l}$, to wykorzystano pełnych $v_1^{\eta,l}-1$ stanowisk pracy. Czas wykonania operacji na ostatnim stanowisku pracy wynosi:

$$T_1^{\eta,l} - (K_1^{\eta,l} - 1) \cdot c.$$

Analogiczna sytuacja wyatępuje po prawej stronie linii. Suma czasów operacji wykonywanych w stanie $P^{\eta,l}$ na ostatnich stanowiskach pracy po obu stronach linii wynosi:

$$T_1^{\eta,l} + T_2^{\eta,l} - (K_1^{\eta,l} + K_2^{\eta,l} - 2) \cdot c.$$

Dodając do tej sumy czasy operacji nie przydzielonych w stanie $P^{\eta,l}$, czyli:

$$\sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_n$$

otrzymujemy łączny nakład czasu pracy na stacjach:

$$K_1^{\eta,l}, \ldots, K$$

po lewej stronie linii

oraz

$$K_2^{\eta,l}, \ldots, K$$

po prawej stronie linii.

Na tej podstawie (stosując formułę $[\cdot]^+$) otrzymujemy minimalną liczbę stanowisk pracy. A zatem prawa strona nierówności (5.32) jest oszacowaniem najmniejszej liczby stanowisk pracy, jaką można osiągnąć ze stanu $P^{\eta,l}$.

Reguła dominacji stanów ma postać twierdzenia.

TWIERDZENIE 5.2

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{1 \leq j \leq 2} \bigwedge_n (\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (\omega_n \notin \Omega^{\eta,l_1}) \wedge (T_j^{\eta,l_1} \leq T_j^{\eta,l_2}) \wedge$$

$$\wedge (T_{\vartheta,n}^{\eta,l_1} \leq T_{\vartheta,n}^{\eta,l_2}) \wedge \left[(v_j^{\eta,l_1} < v_j^{\eta,l_2}) \vee (K_j^{\eta,l_1} = K_j^{\eta,l_2}) \wedge \right.$$

$$\left. \wedge (v_j^{\eta,l_1} \leq v_j^{\eta,l_2})\right] \qquad (5.33)$$

D o w ó d tego twierdzenia polega na wykazaniu, że trajektoria lokalnie optymalna $P^{\eta,l_2}, \ldots, P^{N,l_2^o}$ może być również zrealizowana od stanu P^{η,l_1}, skąd otrzymamy $P^{\eta,l_1}, \ldots, P^{N,\lambda}$. Oznacza to, że współrzędne p_n^{N,l_2^o} dla operacji $\omega_n \notin \Omega^{\eta,l_2}$ są takie same jak $p_n^{N,\lambda}$, tzn:

$$\bigwedge_n (\omega_n \notin \Omega^{\eta,l_1}) \wedge (\omega_n \notin \Omega^{\eta,l_2}) \Rightarrow (p_n^{N,l_2^o} = p_n^{N,\lambda}) \qquad (5.34)$$

Możliwość realizacji trajektorii $P^{\eta,l_2}, \ldots, P^{N,l_2^o}$ również od stanu P^{η,l_1} (dająca $P^{\eta,l_1}, \ldots, P^{N,\lambda}$) wynika bezpośrednio z warunku (5.33). Ponieważ $\Omega^{\eta,l_1} = \Omega^{\eta,l_2}$, zatem w obydwu stanach P^{η,l_1} i P^{η,l_2} pozostały do wykonania te same operacje. W stanie P^{η,l_1} nie później niż w P^{η,l_2} zostały zakończone wszystkie bezpośrednie poprzedniki operacji $\omega_n \notin \Omega^{\eta,l_1}$, a ponadto wszystkie operacje po lewej i po prawej stronie linii. W tych warunkach jakikolwiek przydział operacji $\omega_n \notin \Omega^{\eta,l2}$ (a zatem i lokalnie optymalny) od stanu P^{η,l_2} jest również dopuszczalny od stanu P^{η,l_1}.

W ogólnym przypadku etan $P^{N,\lambda}$ jest różny od P^{N,l_1^o}, a zatem:

$$v_1^{N,l_1^o} + v_2^{N,l_1^o} \leq v_1^{N,\lambda} + v_2^{N,\lambda} \qquad (5.35)$$

Dla trajektorii $P^{\eta,l_2}, \ldots, P^{N,l_2^o}$ możemy napisać:

$$v_1^{N,l_2^o} = v_1^{\eta,l_2} + \Delta v_1^{\eta,l_2} \tag{5.36a}$$

oraz

$$v_2^{N,l_2^o} = v_2^{\eta,l_2} + \Delta v_2^{\eta,l_2} \tag{5.36b}$$

Z kolei dla trajektorii $P^{\eta,l_1},\ldots,P^{N,\lambda}$ otrzymamy:

$$v_1^{N,\lambda} = v_1^{\eta,l_1} + \Delta v_1^{\eta,l_1} \tag{5.37a}$$

oraz

$$v_2^{N,\lambda} = v_2^{\eta,l_1} + \Delta v_2^{\eta,l_1} \tag{5.37b}$$

Zauważmy, że:

$$\Delta v_1^{\eta,l_1} = \begin{cases} \Delta v_1^{\eta,l_2}, & \text{jeśli } K_1^{\eta,l_1} = K_1^{\eta,l_2} \\ \Delta v_1^{\eta,l_2}+1, & \text{jeśli } K_1^{\eta,l_1} < K_1^{\eta,l_2} \end{cases} \tag{5.38a}$$

oraz

$$\Delta v_2^{\eta,l_1} = \begin{cases} \Delta v_2^{\eta,l_2}, & \text{jeśli } K_2^{\eta,l_1} = K_2^{\eta,l_2} \\ \Delta v_2^{\eta,l_2}+1, & \text{jeśli } K_2^{\eta,l_1} < K_2^{\eta,l_2} \end{cases} \tag{5.38b}$$

Ponadto w (5.33) żądamy, by:

$$v_1^{\eta,l_1} \le v_1^{\eta,l_2} \quad \text{dla} \quad K_1^{\eta,l_1} = K_1^{\eta,l_2} \tag{5.39a}$$

lub:

$$v_1^{\eta,l_1} < v_1^{\eta,l_2} \quad \text{dla} \quad K_1^{\eta,l_1} < K_1^{\eta,l_2} \tag{5.39b}$$

oraz

$$v_2^{\eta,l_1} \le v_2^{\eta,l_2} \quad \text{dla} \quad K_2^{\eta,l_1} = K_2^{\eta,l_2} \tag{5.40a}$$

lub:

$$v_2^{\eta,l_1} \le v_2^{\eta,l_2} \quad \text{dla} \quad K_2^{\eta,l_1} < K_2^{\eta,l_2} \tag{5.40b}$$

Uwzględniając (5.38a) i (5.39a) w (5.37a) otrzymamy:

- dla $K_1^{\eta,l_1} = K_1^{\eta,l_2}$

$$v_1^{N,\lambda} = v_1^{\eta,l_1} + \Delta v_1^{\eta,l_1} \le v_1^{\eta,l_2} + \Delta v_1^{\eta,l_2} = v_1^{N,l_2^o} \tag{5.41a}$$

czyli

$$v_1^{N,\lambda} \le v_1^{N,l_2^o} \tag{5.41b}$$

- dla $K_1^{\eta,l_1} < K_1^{\eta,l_2}$

$$v_1^{N,\lambda} = v_1^{\eta,l_1} + \Delta v_1^{\eta,l_1} < v_1^{\eta,l_2} + \Delta v_1^{\eta,l_2} + 1 = v_1^{N,l_2^o} + 1 \tag{5.42a}$$

czyli

$$v_1^{N,\lambda} \le v_1^{N,l_2^o} \tag{5.42b}$$

W analogiczny sposób można uwzględnić (5.38b) i (5.39b) w (5.37b), otrzymując w rezultacie:

$$v_2^{N,\lambda} \le v_2^{N,l_2^o} \tag{5.43}$$

Podstawiając (5.41b) i (5.42b) oraz (5.43) do (5.35) otrzymamy ostatecznie:

$$v_1^{N,l_1^o} + v_2^{N,l_1^o} \le v_1^{N,l_2^o} + v_2^{N,l_2^o} \tag{5.44}$$

Z (5.44) wynika, że stan P^{η,l_1} dominuje nad stanem P^{η,l_2}. □

W przedstawionej procedurze generowania stanów (5.20) przyjęto, że operacja ω_n jest realizowana w stanie $P^{\eta-1,\lambda}$ w najwcześniejszym dopuszczalnym przedziale czasu. Można wykazać, że celowe opóźnienie realizacji tej operacji daje stan nieperspektywiczny.

Załóżmy, że dany jest stan $P^{\eta-1,\lambda}$ oraz operacja ω_n, spełniająca warunek (5.20).

LEMAT

Jeżeli operację ω_n można w stanie $P^{\eta-1,\lambda}$ przydzielić najwcześniej do stacji $k_1^{\eta-1,\lambda}$ lub $k_2^{\eta-1,\lambda}$, to przydzielenie jej do stacji $\mathcal{K}_1$ lub $\mathcal{K}_2$, gdzie:

$$\mathcal{K}_1 > k_1^{\eta-1,\lambda} \tag{5.44a}$$

lub

$$\mathcal{K}_2 > k_2^{\eta-1,\lambda} \tag{5.44b}$$

daje stan nieperspektywiczny.

D o w ó d: Oznaczmy przez P^{η,l_1} stan otrzymany z przydzielenia operacji ω_n do stacji $k_1^{\eta-1,\lambda}$ (lub $k_2^{\eta-1,\lambda}$), natomiast przez P^{η,l_2} stan otrzymany z przydzielenia operacji ω_n do stacji $\mathcal{K}_1$ (lub $\mathcal{K}_2$). Zauważmy, że:

$$\Omega^{\eta,l_1} = \Omega^{\eta,l_2} \tag{5.45a}$$

$$T_j^{\eta,l_1} \leq T_j^{\eta,l_2} \qquad j=1,2 \tag{5.45b}$$

$$T_{\vartheta,n}^{\eta,l_1} \leq T_{\vartheta,n}^{\eta,l_2} \tag{5.45c}$$

a ponadto:

$$v_j^{\eta,l_2} = \begin{cases} v_j^{\eta,l_1}, & \text{jeśli } K_j^{\eta,l_1} = K_j^{\eta,l_2} \\ v_1^{\eta,l_1} + 1, & \text{jeśli } K_j^{\eta,l_1} < K_j^{\eta,l_2} \end{cases} \tag{5.45d}$$

Z (5.45) wynika, że stany P^{η,l_1} i P^{η,l_2} spełniają warunek dominacji (5.33), przy czym stan P^{η,l_2} jest nieperspektywiczny. □

Reguła wyczerpywania ma postać twierdzenia.

TWIERDZENIE 5.3

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeśli spełnia warunek:

$$2K \cdot c - T_1^{\eta,l} - T_2^{\eta,l} < \sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_n \tag{5.46}$$

D o w ó d tego twierdzenia wynika z faktu, że w stanie $P^{\eta,l}$ suma czasów operacji, które należy wykonać, jest większa od czasu, jakim dysponują monterzy na stanowiskach pracy po lewej i po prawej stronie linii. A zatem, przy założeniu co najwyżej dwóch stanowisk pracy na jednej stacji, kompletny montaż obiektu nie jest możliwy.

5.3. RELACJA WYKLUCZANIA OPERACJI

W podstawowym modelu AO w liniach z równoległymi stanowiskami pracy występowała jedynie relacja kolejności operacji. Cechą charakterystyczną procesu montażu na równoległych stanowiskach pracy jest występowanie ograniczeń wykluczających wykonanie dwóch operacji ω_μ i ω_n w tym samym czasie - po różnych stronach linii. Model AO w linii z równoległymi stanowiskami pracy i relacją wykluczania operacji stanowi uogólnienie przedstawionego w punkcie 5.2 modelu podstawowego.

5.3.1. Model matematyczny

Rozważmy linię montażową i proces montażu przyjmując wszystkie założenia z punktu 5.2.1. Ponadto założymy, że dana jest relacja wykluczania operacji, zapisana macierzą:

$$\mathcal{B} = \left[\beta_{\mu,n}\right]_{\substack{\mu=1,\ldots,N \\ n=1,\ldots,N}} \tag{5.47}$$

Elementy tej macierzy określamy następująco:

$$\beta_{\mu,n} = \begin{cases} 1, \text{ jeśli operacje } \omega_\mu \text{ i } \omega_n \text{ nie mogą być wykonywane w tym samym czasie} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{5.47a}$$

Jako cel optymalizacji AO z relację wykluczania przyjmiemy minimalizację liczby stanowisk pracy.

Wprowadzając kodowaną zmienną t_n (tak jak w podstawowym modelu AO w liniach równoległych) otrzymamy model matematyczny rozważanego problemu w postaci:

- kryterium (5.3), (5.7);
- ograniczeń (5.8).

Ponadto w modelu tym wystąpią ograniczenia wykluczania operacji:

$$\bigwedge_{\mu}\bigwedge_{n}\ (\beta_{\mu,n}=1)\Rightarrow\left[(|t_\mu|<|t_n|-\vartheta_n)\vee(|t_n|<|t_\mu|-\vartheta_\mu)\right] \qquad (5.48)$$

Operacje ω_μ i ω_n mogą być wykonane po różnych stronach lub po tej samej stronie linii montażowej.

5.3.2. Algorytm

Do rozwiązania sformułowanego problemu AO w linii równoległej z relację wykluczania operacji wykorzystamy algorytm programowania wieloetapowego, przyjmując określenia: $P^{\gamma,1}$, k_n, $V^{\gamma,1}$, $K_j^{\gamma,1}$, $T_j^{\gamma,1}$ oraz $T_{\vartheta,n}^{\gamma,1}$ - wprowadzone w punkcie 5.2.2. Ponadto dla operacji ω_n takiej, która spełnia warunek $\omega_n \notin \Omega^{\gamma,1}$, określimy podzbiór: $\Omega_{\beta,n}^{\gamma,1}$, gdzie:

$$\bigwedge_{\mu}\ (p_\mu^{\gamma,1}\neq 0)\wedge(\beta_{\mu,n}=1)\Rightarrow(\omega_\mu\in\Omega_{\beta,n}^{\gamma,1}) \qquad (5.49)$$

A zatem $\Omega_{\beta,n}^{\gamma,1}$ jest podzbiorem operacji ω_μ, które wykluczają się z operację ω_n i zostały w stanie $P^{\gamma,1}$ przydzielone do realizacji. Z uwagi na ograniczenia wykluczania operacji, operacja ω_n może rozpocząć się w stanie $P^{\gamma,1}$ najwcześniej w chwili $T_{\beta,n}^{\gamma,1}$, gdzie:

$$T_{\beta,n}^{\gamma,1}=\max_{\omega_\mu\in\Omega_{\beta,n}^{\gamma,1}}|p_\mu^{\gamma,1}| \qquad (5.50)$$

Optymalny stan końcowy P^o wynika z warunku (5.19).

G e n e r o w a n i e s t a n ó w

Analogicznie jak w algorytmie dla podstawowego problemu AO w linii równoległej przyjmiemy, że z wybranego stanu $P^{\gamma-1,\lambda}$ za pomocą operacji ω_n generowane są stany P^1 i P^2. Stany P^j, j=1,2 są dopuszczalne, jeżeli $t_n^j \leq K\ .\ c$.

Relacja $\mathcal{J}\!\!\mathcal{B}$ jest traktowana lokalnie w stanie $P^{\gamma-1,\lambda}$ jako relacja Γ. A zatem operacja ω_n może być przydzielona w stanie $P^{\gamma-1,\lambda}$ nie wcześniej niż w chwili $T_{\beta,n}^{\gamma-1,\lambda}$.

Procedura generowania stanów ma postać:

$$\bigwedge_{n}\bigwedge_{\vartheta}(p_n^{\gamma-1,\lambda}=0)\wedge\left[(\gamma_{\vartheta,n}=1)\Rightarrow(p_\vartheta^{\gamma-1,\lambda}\neq 0)\right]\Rightarrow\left[(P^1=P^{\gamma-1,\lambda}+\Delta P^1)\wedge\right.$$

$$\left.\wedge(P^2=P^{\gamma-1,\lambda}+\Delta P^2)\right] \qquad (5.51)$$

Elementy wektorów ΔP^j, j=1,2 określamy tak jak w (5.21) i (5.22).

Chwilę t_n^1 wyznaczamy wyróżniając dwa przypadki:

- Jeżeli spełniony jest warunek (5.23), to t_n^1 otrzymujemy ze wzoru (5.24)
- Jeżeli warunek (5.23) nie jest spełniony, to t_n^1 wyznaczamy ze wzoru:

$$t_n^1=\begin{cases}-\left[\max(T_1^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})+\vartheta_n\right],\ \text{jeśli}\ \left[\frac{1}{c}\max(T_1^{\gamma-1,\lambda},\right.\\ \left.T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})\right]^+=\left[\frac{1}{c}\left\{\max(T_1^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})+\vartheta_n\right\}\right]^+\\ -\left\{\left[\frac{1}{c}\max(T_1^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})\right]^+\ .\ c+\vartheta_n\right\},\ \text{w przypadku}\\ \text{przeciwnym.}\end{cases} \qquad (5.52)$$

W analogiczny sposób wyznaczymy chwilę t_n^2, wyróżniając dwa przypadki:

- Jeżeli spełniony jest warunek (5.26), to t_n^2 otrzymujemy ze wzoru (5.27).
- Jeżeli warunek (5.26) nie jest spełniony, to t_n^2 wyznaczamy ze wzoru:

$$t_n^2=\begin{cases}\max(T_2^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})+\vartheta_n,\ \text{jeśli}\ \left[\frac{1}{c}\max(T_2^{\gamma-1,\lambda},\right.\\ \left.T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})\right]^+=\left[\frac{1}{c}\left\{\max(T_2^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})+\vartheta_n\right\}\right]^+\\ \left[\frac{1}{c}\max(T_2^{\gamma-1,\lambda},\ T_{\beta,n}^{\gamma-1,\lambda},\ T_{\vartheta,n}^{\gamma-1,\lambda})\right]^+\ .\ c+\vartheta_n,\ \text{w przypadku}\\ \text{przeciwnym.}\end{cases} \qquad (5.53)$$

Równocześnie z generowaniem stanów wyznaczamy $V^{\gamma,1}$, $T_j^{\gamma,1}$ korzystając z formuł (5.29), (5.30) i (5.31).

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Do eliminacji stanów nieperspektywicznych w rozważanym problemie wykorzystujemy reguły wyczerpywania, sondowania oraz dominacji. Reguła wyczerpywania ma postać (5.46), a reguła sondowania - (5.32).

Reguła dominacji jest oparta na twierdzeniu.

TWIERDZENIE 5.4

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{1\le j\le 2}\ \bigwedge_{n}\ (\Omega^{\eta,l_1}=\Omega^{\eta,l_2})\wedge(\omega_n\in\Omega^{\eta,l_1})\wedge(T_j^{\eta,l_1}\le T_j^{\eta,l_2})\wedge$$

$$\wedge(T_{\beta,n}^{\eta,l_1}\le T_{\beta,n}^{\eta,l_2})\wedge(T_{\varphi,n}^{\eta,l_1}\le T_{\varphi,n}^{\eta,l_2})\wedge\left[(v_j^{\eta,l_1}<v_j^{\eta,l_2})\vee K_j^{\eta,l_1}=\right.$$

$$\left.=k_j^{\eta,l_2})\wedge(v_j^{\eta,l_1}\le v_j^{\eta,l_2})\right] \qquad (5.54)$$

D o w ó d tego twierdzenia można przeprowadzić analogicznie jak dla twierdzenia 5.2.

W przedstawionym algorytmie relację $\mathcal{J}\beta$ traktowano lokalnie w stanie $P^{\eta-1,\lambda}$ jako relację Γ. A zatem nie analizowano możliwości przydzielenia operacji ω_n przed chwilą $T_{\beta,n}^{\eta-1,\lambda}$. Gdyby taka możliwość istniała, to stan wygenerowany za pomocą procedury (5.51) byłby nieperspektywiczny. Ponieważ w algorytmie programowania wieloetapowego generowane są wszystkie trajektorie (z pominięciem tych, których nieperspektywiczność udowodniono), zatem odpowiedni stan perspektywiczny zostanie również wygenerowany. Stąd nieperspektywiczny stan otrzymany z (5.51) jest eliminowany za pomocą reguły dominacji.

5.4. RELACJA SYNCHRONIZACJI OPERACJI

W procesie montażu na równoległych stanowiskach pracy występują przypadki synchronicznego wykonywania pewnych operacji po obu stronach linii. Zatem rozważymy obecnie model AO w równoległej linii montażowej z uwzględnieniem relacji synchronizacji. Model taki stanowi uogólnienie przedstawionego w punkcie 5.2 modelu podstawowego.

5.4.1. Model matematyczny

Rozważmy linię montażową i proces montażu przyjmując wszystkie założenia wprowadzone w punkcie 5.2.1. Dodatkowo założymy, że dana jest relacja synchronizacji operacji, zapisana macierzą:

$$\mathcal{A}=\left[\alpha_{\mu,n}\right]_{\substack{\mu=1,\dots,N\\ n=1,\dots,N}} \qquad (5.55)$$

Elementy tej macierzy określamy następująco:

$$\alpha_{\mu,n}=\begin{cases}1, \text{ jeśli operacje } \omega_\mu \text{ i } \omega_n \text{ muszą być wykonane równocześnie}\\ 0, \text{ w przypadku przeciwnym.}\end{cases} \qquad (5.55a)$$

Przyjmiemy, że:

$$\nu_\mu^\eta=\nu_n^\eta \qquad (5.56)$$

oraz że operacje ω_μ i ω_n rozpoczynają się w tej samej chwili i są wykonane po różnych stronach linii.

Jako cel optymalizacji AO z relacją synchronizacji operacji przyjmiemy minimalizację liczby stanowisk pracy.

Analogicznie jak w modelu podstawowym AO w liniach z równoległymi stanowiskami pracy, przyjmiemy kodowaną zmienną t_n. Model matematyczny rozważanego problemu przyjmie postać:

- kryterium (5.3), (5.7),
- ograniczeń (5.8).

Ponadto w modelu tym wystąpią ograniczenia synchronizacji operacji w postaci:

$$\bigwedge_{n}\bigwedge_{\mu}(\alpha_{\mu,n}=1)\Rightarrow(t_\mu=t_n) \qquad (5.57)$$

A zatem relacja synchronizacji dotyczy dwóch operacji ω_μ i ω_n $(\alpha_{\mu,n}=1)$, które są wykonywane po przeciwnych stronach linii.

5.4.2. Algorytm

Sformułowany problem rozwiążemy za pomocą algorytmu programowania wieloetapowego, przyjmując określenia: $P^{\eta,1}$, k_n, $V^{\eta,1}$, $K_j^{\eta,1}$, $T_j^{\eta,1}$ oraz $T_{\varphi,n}^{\eta,1}$ - wprowadzone w punkcie 5.2.2. W algorytmie tym uwzględnimy ograniczenia synchronizacji operacji (5.57).

G e n e r o w a n i e s t a n ó w

Analogicznie jak w modelu podstawowym, ze stanu $P^{\eta-1,\lambda}$ można otrzymać dwa stany P^1 i P^2 - wykorzystując operację ω_n. Stany P^j, $j=1,2$ są dopuszczalne, jeżeli $t_n^j\le K$. c.

Wyróżnimy dwie procedury generowania stanów:

1) dla operacji ω_n, która spełnia warunek:

$$\bigwedge_{1 \le \mu \le N} \alpha_{\mu,n} = 0 \tag{5.58}$$

2) dla operacji ω_n, która nie spełnia warunku (5.58).

Pierwsza procedura generowanie stanów ma postać:

$$\bigwedge_n \bigwedge_\nu \bigwedge_{1 \le \mu \le N} (p_n^{\eta-1,\lambda}=0) \wedge (\alpha_{\mu,n}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p_\nu^{\eta-1,\lambda}\neq 0)\right] \Rightarrow$$

$$= (P^1 = P^{\eta-1,\lambda} + \Delta P^1) \wedge (P^2 = P^{\eta-1,\lambda} + \Delta P^2) \tag{5.59}$$

Wektory ΔP^j, j=1,2 określone są przez (5.21) i (5.22).

Chwilę t_n^1 wyznaczamy z następujęcych wzorów:

- Jeżeli spełniony jest warunek (5.23), to t_n^1 jest określone przez (5.24).
- Jeżeli warunek (5.23) nie jest spełniony, to t_n^2 wyznaczamy z (5.25).

W analogiczny sposób, biorąc pod uwagę warunek (5.26), wyznaczamy chwilę t_n^2 wykorzystując wzory, odpowiednio (5.27) i (5.28).

A zatem pierwsza procedura generowenia stanów (5.59) daje identyczne rezultaty jak procedura (5.20) generowania etanów w modelu podstawowym. Stąd też wykorzystujemy formuły (5.30) i (5.31) dla wyznaczania $V^{\eta,1}$ i $T_j^{\eta,1}$, j=1,2.

Druga procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\nu \bigvee_{n<\mu} (p_n^{\eta-1,\lambda}=0) \wedge (\alpha_{\mu,n}=1) \wedge (p_\mu^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow\right.$$

$$\left.\Rightarrow (p_\nu^{\eta-1,\lambda}\neq 0)\right] \wedge \left[(\vartheta_{\nu,\mu}=1) \Rightarrow (p_\nu^{\eta-1,\lambda}\neq 0)\right] \Rightarrow (P = P^{\eta-1,\lambda} + \Delta P) \tag{5.60}$$

Wektor ΔP określamy następujęco:

$$\Delta p_i = \begin{cases} t_n, & \text{dla } i=n \\ t_\mu, & \text{dla } i=\mu \\ 0, & \text{w pozostałych przypadkach.} \end{cases} \tag{5.61}$$

Ponieważ operacje ω_μ i ω_n są wykonywane równocześnie po przeciwnych stronach linii, zatem:

$$t_\mu = -t_n \tag{5.62}$$

Chwilę t_n wyznaczamy ze wzoru:

$$t_n = \begin{cases} \max(T_1^{\eta-1,\lambda}, T_2^{\eta-1,\lambda}) + \vartheta_n, \text{ jeśli } \left[\frac{1}{c}\max(T_1^{\eta-1,\lambda}, T_2^{\eta-1,\lambda})\right]^+ = \\ \qquad = \left[\frac{1}{c}\left\{\max(T_1^{\eta-1,\lambda}, T_2^{\eta-1,\lambda}) + \vartheta_n\right\}\right]^+ \\ \left[\frac{1}{c}\max(T_1^{\eta-1,\lambda}, T_2^{\eta-1,\lambda})\right]^+ \cdot c + \vartheta_n, \text{ w przypadku przeciwnym} \end{cases} \tag{5.63}$$

A zatem operacja ω_n została przydzielona na prawę stronę linii.

Zgodnie z procedurą (5.60) generowany jest tylko jeden stan P, ponieważ stany P^1 i P^2 byłyby dla dalszych obliczeń identyczne. Ponadto procedura (5.60) jest dwukrokowa, tzn. ze stanu $P^{\eta-1,\lambda}$ otrzymujemy po przydzieleniu operacji ω_μ i ω_n - stan $P^{\eta+1,1}$.

Równocześnie z generowaniem stanu wyznaczamy $V^{\eta+1,1}$ na podstawie wzoru:

$$V^{\eta+1,1} = V^{\eta-1,\lambda} + \Delta V \tag{5.64}$$

przy tym

$$\Delta v_1 = \begin{cases} 1, \text{ jeśli } \left[\frac{1}{c}\,|t_\mu|\right]^+ > K_1^{\eta-1,\lambda} \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.65a}$$

oraz

$$\Delta v_2 = \begin{cases} 1, \text{ jeśli } \left[\frac{1}{c}\,t_n\right]^+ > K_2^{\eta-1,\lambda} \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{5.65b}$$

ponadto

$$T_1^{\eta+1,1} = T_2^{\eta+1,1} = t_n \tag{5.66}$$

A zatem wzory (5.65) i (5.66) upraszczają obliczenia.

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

W rozważanym modelu do eliminacji stanów nieperspektywicznych wykorzystywane są reguły sondowania (5.32), dominacji (5.33) oraz wyczerpywanie

(5.46). W oparciu o regułę dominacji można wykazać, że wystarczy, by procedura generowania stanów (5.60) pozwalała wyznaczyć jeden stan $P^{\eta+1,l}$ dla generacji ω_μ i ω_n, gdy $\alpha_{\mu,n} = 1$. Załóżmy, że stan $P^{\eta+1,l_1}$ otrzymujemy z $P^{\eta-1,\lambda}$ przydzielając ω_μ na lewą, a ω_n na prawą stronę linii. Z kolei $P^{\eta+1,l_2}$ otrzymujemy przydzielając ω_μ na prawą, a ω_n na lewą stronę linii ($\alpha_{\mu,n}=1$).

LEMAT 5.2

Stan $P^{\eta+1,l_1}$ dominuje nad stanem $P^{\eta+1,l_2}$.

D o w ó d: Zauważmy, że:

$$\Omega^{\eta+1,l_1} = \Omega^{\eta+1,l_2} \tag{5.67}$$

Ponadto:

$$\bigwedge_{1\le j\le 2} T_j^{\eta+1,l_1} = T_j^{\eta+1,l_2} \tag{5.68}$$

oraz

$$\bigwedge_{1\le j\le 2} \bigwedge_n (\omega_n \notin \Omega^{\eta+1,l_1}) \Rightarrow (T_{\eta,n}^{\eta+1,l_1} \le T_j^{\eta+1,l_1}) \tag{5.69a}$$

$$\bigwedge_{1\le j\le 2} \bigwedge_n (\omega_n \notin \Omega^{\eta+1,l_2}) \Rightarrow (T_{\eta,n}^{\eta+1,l_2} \le T_j^{\eta+1,l_2}) \tag{5.69b}$$

Z (5.68) wynika, że:

$$\bigwedge_{1\le j\le 2} K_j^{\eta+1,\lambda} = K_j^{\eta+1,\lambda} \tag{5.70}$$

Ponadto z (5.62) oraz (5.65) wnioskujemy, że:

$$\bigwedge_{1\le j\le 2} v_j^{\eta+1,l_1} = v_j^{\eta+1,l_2} \tag{5.71}$$

Z uwagi na (5.69), chwile $T_{\eta,n}^{\eta+1,l}$ nie są istotne w regule dominacji stanów (5.33). Ponadto z równości: (5.67), 5.68), (5.70) i (5.71) wnioskujemy, że stan $P^{\eta+1,l_1}$ dominuje nad stanem $P^{\eta+1,l_2}$.

Ponieważ reguła dominacji (5.33) jest spełniona przez same równości, stany $P^{\eta+1,l_1}$ i $P^{\eta+1,l_2}$ zostały wcześniej nazwane identycznymi.

W procedurze generowania stanów (5.60), operacje ω_μ i ω_n zostają przydzielone do najwcześniejszego dopuszczalnego przedziału czasu. Przydzielanie tych operacji do późniejszych przedziałów czasu daje stany nieperspektywiczne. Wynika to z lematu 5.1.

6. ALOKACJA OPERACJI I ZASOBÓW

6.1. WSTĘP

W rozważanym dotąd procesie montażu uwzględniono jedynie zasób podstawowy, który stanowią stacje montażowe (stanowiska pracy). Na liniach montażowych istotne znaczenie mają również zasoby dodatkowe, takie jak monterzy (lub roboty przemysłowe), narzędzia (lub aparaty montażowe) oraz kontenery z detalami (lub transportery specjalne). W niniejszym rozdziale zostaną przedstawione problemy alokacji operacji i wymienionych zasobów dodatkowych - dla linii z szeregowymi stanowiskami pracy.

W powszechnie przyjętej klasyfikacji wyróżnia się - w sensie zużycia - zasoby odnawialne, nieodnawialne oraz podwójnie ograniczone. Zasoby podwójnie ograniczone są ograniczone w każdej chwili (tak jak zasoby odnawialne) oraz w pewnym okresie czasu (tak jak zasoby nieodnawialne). Z punktu widzenia tej klasyfikacji dodatkowe zasoby w procesie montażu są podwójnie ograniczone, co można uzasadnić następująco.

Proces montażu na linii jest dyskretny, bowiem wyróżniamy w nim niepodzielne operacje oraz cykl pracy na kolejnych stanowiskach (stacjach). Rozważmy montaż pojedynczego obiektu. Chassis tego obiektu jest podawane na pierwsze stanowisko pracy. Dla montażu kompletnego obiektu wzdłuż linii wykorzystywane są dodatkowe zasoby: monterzy, narzędzia oraz kontenery z detalami. Zbiory monterów, narzędzi (ich typów) oraz kontenerów są ograniczone w procesie montażu pojedynczego obiektu. Ponadto na każdym stanowisku pracy (w każdym cyklu) może występować tylko jeden monter, najwyżej jedno narzędzie określonego typu oraz co najwyżej jeden kontener z detalami określonego typu. Zasoby te są przyporządkowane do określonych stacji montażowych.

Według innej klasyfikacji wyróżnia się zasoby substytucyjne oraz niesubstytucyjne. Ponieważ monterzy mogą wykonywać różne operacje, zatem stanowią oni zasób substytucyjny. W ogólnym przypadku za pomocą pewnego narzędzia można wykonać więcej niż jednę operację, stąd narzędzia można również traktować jako zasób substytucyjny. W przypadku kontenerów istnieje jednoznaczne przyporządkowanie typu kontenera do typu detali i operacji. Z tego względu kontenery potraktujemy jako zasób niesubstytucyjny.

Problemy z dodatkowymi zasobami są zwykle wielokryterialne. W tym sensie rozważane w poprzednich rozdziałach jednokryterialne problemy BLM oraz AO stanowią przypadki szczególne, w których dodatkowe zasoby były nieograniczone. W problemach AOiZ założymy, że dodatkowe zasoby są ograniczone.

6.2. ALOKACJA OPERACJI I KONTENERÓW

Rozważmy problem, w którym detale potrzebne do wykonania operacji montażowych znajdują się w kontenerach określonego typu. Założymy, że dla każdego detalu (operacji) istnieje tylko jeden typ kontenera. Równocześnie istnieją operacje wymagające detali tego samego typu - czyli tego samego kontenera. Grupując takie operacje na jednym stanowisku pracy minimalizuje się liczbę kontenerów (punktów dostaw detali) na linii montażowej. Jednakże takie podejście może spowodować zwiększenie liczby stanowisk pracy ponad ich minimalną liczbę. Zatem powstaje problem wielokryterialny.

6.2.1. Model matematyczny

Załóżmy, że dany jest zbiór Ω operacji montażowych ω_n, $n = 1,\ldots,N$, gdzie N - liczba operacji. Dane są czasy ϑ_n operacji ω_n oraz relacja kolejności w postaci macierzy Γ. Cykl c procesu montażu spełnia warunek:

$$\max_{1 \leq n \leq N} \vartheta_n \leq c < \sum_{n=1}^{n=N} \vartheta_n \tag{6.1}$$

Załóżmy, że mamy M typów kontenerów z detalami ($M < N$). Liczba kontenerów każdego typu dana jest wektorem:

$$Z = \left[z_m\right]_{m=1,\ldots,M} \tag{6.2}$$

gdzie: z_m - liczba kontenerów m-tego typu.

Przyporządkowanie typów zasobów (kontenerów, detali) do operacji dane jest macierzą:

$$D = \left[d_{m,n}\right]_{\substack{m=1,\ldots,M \\ n=1,\ldots,N}} \tag{6.3}$$

Elementy macierzy (6.3) określamy następująco:

$$d_{m,n} = \begin{cases} 1, \text{ jeśli do wykonania operacji } \omega_n \text{ potrzebny jest kontener z detalami m-tego typu} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{6.3a}$$

Ponadto przyjmujemy, że:

$$\bigwedge_{1 \leq n \leq N} \sum_{m=1}^{m=M} d_{m,n} = 1 \tag{6.4}$$

A zatem zasoby nie są substytucyjne.

Ponieważ $M < N$, zatem pewne operacje wymagają detali tego samego typu, tzn.:

$$\exists_m \sum_{n=1}^{n=N} d_{m,n} > 1 \tag{6.5}$$

Problem alokacji zasobów (kontenerów) pojawia się wówczas, gdy:

$$\exists_m \; z_m < \sum_{n=1}^{n=N} d_{m,n} \tag{6.6}$$

czyli liczba operacji wymagających detali z kontenerów m-tego typu jest większa od liczby tych kontenerów. Rozwiązanie tego problemu jest możliwe tylko przez grupowanie na jednym stanowisku pracy operacji wymagających detali z tego samego kontenera. Przyjmujemy bowiem, że na stanowisku pracy może znajdować się najwyżej jeden kontener każdego typu.

W rozważanym problemie AOiZ chodzi o minimalizację liczby stanowisk pracy oraz liczby kontenerów każdego typu. Przez Q_0 oznaczymy liczbę stanowisk pracy, natomiast przez Q_m, $m = 1,\ldots,M$ - liczbę kontenerów m-tego typu. Tak więc napiszemy:

$$Q_0 = \left[\frac{1}{c} \max_{1 \leq n \leq N} t_n\right]^+ \longrightarrow \min \tag{6.7}$$

gdzie: t_n - chwila zakończenia operacji ω_n.
Ponadto

$$Q_m = \sum_{k=1}^{k=K} q_{m,k} \longrightarrow \min \tag{6.8}$$

przy tym

$$q_{m,k} = \begin{cases} 1, \text{ jeśli } \exists_n \left\{\left[\frac{1}{c} t_n\right]^+ = k\right\} \wedge (d_{m,n}=1) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{6.8a}$$

W dalszym ciągu rozważań przyjmiemy, że określona jest hierarchia kryteriów: $Q_0, Q_1,\ldots,Q_m,\ldots,Q_M$ (Q_0 jest najwyższym w hierarchii).

Rozwiązanie dopuszczalne musi spełniać ograniczenia:

- niepodzielności operacji

$$\bigwedge_{1 \leq n \leq N} \left[\frac{1}{c} t_n\right]^+ = \left[\frac{1}{c}(t_n - v_n^{\gamma,l})\right]^+ \tag{6.9}$$

- kolejnościowe

$$\bigwedge_n \bigwedge_\nu (q_{\nu,n}=1) \Longrightarrow (t_\nu \leq t_n - v_n^{\gamma,l}) \tag{6.10}$$

- zasobowe

$$\bigwedge_{1 \leq m \leq M} Q_m \leq z_m \tag{6.11}$$

Sformułowany problem zostanie rozwiązany metodą programowania wieloetapowego.

6.2.2. Algorytm

W algorytmie programowania wieloetapowego zdefiniujemy stan $P^{\gamma,l}$ procesu decyzyjnego, wartość stanu $v^{\gamma,l}$ oraz określimy procedury generowania stanów i eliminacji stanów nieperspektywicznych. Stan $P^{\gamma,l}$ definiujemy tak, jak dla klasycznego problemu BLM.

Definicja 6.1

Wartość stanu jest wektorem:

$$v^{\gamma,l} = \left[v_j^{\gamma,l}\right]_{j=0,\ldots,M} \tag{6.12}$$

Współrzędne tego wektora można wyznaczyć na podstawie stanu $P^{\gamma,l}$ z formuł:

$$v_0^{\gamma,l} = \left[\frac{1}{c} \max_{1 \leq n \leq N} p_n^{\gamma,l}\right]^+ \tag{6.13}$$

oraz

$$v_m^{\gamma,l} = \sum_{k=1}^{k=K^{\gamma,l}} q_{m,k}^{\gamma,l} \tag{6.14}$$

przy tym:

$$q_{m,k}^{\eta,1} = \begin{cases} 1, \text{ jeśli } \exists_n \left\{ \left[\frac{1}{c} p_n^{\eta,1} \right]^+ = k \right\} \wedge (d_{m,n}=1) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{6.14a}$$

oraz

$$K^{\eta,1} = v_0^{\eta,1} \tag{6.14b}$$

Współrzędne wektora (6.12) można również wyznaczyć na podstawie wzorów rekurencyjnych - co zostanie pokazane dalej.

G e n e r o w a n i e s t a n ó w

Generowanie stanu $P^{\eta,1}$ na podstawie wybranego stanu $P^{\eta-1,\lambda}$ polega na przydzieleniu operacji ω_n do pewnego stanowiska pracy. Z uwagi na ograniczenia zasobowe operacja ω_n musi być przydzielona do stanowiska tak, by nie utracić rozwiązania polioptymalnego, a jednocześnie nie generować stanów niePerspektywicznych. W związku z tym istotne znaczenie mają warunki:

$$K^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} \geq \vartheta_n \tag{6.15}$$

oraz

$$\exists_i \left\{ \left[\frac{1}{c} p_i^{\eta-1,\lambda} \right]^+ = K^{\eta-1,\lambda} \right\} \wedge (d_{m,i}=1) \wedge (d_{m,n}=1) \tag{6.16}$$

gdzie: $T^{\eta-1,\lambda}$ jest chwilą zakończenia wszystkich operacji w stanie $P^{\eta-1,\lambda}$.

Jeżeli warunek (6.15) jest spełniony, to operacja ω_n jest przydzielana do $K^{\eta-1,\lambda}$-tej stacji. Spełnienie warunku (6.16) oznacza, że na $K^{\eta-1,\lambda}$-tej stacji znajduje się operacja ω_i, która wymaga detalu z tego samego kontenera co operacja ω_n. A zatem przydzielenie operacji ω_n do stacji $K^{\eta-1,\lambda}$ nie zwiększa liczby potrzebnych kontenerów.

Jeżeli warunek (6.16) nie jest spełniony, to operację ω_n trzeba przydzielać do stacji $K^{\eta-1,\lambda}$ oraz $K^{\eta-1,\lambda}+1$. Uzasadnienie takiego postępowania zostanie pokazane przy dyskutowaniu eliminacji stanów niePerspektywicznych.

W związku z powyższymi uwagami wyróżniamy cztery procedury generowania stanów:

I. Dla stanu $P^{0,1}$ procedura generowania stanów ma postać:

$$\forall_n \forall_{1 \leq \nu \leq N} (\vartheta_{\nu,n}=0) \Longrightarrow (P^{1,1} = P^{0,1} + \Delta P) \tag{6.17}$$

gdzie:

$$\Delta p_i = \begin{cases} t_n, \text{ jeśli } i=n \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{6.18}$$

oraz

$$t_n = \vartheta_n \tag{6.19}$$

II. Dla stanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunki (6.15) i (6.16) są spełnione, to procedura generowania stanów ma postać:

$$\forall_n \forall_\nu (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Longrightarrow (p_\nu^{\eta-1,\lambda} > 0) \right] \Rightarrow (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \tag{6.20}$$

Elementy wektora ΔP określamy na podstawie (6.18), natomiast t_n wyznaczamy ze wzoru:

$$t_n = T^{\eta-1,\lambda} + \vartheta_n \tag{6.21}$$

III. Dla stanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunek (6.15) nie jest spełniony, to (bez względu na warunek (6.16)), procedura generowania stanów ma postać:

$$\forall_n \forall_\nu (p_n^{\eta-1,\lambda}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Longrightarrow (p_\nu^{\eta-1,\lambda} > 0) \right] \wedge (K^{\eta-1,\lambda} < K) \wedge$$

$$\wedge (v_m^{\eta-1,\lambda} + 1 \leq z_m) \Longrightarrow (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \tag{6.22}$$

Elementy wektora ΔP określamy na podstawie (6.18), natomiast t_n wyznaczamy ze wzoru:

$$t_n = K^{\eta-1,\lambda} \cdot c + \vartheta_n \tag{6.23}$$

IV. Dla stanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunek (6.15) jest spełniony, a warunek (6.16) nie jest spełniony, to procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\nu (p_n^{\eta-1,\lambda}=0) \wedge \left[(\gamma_{\nu,n}=1) \Rightarrow (p_\nu^{\eta-1,\lambda} > 0)\right] \wedge (v_m^{\eta-1,\lambda}+1 \leq z_m) \Rightarrow$$

$$\Rightarrow (P^{\eta,l^1} = P^{\eta-1,\lambda} + \Delta P^1) \wedge (P^{\eta,l^2} = P^{\eta-1,\lambda} + \Delta P^2) \qquad (6.24)$$

Elementy wektorów ΔP^1 i ΔP^2 wyznaczamy na podstawie (6.18). Ponadto t_n^1 otrzymujemy ze wzoru (6.21), natomiast t_n^2 - ze wzoru (6.23). Stan P^{η,l^2} jest dopuszczalny, gdy $t_n^2 \leq K \cdot c$.

Równocześnie z generowaniem stanów można wyznaczyć współrzędne wektora wartości stanu $V^{\eta,l}$, bowiem:

$$v_0^{\eta,l} = \left[\frac{1}{c} t_n\right]^+ \qquad (6.25a)$$

oraz

$$v_j^{\eta,l} = \begin{cases} v_m^{\eta-1,\lambda}+1, & \text{jeśli } j=m \text{ oraz stan } P^{\eta,l} \text{ był wygenerowany} \\ & \text{za pomocą procedury I, III lub IV,} \\ v_j^{\eta-1,\lambda}, & \text{w przypadku przeciwnym} \end{cases} \qquad (6.25b)$$

$$j=1,\ldots,M.$$

Stosowanie wzorów (6.25) wymaga zapamiętywania wartości stanu.

6.2.3. Eliminowanie stanów nieperspektywicznych

W algorytmach programowania wielostepowego stany nieperspektywiczne są eliminowane za pomocą reguł wyczerpywania, sondowania oraz dominacji. Rozważane dotąd problemy były jednokryterialne. Dla rozwiązywanego w tym rozdziale problemu wielokryterialnego wprowadzimy pewne dodatkowe określenia.

Definicja 6.2a

Stan P^{N,l_1} dominuje nad stanem P^{N,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{0 \leq j \leq M} \bigvee_{0 \leq m \leq M} (v_j^{N,l_1} \leq v_j^{N,l_2}) \wedge (v_m^{N,l_1} < v_m^{N,l_2}) \qquad (6.26)$$

Definicja 6.2b

Stan P^{N,l_1} dominuje nad stanem P^{N,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{0 \leq j \leq M} (v_j^{N,l_1} = v_j^{N,l_2}) \wedge (l_1 < l_2) \qquad (6.27)$$

A zatem stan P^{N,l_1} został wygenerowany wcześniej niż stan P^{N,l_2}.

Definicja 6.3

Stany niezdominowane w sensie (6.26) lub (6.27) są Pareto-optymalne.

Stany Pareto-optymalne otrzymane ze stanu $P^{\eta,l}$ nazywamy lokalnie Pareto-optymalnymi i oznaczamy je przez P^{N,l^p}.

Definicja 6.4

Stan $P^{\eta,l}$ dominuje nad stanem $P^{e,\lambda}$, jeżeli każdy stan lokalnie Pareto-optymalny P^{N,λ^p} jest zdominowany przez pewien stan lokalnie Pareto-optymalny P^{N,l^p}.

W algorytmach programowania wielostapowego o dominacji stanów $P^{\eta,l}$ i $P^{e,\lambda}$ rozstrzyga się bez znajomości stanów lokalnie Pareto-optymalnych, na podstawie odpowiednich reguł dominacji.

Dla każdego stanu $P^{\eta,l}$ można wyznaczyć dolne ograniczenie $b_j^{\eta,l}$ $(j = 0,\ldots,M)$, j-tej współrzędnej v_j^{N,l_η} wartości odpowiednich stanów końcowych, tzn.:

$$b_j^{\eta,l} \leq v_j^{N,l_\eta^p} \qquad (6.28)$$

W oparciu o dolne ograniczenia $b_j^{\eta,l}$ formułowane są reguły sondowania oraz wyczerpywania.

Reguła sondowania w ogólnym przypadku ma postać twierdzenia.

TWIERDZENIE 6.1

Stan $P^{\eta,\lambda}$ jest nieperspektywiczny, jeżeli istnieje stan Pareto-optymalny P^{N,l^p} spełniający warunek:

$$\bigwedge_{0 \leq j \leq M} b_j^{\eta,\lambda} \geq v_j^{N,l^p} \qquad (6.29)$$

D o w ó d tego twierdzenia wynika wprost z (6.28) oraz (6.26) lub (6.27).

W regule wyczerpywania można wykorzystać dopuszczalną wartość g_j (K lub z_m, m =1,...,M) j-tego wskaźnika kryterialnego.

TWIERDZENIE 6.2

Stan $P^{\eta,l}$ jest nieperspektywiczny, jeśli spełnia warunek:

$$\exists_j \; g_j < b_j^{\eta,l} \tag{6.30}$$

D o w ó d wynika z faktu, że przy spełnieniu warunku (6.30) nie można wygenerować dopuszczalnego stanu końcowego.

W rozwiązywanym problemie AOiZ reguły eliminacji stanów nieperspektywicznych można sformułować następująco.

TWIERDZENIE 6.3

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$\bigwedge_{1 \leq m \leq M} (\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \wedge (v_m^{\eta,l_1} < v_m^{\eta,l_2}) \tag{6.31}$$

D o w ó d: Załóżmy, że dane są trajektorie lokalnie Pareto-optymalne $P^{\eta,l_2},\ldots,P^{N,l_2^p}$. Dla każdej trajektorii można napisać:

$$v_j^{N,l_2^p} = v_j^{\eta,l_2} + \Delta v_j^{N,l_2^p} \qquad j=0,\ldots,M \tag{6.32}$$

Jeżeli operacje $\omega_n \notin \Omega^{\eta,l_1}$ (oraz $\omega_n \notin \Omega^{\eta,l_2}$) zostaną przydzielone do stacji montażowych tak jak w trajektorii $P^{\eta,l_2},\ldots,P^{N,l_2^p}$, lecz od stanu P^{η,l_1}, to otrzymamy dopuszczalną trajektorię $P^{\eta,l_1},\ldots,P^{N,\lambda}$. Dopuszczalność tej trajektorii wynika bezpośrednio z (6.31). Analogicznie do (6.32) napiszemy:

$$v_j^{N,\lambda} = v_j^{\eta,l_1} + \Delta v_j^{N,\lambda} \qquad j=0,\ldots,M \tag{6.33}$$

Zauważmy, że:

$$v_0^{N,l_2^p} \geq v_0^{N,\lambda} \tag{6.34a}$$

a ponadto:

$$\Delta v_m^{N,\lambda} - \Delta v_m^{N,l_2^p} \leq 1 \tag{6.34b}$$

A zatem z (6.32) i (6.33) otrzymamy:

$$v_m^{N,\lambda} - v_m^{N,l_2^p} = (v_m^{\eta,l_1} - v_m^{\eta,l_2}) + (\Delta v_m^{N,\lambda} - \Delta v_m^{N,l_2^p}) \qquad m=1,\ldots,M \tag{6.35}$$

Uwzględniając (6.31) i (6.34b) w (6.35) otrzymamy:

$$v_m^{N,\lambda} \leq v_m^{N,l_2^p} \qquad m=1,\ldots,M \tag{6.36}$$

Ponieważ $T^{\eta,l_1} \leq T^{\eta,l_2}$, zatem:

$$v_j^{N,l_1^p} \leq v_j^{N,\lambda} \qquad j=0,\ldots,M \tag{6.37}$$

Z (6.34a), (6.36) i (6.37) wynika, że stan P^{η,l_1} dominuje nad stanem P^{η,l_2}. □

Dla reguł sondowania i wyczerpywania należy określić dolne ograniczenia współrzędnych wartości stanu lokalnie Pareto-optymalnego.

TWIERDZENIE 6.4

Dolne ograniczenia $b_j^{\eta,l}$, j=0,...,M, wyrażają wzory:

$$b_0^{\eta,l} = \left[\frac{1}{c}(T^{\eta,l} + \sum_{\omega_n \notin \Omega^{\eta,l}} \vartheta_n)\right]^+ \tag{6.38a}$$

$$b_m^{\eta,l} = v_m^{\eta,l} + \left[\frac{1}{c}(T^{\eta,l} + \sum_{\omega_n^m \notin \Omega^{\eta,l}} \vartheta_n)\right]^+ - v_0^{\eta,l} \qquad m=1,\ldots,M \tag{6.38b}$$

gdzie: ω_n^m - operacja wymagająca detalu m-tego typu.

D o w ó d: Wzór (6.38a) udowodniono w klasycznym modelu BLM. Dla dowodu (6.38b) zauważmy, że w stanie $P^{\eta,l}$ liczba wykorzystanych zasobów (kontenerów) m-tego typu wynosi $v_m^{\eta,l}$, przy tym na ostatnim stanowisku pracy $v_0^{\eta,l}$ może znajdować się zasób m-tego typu. Najmniejsze zapotrzebowanie na zasób m-tego typu wystąpi wówczas, jeżeli operacje ω_n^m będą

tworzyły enklawę, wykonywanę od chwili $T^{\eta,1}$. Liczba stanowisk pracy (pomijając $v_0^{\eta,1}$) potrzebnych dla wykonania tej enklawy jest najmniejszą liczbę kontenerów m-tego typu.

Rozważmy obecnie uzasadnienie dla przedstawionych wcześniej procedur generowania stanów.

TWIERDZENIE 6.5

Jeżeli w etanie $P^{\eta-1,\lambda}$, $1 < \eta \leq N$, dla operacji ω_n spełnione są warunki (6.15) i (6.16), to przydzielenie tej operacji do k-tej stacji, $K^{\eta-1,\lambda} < k \leq K$, daje stan nieperspektywiczny.

D o w ó d: Załóżmy, że przydzielając operację ω_n do $K^{\eta-1,\lambda}$-tej stacji otrzymamy stan $P^{\eta,1_1}$, natomiast przydzielając tę operację do k-tej stacji, $K^{\eta-1,\lambda} < k \leq K$, otrzymamy stan $P^{\eta,1_2}$. Rozważmy trajektorie lokalnie Pareto-optymalne $P^{\eta,1_2},\ldots,P^{N,1_2^p}$. Trajektorii takiej odpowiadają podzbiory operacji:

$$\Omega_k^{N,1_2^p},\ldots,\Omega_K^{N,1_2^p}$$

Zauważmy, że wychodząc ze stanu $P^{\eta,1_1}$ można skonstruować trajektorię $P^{\eta,1_1},\ldots,P^{N,\lambda}$, której odpowiadają podzbiory operacji:

$$\Omega_{K^{\eta,1_1}}^{N,\lambda},\ldots,\Omega_k^{N,\lambda},\ldots,\Omega_K^{N,\lambda}$$

gdzie:

$$\bigvee_{K^{\eta,1_1} < \varkappa < k} \Omega_\varkappa^{N,\lambda} = 0 \tag{6.39a}$$

a ponadto:

$$\Omega_k^{N,\lambda} = \Omega_k^{N,1_2^p} \setminus \omega_n \tag{6.39b}$$

$$\bigvee_{k < j \leq K} \Omega_j^{N,\lambda} = \Omega_j^{N,1_2^p} \tag{6.39c}$$

Ponadto:

$$\Omega_{K^{\eta,1_1}}^{N,\lambda} = \Omega_{K^{\eta,1_1}}^{N,1_2^p} \cup \omega_n \tag{6.39d}$$

oraz

$$\bigvee_{1 \leq i \leq K^{\eta,1_1}} \Omega_i^{N,\lambda} = \Omega_i^{N,1_2^p} \tag{6.39e}$$

Dla trajektorii $P^{0,1},\ldots,P^{\eta,1_1},\ldots,P^{N,\lambda}$ możemy napisać:

$$v_m^{N,\lambda} = \sum_{s=1}^{s=K} q_{m,s}^{N,\lambda} \qquad m=1,\ldots,M \tag{6.40a}$$

natomiast dla trajektorii $P^{0,1},\ldots,P^{\eta,1_2},\ldots,P^{N,1_2^p}$ - analogicznie:

$$v_m^{N,1_2^p} = \sum_{s=1}^{s=K} q_{m,s}^{N,1_2^p} \qquad m=1,\ldots,M \tag{6.40b}$$

Z (6.39) oraz (6.15) i (6.16) widzimy, że:

$$q_{m,s}^{N,\lambda} = q_{m,s}^{N,1_2^p}, \quad (s=1,\ldots,K) \wedge (s \neq k) \tag{6.41a}$$

$$q_{m,k}^{N,\lambda} \leq q_{m,k}^{N,1_2^p}, \quad s=k \tag{6.41b}$$

A zatem:

$$v_m^{N,\lambda} < v_m^{N,1_2^p} \tag{6.42a}$$

oraz

$$v_\mu^{N,\lambda} = v_\mu^{N,1_2^p}, \quad (\mu=0,\ldots,M) \wedge (\mu \neq m) \tag{6.42b}$$

Na podstawie (6.42) oraz definicji (6.26) i (6.27) wnioskujemy, że stan $P^{\eta,1_1}$ dominuje nad stanem $P^{\eta,1_2}$. □

WNIOSEK 6.1

Jeżeli w stanie $P^{\eta-1,\lambda}$ operacja ω_n spełnia warunek (6.15), lecz nie spełnia warunku (6.16), to przydzielenie tej operacji tylko do $K^{\eta-1,\lambda}$-tej stacji może spowodować utratę trajektorii polioptymalnej.

D o w ó d: Jeżeli warunek (6.16) nie jest spełniony, to stosując te same oznaczenia jak w tw. 6.5 otrzymamy:

$$(q^{N,\lambda}_{m,K^{\eta,l_1}} \geq q^{N,l_2^p}_{m,K^{\eta,l_1}}) \wedge (q^{N,\lambda}_{m,k} \leq q^{N,l_2^p}_{m,k}) \tag{6.43}$$

Pozostałe składniki sum (6.40) pozostają bez zmiany. W konsekwencji nierówność (6.42a) może być zastępiona przez:

$$v_m^{N,\lambda} \geq v_m^{N,l_2^p} \tag{6.44}$$

A zatem stan P^{η,l_2} może dominować nad stanem P^{η,l_1}. Stąd też zastosowano procedurę (6.24) generowania stanów.

6.2.4. Polioptymalizacja

Rozwiązanie polioptymalne jest jednym z rozwiązań Pareto-optymalnych. Można je wyznaczyć stosując globalny model preferencji decydenta Q. Model ten może wyrażać: sumę ważoną, odpowiedni dystans pomiędzy punktami w przestrzeni kryteriów lub zagregowaną funkcję przynależności (dla programowania rozmytego).

Definicja 6.5

Stan polioptymalny P^p jest stanem Pareto-optymalnym spełniającym warunek:

$$\min_l Q^{N,l^p} = Q^p \tag{6.45}$$

Tak więc w algorytmach programowania wieloetapowego można generować kolejno stany końcowe i porównywać je ze stanem aktualnie polioptymalnym (najlepszym spośród wygenerowanych). Zatem nie trzeba zapamiętywać stanów Pareto-optymalnych.

Wprowadzenie syntetycznej wartości stanu (zamiast wektora (6.12)) jest dopuszczalne tylko w przypadku braku ograniczeń g_j, j=1,...,M. Jeżeli takie ograniczenia istnieją, to wartość stanu musi być wektorem. Ponadto reguły dominacji oraz wyczerpywania stanów wymagają znajomości współrzędnych wektora wartości stanu. Taki przypadek reprezentuje funkcja użyteczności w postaci ograniczeń progowych lub hierarchizacji kryteriów.

W metodzie hierarchizacji kryteriów przyjmuje się, że kryteria są uporządkowane w sensie ich ważności (załóżmy, że od Q_0 do Q_M). Ponadto zakłada się tolerancje (bezwzględne) δ_j, dla wskaźników kryterialnych Q_j, j = 0,...,M. W ten sposób z dwóch rozwiązań pierwsze jest lepsze od drugiego, jeżeli jest spełniony warunek:

$$\exists_m \bigvee_{1 \leq j \leq m} (Q_1^1 < Q_1^2 - \delta_1) \vee \left\{ |Q_j^1 - Q_j^2| \leq \delta_j \right\} \wedge (Q_m^1 < Q_m^2 - \delta_m) \tag{6.46}$$

W programowaniu wieloetapowym warunek (6.46) odnosi się do dwóch stanów końcowych (N-tego etapu). Jednakże stany końcowe są generowane sekwencyjnie. Z tego względu w metodzie hierarchizacji kryteriów nie wystarcza zapamiętywać tylko stanu aktualnie polioptymalnego. W ogólnym przypadku trzeba zapamiętywać pewien podzbiór stanów Pareto-optymalnych, które nazwiemy perspektywicznymi stanami Pareto-optymalnymi.

Algorytm wyznaczania rozwiązania polioptymalnego metodą hierarchizacji kryteriów składa się z następujących kroków obliczeniowych:

Krok 1. Sprawdzić, czy wygenerowano wszystkie stany N-tego etapu. Jeżeli nie, to wygenerować kolejny stan końcowy $P^{N,\lambda}$ i przejść do kroku 2. Jeżeli tak, to przejść do kroku 4.

Krok 2. Sprawdzić, czy stan $P^{N,\lambda}$ jest Pareto-optymalny. Jeżeli tak, to przejść do kroku 3. Jeżeli nie, to pominąć stan $P^{N,\lambda}$ i przejść do kroku 1.

Krok 3. Sprawdzić, czy stan $P^{N,\lambda}$ jest perspektywicznym stanem Pareto-optymalnym. Jeżeli tak, to włączyć go do zbioru perspektywicznych stanów Pareto-optymalnych. Jeżeli nie, to pominąć stan $P^{N,\lambda}$ i przejść do kroku 1.

Krok 4. Wyznaczyć stan polioptymalny ze zbioru perspektywicznych stanów Pareto-optymalnych.

Sprawdzenie, czy stan $P^{N,\lambda}$ jest perspektywicznym stanem Pareto-optymalnym, jest oparte tylko na najwyższym kryterium w przyjętej hierarchii. Niech P^{N,l^p} jest perspektywicznym stanem Pareto-optymalnym. Stan $P^{N,\lambda}$ jest perspektywicznym stanem Pareto-optymalnym, jeżeli spełnia warunek:

$$v_0^{N,\lambda} < \min_l v_0^{N,l^p} + \delta_0 \tag{6.47}$$

A zatem stosowany jest tylko pierwszy człon alternatywy (6.46). Warto podkreślić, że jeżeli:

$$v_0^{N,\lambda} + \delta_0 < v_0^{N,i^P} \tag{6.48}$$

to stany P^{N,i^P} są usuwane ze zbioru perspektywicznych stanów Pareto-optymalnych.

Wyznaczenie stanu polioptymalnego P^P ze zbioru perspektywicznych stanów Pareto-optymalnych jest oparte na kryteriach: $Q_1,\ldots,Q_m,\ldots,Q_M$. Ponieważ zbiór perspektywicznych stanów Pareto-optymalnych jest dany explicite, zatem rozwiązanie polioptymalne wyznaczamy według znanego schematu. W k-tym kroku, k = 1,...,M, eliminujemy stany P^{N,i^P}, które spełniają warunek:

$$\min_l v_k^{N,l} + \delta_k < v_k^{N,i^P} \tag{6.49}$$

W ten sposób w M-tym kroku otrzymujemy rozwiązanie polioptymalne.

6.3. ALOKACJA OPERACJI I NARZĘDZI

Rozważmy obecnie problem alokacji operacji i narzędzi. Narzędzia są traktowane jako zasoby substytucyjne. Przyjmiemy, że dla wykonania każdej operacji potrzebne jest jedno narzędzie. Jednakże może to być jedno z narzędzi dopuszczalnego typu. Ponadto założymy, że na stanowiska pracy może być przydzielone najwyżej jedno narzędzie każdego typu. A zatem różne operacje mogą być wykonywane narzędziem tego samego typu oraz różnymi (dopuszczalnymi) narzędziami można wykonywać tę samą operację.

Analogicznie jak dla alokacji operacji i kontenerów rozwiązany zostanie problem wielokryterialny. Jako kryteria przyjmiemy liczbę stanowisk pracy oraz liczbę narzędzi każdego typu. Założymy, że kryteria te tworzę określoną hierarchię. Grupując na stanowisku pracy operacje wymagające tego samego narzędzia, minimalizuje się liczbę potrzebnych narzędzi. Równocześnie jednak może to prowadzić do zwiększenia liczby (ponad minimalną) stanowisk pracy. Rozwiązanie poliotymalne zostanie wyznaczone metodą hierarchizacji kryteriów.

6.3.1. Model matematyczny

Załóżmy, że dany jest zbiór Ω operacji ω_n, n = 1,...,N. Relacja kolejności wykonywania operacji dana jest macierzą Γ. Przyjmiemy, że danych jest M typów narzędzi (M < N). Przez z_m oznaczymy liczbę narzędzi m-tego typu. Przyporządkowanie typów zasobów (narzędzi) do operacji dane jest macierzą (6.3), gdzie $d_{m,n}$ = 1, jeśli do wykonania operacji ω_n potrzebne jest narzędzie m-tego typu ($d_{m,n}$ = 0, w przypadku przeciwnym).

Ponieważ zakładamy, że narzędzia są zasobami substytucyjnymi, zatem zamiast (6.4) napiszemy:

$$\exists_{1 \le n \le N} \sum_{m=1}^{m=M} d_{m,n} > 1 \tag{6.50}$$

czyli operacja ω_n może być wykonana różnymi narzędziami. Ponadto nierówność (6.5) pozostaje w mocy.

Problem alokacji zasobów (narzędzi) pojawia się wówczas, gdy jest spełniona nierówność (6.6), czyli w konsekwencji:

$$\sum_{m=1}^{m=M} z_m < N \tag{6.51}$$

A zatem w celu rozwiązania problemu należy grupować na stanowiskach operacje, które mogę być wykonywane narzędziem tego samego typu.

Założymy, że czasy wykonania operacji dane są macierzą:

$$\Theta = \left[\vartheta_{m,n}\right]_{\substack{m=1,\ldots,M \\ n=1,\ldots,N}} \tag{6.52}$$

gdzie: $\vartheta_{m,n}$ - czas wykonania operacji ω_n narzędziem m-tego typu. Wprowadzając macierz (6.52) można przyjąć, że $d_{m,n}$ = 1, jeśli $\vartheta_{m,n} \le c$. A zatem przyporządkowanie zasobów (narzędzi) do operacji jest zależne od cyklu.

W modelu matematycznym AOiZ (narzędzi jako zasobu substytucyjnego) przyjmiemy hierarchię kryteriów: $Q_0,Q_1,\ldots,Q_m,\ldots,Q_M$ (Q_0 - jest najwyższym w hierarchii). Funkcje kryterialne mają postać (6.7) i (6.8), z tym że dla zasobów substytucyjnych przyjmujemy:

$$q_{m,k} = \begin{cases} 1, \text{ jeśli na k-tej stacji wykorzystywane jest narzędzie m-tego typu} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{6.53}$$

Rozwiązanie dopuszczalne musi spełniać ograniczenia: niepodzielności operacji (6.9), kolejnościowe (6.10) oraz zasobowe (6.11). Sformułowany problem zostanie rozwiązany metodą programowania wieloetapowego.

6.3.2. Algorytm

W algorytmie programowania wieloetapowego dla AOiZ, w przypadku zasobów substytucyjnych, wprowadzona zostanie odmienna od dotychczasowych, definicja stanu. Stan powinien zawierać informację o numerze stacji, do któ-

rej przydzielono operację oraz o typie narzędzia wykorzystanego do wykonania operacji. Pozostałe elementy algorytmu są analogiczne do przedstawionych w punkcie 6.2.

Definicja 6.6

Dla zasobów substytucyjnych stan jest macierzą:

$$P^{\eta,l} = \left[p^{\eta,l}_{i,j}\right]_{\substack{i=1,\ldots,N \\ j=1,2}} \tag{6.54}$$

Elementy tej macierzy określamy następująco:

$$p^{\eta,l}_{i,1} = \begin{cases} m, & \text{jeśli do wykonania operacji } \omega_i \text{ wykorzystano narzędzie } m\text{-tego typu} \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{6.54a}$$

$$p^{\eta,l}_{i,2} = \begin{cases} t_{m,n}, & \text{jeśli } p^{\eta,l}_{p,1} = m \\ 0, & \text{w przypadku przeciwnym,} \end{cases} \tag{6.54b}$$

gdzie: $t_{m,n}$ - chwila zakończenia operacji ω_n za pomocą narzędzia m-tego typu.

A zatem stan początkowy $P^{0,1}$ jest macierzą zerową, natomiast każdy stan końcowy $P^{N,l}$ ma wszystkie współrzędne dodatnie. Ze stanu $P^{N,l}$ można odczytać wprost dopuszczalną AOiZ, ponieważ:

$$k_n = \left[\frac{1}{c}\, p^{N,l}_{n,2}\right]^{+} \tag{6.55}$$

gdzie: k_n - numer stacji, do której przydzielono operację ω_n. Ponadto współrzędna $p^{N,l}_{n,1}$ określa wprost typ narzędzia wykorzystanego do wykonania operacji ω_n.

Wartość stanu $V^{\eta,l}$ zdefiniujemy w postaci wektora (6.12). Współrzędne tego wektora można wyznaczyć na podstawie stanu $P^{\eta,l}$ lub rekurencyjnie w trakcie generowania stanów. Ze stanu $P^{\eta,l}$ otrzymujemy:

$$v^{\eta,l}_0 = \left[\frac{1}{c} \max_{1 \le i \le N} p^{\eta,l}_{i,2}\right]^{+} \tag{6.56a}$$

oraz:

$$q^{\eta,l}_{m,k} = \begin{cases} 1, & \text{jeśli } \exists_n \left\{\left[\frac{1}{c}\, p^{\eta,l}_{n,2}\right]^{+} = k\right\} \wedge (p^{\eta,l}_{n,1} = m) \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{6.56b}$$

Dla wyznaczenia $v^{\eta,l}_m$ stosujemy wzór (6.14), którego składniki wyznaczamy z (6.56b).

Zauważmy, że dla problemu AOiZ, w przypadku zasobów substytucyjnych, chwilę $T^{\eta,l}$ zakończenia operacji w stanie $P^{\eta,l}$ wyznaczamy ze wzoru:

$$T^{\eta,l} = \max_{1 \le i \le N} p^{\eta,l}_{i,2} \tag{6.57}$$

G e n e r o w a n i e s t a n ó w

Generowanie stanów dla substytucyjnych zasobów przebiega analogicznie jak dla zasobów niesubstytucyjnych. A zatem stosowane są cztery procedury generowania stanów, w zależności od spełniania warunków:

$$\exists_m K^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} \ge v^{\eta}_{m,n} \tag{6.58}$$

oraz:

$$\exists_i \left\{\left[\left[\frac{1}{c}\, p^{\eta-1,\lambda}_{i,2}\right]^{+} = K^{\eta-1,\lambda}\right] \wedge (p^{\eta-1,\lambda}_{i,1}=1) \wedge (d_{m,n}=1)\right\} \tag{6.59}$$

Istotna różnica polega jedynie na tym, że dla realizacji operacji ω_n przydziela się narzędzia (dopuszczalne) różnego typu.

Generując stany sprawdzamy warunki (6.58) i (6.59) dla każdej operacji ω_n, która nie została przydzielona do realizacji. Ponadto warunki te sprawdzamy dla każdego narzędzia dopuszczalnego typu, tzn. dla $d_{m,n}=1$. A zatem otrzymujemy następujące procedury generowania stanów:

I. Dla stanu $P^{0,1}$

$$\forall_n \; \forall_{1 \le \eta \le N} \; \forall_m \; (q_{\eta,n}=0) \wedge (d_{m,n}=1) \wedge (v^{\eta}_{m,n} \le c) \Rightarrow (P^{1,1} = P^{0,1} + \Delta P) \tag{6.60}$$

gdzie:

$$\Delta p_{i,1} = \begin{cases} m, & \text{jeśli } i=n \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{6.61a}$$

oraz

$$\Delta p_{i,2} = \begin{cases} v^{\eta}_{m,n}, & \text{jeśli } \Delta p_{i,1}=m \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{6.61b}$$

II. Dla stanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunki (6.58) i (6.59) są spełnione:

$$\bigwedge_n \bigwedge_\vartheta \bigwedge_m (p^{\eta-1,\lambda}_{n,1}=0) \wedge \left[(\vartheta_{\vartheta,n}=1) \Rightarrow (p^{\eta-1,\lambda}_{\vartheta,1} > 0)\right] \wedge (d'_{m,n}=1) \Rightarrow$$

$$\Rightarrow (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \tag{6.62}$$

Elementy macierzy ΔP określamy następująco:

$$\Delta p_{i,1} = \begin{cases} m, & \text{jeśli } i=n \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{6.63a}$$

oraz

$$\Delta p_{i,2} = \begin{cases} t_{m,n}, & \text{jeśli } \Delta p_{i,1}=m \\ 0, & \text{w przypadku przeciwnym} \end{cases} \tag{6.63b}$$

ponadto przyjmując najkrótszy czas $\vartheta^{\eta}_{m,n}$ wykonania operacji, otrzymamy

$$t_{m,n} = T^{\eta-1,\lambda} + \vartheta^{\eta}_{m,n} \tag{6.63c}$$

III. Dla stanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunek (6.58) nie jest spełniony, to (bez względu na warunek (6.59)) procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\vartheta \bigwedge_m (p^{\eta-1,\lambda}_{n,1}=0) \wedge \left[(\vartheta_{\vartheta,n}=1) \Rightarrow (p^{\eta-1,\lambda}_{\vartheta,1} > 0)\right] \wedge (K^{\eta-1,\lambda} < K) \wedge$$

$$\wedge (d_{m,n}=1) \wedge (v^{\eta-1,\lambda}_{m}+1 \leq z_m) \Rightarrow (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \tag{6.64}$$

Elementy macierzy ΔP określamy przez (6.63a) i (6.63b), jednakże:

$$t_{m,n} = K^{\eta-1,\lambda} \cdot c + \vartheta^{\eta}_{m,n} \tag{6.65}$$

IV. Dla etanu $P^{\eta-1,\lambda}$ $(1 < \eta \leq N)$, jeżeli warunek (6.58) jest spełniony, a warunek (6.59) nie jest spełniony, to procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_\vartheta \bigwedge_m (p^{\eta-1,\lambda}_{n,1}=0) \wedge \left[(\vartheta_{\vartheta,n}=1) \Rightarrow (p^{\eta-1,\lambda}_{\vartheta,1} > 0)\right] \wedge (d_{m,n}=1) \wedge$$

$$\wedge (v^{\eta-1,\lambda}_{m}+1 \leq z_m) \Rightarrow \left[(P^{\eta,1_1} = P^{\eta-1,\lambda} + \Delta P^1) \wedge (P^{\eta,1_2} = P^{\eta-1,\lambda} + \Delta P^2)\right] \tag{6.66}$$

Elementy macierzy ΔP^1 i ΔP^2 wyznaczamy z (6.63a) i (6.63b). Ponadto $t^1_{m,n}$ otrzymujemy z formuły (6.63c)*), natomiast $t^2_{m,n}$ - z formuły (6.65). Stan $P^{\eta,1_2}$ jest dopuszczalny, gdy $t^2_{m,n} \leq K$. c.

Równocześnie z generowaniem stanów można wyznaczyć współrzędne wektora wartości $V^{\eta,1}$, bowiem:

$$v^{\eta,1}_0 = \left[\frac{1}{c} t_{m,n}\right]^* \tag{6.67a}$$

oraz

$$v^{\eta,1}_j = \begin{cases} v^{\eta-1,\lambda}_m + 1, & \text{dla } j=m, \text{ jeśli stan } P^{\eta,1} \text{ był wygenerowany za pomocą proceduryI,III lub IV} \\ v^{\eta-1,\lambda}_j, & \text{w przypadku przeciwnym.} \end{cases} \tag{6.67b}$$

E l i m i n o w a n i e s t a n ó w n i e p e r s p e k t y w i c z n y c h

Do eliminacji stanów nieperspektywicznych można wykorzystać reguły dominacji, wyczerpywania oraz sondowania.

Reguła dominacji jest oparta na twierdzeniu (6.31). Dowód tego twierdzenia dla zasobów substytucyjnych przeprowadza się identycznie jak dla zasobów niesubstytucyjnych. Ponadto analogicznie jak dla zasobów niesubstytucyjnych można uzasadnić stosowanie czterech procedur generowania stanów.

W ogólnym przypadku określenie dolnych ograniczeń zużycia zasobów substytucyjnych jest problemem złożonym. Z tego względu pominiemy regułę sondowania.

Reguła wyczerpywania jest oparta na oszacowaniu (6.38a).

TWIERDZENIE 6.6

Stan $P^{\eta,1}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$(b^{\eta,1}_0 > K) \vee \left(\sum_{m=1}^{m=M} v^{\eta,1}_m + b^{\eta,1}_0 - K^{\eta,1} > \sum_{m=1}^{m=M} z_m\right) \tag{6.68}$$

*) podstawiając czas $\vartheta^{\eta}_{m,n}$

D o w ó d: Pierwszy składnik alternatywy (6.68) można uwzględnić wprost w (6.30). Natomiast drugi składnik tej alternatywy dotyczy minimalnego zapotrzebowania na narzędzia w stanie $P^{\eta,1}$. Jeżeli liczba potrzebnych narzędzi jest większa od liczby dostępnych narzędzi, to ze stanu $P^{\eta,1}$ nie można uzyskać dopuszczalnego rozwiązania problemu.

Rozwiązanie Pareto-optymalne oraz polioptymalne można wyznaczyć za pomocą algorytmu przedstawionego w punkcie 6.2.

6.4. ALOKACJA OPERACJI I MONTERÓW

Monterzy stanowią zasób substytucyjny. Na każdej stacji może pracować jeden monter (założono linię o szeregowych stanowiskach pracy). Ponadto każdy monter może być przydzielony tylko do jednej stacji. Z tego względu problem alokacji monterów może być traktowany jako szczególny przypadek problemu alokacji narzędzi, w którym $z_m = 1$, $m = 1,\ldots,M$.

Ponieważ na każdej stacji musi pracować jeden monter, zatem problem alokacji operacji i monterów jest jednokryterialny. Liczba monterów w procesie montażu jest równa liczbie stanowisk pracy. Jednakże z uwagi na różne czasy wykonywania tej samej opracji przez różnych monterów, problem alokacji operacji i monterów nie jest równoważny problemom BLM lub AO.

6.4.1. Model matematyczny

Załóżmy, że dany jest zbiór Ω operacji ω_n, $n = 1,\ldots,N$. Relacja kolejności wykonywania operacji określona jest macierzą Γ. Do wykonania operacji danych jest M monterów ($M < N$). Czasy wykonania operacji dane są macierzą (6.52), gdzie $\vartheta_{m,n}$ oznacza czas wykonania operacji ω_n przez m-tego montera. W procesie montażu dany jest cykl c. Jeżeli:

$$c < \vartheta_{m,n} \tag{6.69}$$

to m-ty monter nie może na linii wykonać operacji ω_n.

Na linii szeregowej na każdym stanowisku pracuje jeden monter, co stanowi ograniczenie zasobowe. Dopuszczalne rozwiązanie problemu alokacji operacji i zasobów musi spełniać ponadto ograniczenia niepodzielności orez kolejności operacji:

$$\bigwedge_{1 \leq n \leq N} \exists_m \left[\frac{1}{c} t_{m,n}\right]^* = \left[\frac{1}{c}(t_{m,n} - \vartheta_{m,n})\right]^* \tag{6.70a}$$

$$\bigwedge_n \bigwedge_\nu \exists_m \exists_\mu (\delta_{\nu,n}=1) \Rightarrow (t_{\mu,\nu} \leq t_{m,n} - \vartheta_{m,n}) \tag{6.70b}$$

gdzie: $t_{m,n}$ - chwila zakończenia operacji ω_n przez m-tego montera.

W rozważanym problemie AOiZ przyjmujemy kryterium minimalizacji liczby stanowisk pracy, co zapiszemy w postaci:

$$Q = \left[\frac{1}{c} \max_{1 \leq n \leq N} t_{m_n,n}\right]^* \longrightarrow \min \tag{6.71}$$

Dla rozwiązania dopuszczalnego spełniony jest warunek:

$$Q \leq M \leq K \tag{6.71a}$$

6.4.2. Algorytm

Dla rozwiązania sformułowanego problemu wykorzystamy algorytm programowania wieloetapowego. Stan $P^{\eta,1}$ zdefiniujemy tak jak w (6.54) przyjmując, że $p^{\eta,1}_{n,1} = m$, jeżeli m-ty monter wykonuje operację ω_n.

Definicja 6.7

Wartość stanu jest skalarem:

$$v^{\eta,1} = \left[\frac{1}{c} \max_{1 \leq i \leq N} p^{\eta,1}_{i,2}\right]^* \tag{6.72}$$

Wartość stanu można również wyznaczyć wprost w trakcie generowania stanów.

G e n e r o w a n i e s t a n ó w

Załóżmy, że dany jest stan $P^{\eta-1,\lambda}$, $1 < \eta \leq N$. Oznaczmy przez μ numer montera, który wg stanu $P^{\eta-1,\lambda}$ został przydzielony na $K^{\eta-1,\lambda}$-te stanowisko pracy (stację montażową). Numer ten określamy ze wzoru:

$$\exists_i (p^{\eta-1,\lambda}_{i,2} = T^{\eta-1,\lambda}) \Rightarrow (\mu = p^{\eta-1,\lambda}_{i,1}). \tag{6.73}$$

Wyróżnimy dwie procedury generowania stanów, w zależności od spełnienia warunku:

$$v^{\eta-1,\lambda} \cdot c - T^{\eta-1,\lambda} > \vartheta_{\mu,n} \tag{6.74}$$

Jeżeli warunek (6.74) jest spełniony, to procedura generowanie stanów ma postać:

$$\bigwedge_n \bigwedge_\nu (p^{\eta-1,\lambda}_{n,1}=0) \wedge \left[(\delta_{\nu,n}=1) \Rightarrow (p^{\eta-1,\lambda}_{\nu,1} > 0)\right] \Rightarrow$$
$$\Rightarrow (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \tag{6.75}$$

Elementy macierzy ΔP określamy z (6.63), ponieważ operacja ω_n jest przydzielona μ-temu monterowi.

Jeżeli warunek (6.74) nie jest spełniony, to operacja ω_n jest wykonywana na $K^{\eta-1,\lambda}+1$-szej stacji przez m-tego montera, który nie został przydzielony do poprzednich stacji montażowych. A zatem w tym przypadku procedura generowania stanów ma postać:

$$\bigwedge_n \bigwedge_{\nu} \bigwedge_m \bigwedge_{1\le j\le N} (p^{\eta-1,\lambda}_{n,1}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p^{\eta-1,\lambda}_{\nu,1} > 0)\right] \wedge$$

$$\wedge (p^{\eta-1,\lambda}_{j,1}\neq m) \wedge (\vartheta^{\nu}_{m,n} < c) = (P^{\eta,1} = P^{\eta-1,\lambda} + \Delta P) \qquad (6.76)$$

Elementy macierzy ΔP określamy w tym przypadku przez (6.63a), (6.63b) oraz (6.65).

Generowanie stanów z $P^{0,1}$ przeprowadza się zgodnie z procedurą (6.60).

W trakcie generowania stanów wyznaczamy wprost wartość stanu:

$$v^{\eta,1} = \left[\frac{1}{c}\, t_{m,n}\right]^{+} \qquad (6.77)$$

jeżeli operację ω_n przydzielono m-temu monterowi.

E l i m i n o w a n i e e t a n ó w n i e p e r s p e k t y w i c z n y c h

Do eliminacji etanów nieperspektywicznych wykorzystuje się reguły wyczerpywania, sondowania oraz dominacji. Reguła wyczerpywania jest oparta na następującym twierdzeniu

TWIERDZENIE 6.7

Stan $P^{\eta,1}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$\bigvee_n \bigwedge_m \bigwedge_{1\le i\le N} (p^{\eta,1}_{n,1}=0) \wedge (p_{i,1}\neq m) \wedge (c < \vartheta^{\nu}_{m,n}) \vee (K^{\eta,1}.c - T^{\eta,1} < \vartheta^{\nu}_{\mu,n}) \qquad (6.78)$$

D o w ó d: Zgodnie z przedstawionymi procedurami generowania stanów oraz (6.78) operacja ω_n nie spełnia warunku dla wygenerowania stanu. A zatem ze stanu $P^{\eta,1}$ nie można dojść do dopuszczalnego stanu końcowego.

Reguła sondowania ma postać twierdzenia.

TWIERDZENIE 6.8

Stan $P^{\eta,1}$ jest nieperspektywiczny, jeżeli spełnia warunek:

$$v^{a} \le \left[\frac{1}{c}\,(T^{\eta,1} + \sum_{\omega_n \notin \Omega^{\eta,1}} \vartheta^{\nu'}_{m_n,n})\right]^{+} \qquad (6.79)$$

gdzie: v^{a} - wartość aktualnie najlepszego stanu końcowego, $\vartheta^{\nu'}_{m_n,n}$ - najkrótszy czas wykonania operacji ω_n przez jednego z monterów nie przydzielonego do k-tej stacji, $k = 1,\ldots,K^{\eta,1}-1$.

D o w ó d: Można zauważyć, przez analogię do (6.38a), że prawa strona nierówności (6.79) jest dolnym ograniczeniem liczby stanowisk pracy. A zatem zgodnie z definicję (6.29) etan $P^{\eta,1}$ jest nieperspektywiczny. Formuły (6.38a) i (6.79) pokrywają się, jeżeli czasy wykonania operacji są takie same dla różnych monterów.

Rozważmy dwie reguły dominacji stanów, stosowane w zależności od spełnienia warunku:

$$\bigwedge_n \bigwedge_{\nu} (p^{\eta,1}_{n,1}=0) \wedge \left[(\vartheta_{\nu,n}=1) \Rightarrow (p^{\eta,1}_{\nu,1})>0\right] \Rightarrow (T^{\eta,1} + \vartheta^{\nu}_{\mu,n} > K^{\eta,1}\,.\,c) \qquad (6.80)$$

A zatem w stanie $P^{\eta,1}$ na ostatnią stację, tzn. $K^{\eta,1}$, nie można już przydzielić żadnej operacji.

TWIERDZENIE 6.9

Stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}, jeżeli oba spełniają warunek (6.80) oraz warunek:

$$(\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}) \wedge (W^{\eta_1,l_1} = W^{\eta_2,l_2}) \qquad (6.81)$$

gdzie: $W^{\eta,1}$ - zbiór monterów wykorzystywanych w stanie $P^{\eta,1}$.

Jeżeli

$$\bigvee_n p^{\eta,1}_{n,i} = m \qquad (6.82)$$

to m-ty monter należy do zbioru $W^{\eta,1}$.

D o w ó d: Z warunku (6.81) wynika, że:

$$v^{\eta_1,l_1} = v^{\eta_2,l_2} \qquad (6.83)$$

Dla trajektorii lokalnie optymalnej $P^{\eta_2,l_2},\ldots,P^{N,l_2^{o}}$ możemy napisać:

$$v^{N,l_2^{o}} = v^{\eta_2,l_2} + \Delta v^{2} \qquad (6.84)$$

gdzie: Δv^2 - minimalna liczba stanowisk pracy dla wykonania operacji $\omega_n \notin \Omega^{\eta_2,l_2}$ przez monterów nie należących do zbioru W^{η_2,l_2}. Analogicznie dla lokalnie optymalnej trajektorii $P^{\eta_1,l_1},\ldots,P^{N,l_1^o}$ otrzymamy:

$$v^{N,l_1^o} = v^{\eta_1,l_1} + \Delta v^1 \tag{6.85}$$

Ponieważ zgodnie z (6.81), $\Omega^{\eta_1,l_1} \supseteq \Omega^{\eta_2,l_2}$ oraz $W^{\eta_1,l_1} = W^{\eta_2,l_2}$ zatem:

$$\Delta v^1 \leq \Delta v^2 \tag{6.86}$$

Uwzględniając (6.86) w (6.85) otrzymamy:

$$v^{N,l_1^o} \leq v^{N,l_2^o} \tag{6.87}$$

czyli stan P^{η_1,l_1} dominuje nad stanem P^{η_2,l_2}. □

W przypadku gdy stany P^{η,l_1} i P^{η,l_2} nie spełniają warunku (6.80) reguła dominacji stanów ma postać twierdzenia.

TWIERDZENIE 6.10

Stan P^{η,l_1} dominuje nad stanem P^{η,l_2}, jeżeli jest spełniony warunek:

$$(\Omega^{\eta,l_1} = \Omega^{\eta,l_2}) \wedge (W^{\eta,l_1} = W^{\eta,l_2}) \wedge (\mu^{\eta,l_1} = \mu^{\eta,l_2}) \wedge$$

$$\wedge (T^{\eta,l_1} \leq T^{\eta,l_2}) \tag{6.88}$$

gdzie: $\mu^{\eta,l}$ - numer montera, który został przydzielony na ostatnią stację, tzn. $K^{\eta,l}$, w stanie $P^{\eta,l}$.

D o w ó d: Dla trajektorii lokalnie optymalnej $P^{\eta,l_2},\ldots,P^{N,l_2^o}$ napiszemy:

$$v^{N,l_2^o} = v^{N,l_2} + \Delta v^2 \tag{6.89}$$

Jeżeli jest spełniony warunek (6.88), to od stanu P^{η,l_1} można zrealizować alokację operacji i monterów wynikającą z lokalnie optymalnej trajektorii $P^{\eta,l_2},\ldots,P^{N,l_2^o}$. W ten sposób otrzymujemy dopuszczalną trajektorię $P^{\eta,l_1},\ldots,P^{N,\mathfrak{L}}$.

Ponieważ $W^{\eta,l_1} = W^{\eta,l_2}$, zatem:

$$v^{N,l_2^o} = v^{N,\mathfrak{L}} \tag{6.90}$$

Lecz dla stanu lokalnie optymalnego P^{N,l_1^o}, z uwagi na $T^{\eta,l_1} \leq T^{\eta,l_2}$, otrzymamy:

$$v^{N,l_1^o} \leq v^{N,\mathfrak{L}} \tag{6.91}$$

Z (6.90) i (6.91) wynika, że stan P^{η,l_1} dominuje nad stanem P^{η,l_2}. □

TWIERDZENIE 6.11

Jeżeli w stanie $P^{\eta-1,\mathfrak{L}}$ spełniony jest warunek (6.74), to przydzielenie operacji ω_n do k-tej stacji, $K^{\eta-1,\mathfrak{L}} < k \leq K$, daje stan niepersperktywiczny.

D o w ó d: Załóżmy, że przydzielając operację ω_n do $K^{\eta-1,\mathfrak{L}}$-tej stacji otrzymamy stan P^{η,l_1}, natomiast przydzielając operację ω_n do k-tej stacji otrzymamy stan P^{η,l_2}. Zauważmy, że:

$$\Omega^{\eta,l_1} = \Omega^{\eta,l_2} \tag{6.92}$$

$$T^{\eta,l_1} < T^{\eta,l_2} \tag{6.93}$$

$$W^{\eta,l_1} \subset W^{\eta,l_2} \tag{6.94}$$

Rozważmy trajektorię lokalnie optymalną $P^{\eta,l_2},\ldots,P^{\eta,l_2^o}$, dla której napiszemy:

$$v^{N,l_2^o} = v^{\eta,l_2} + \Delta v^2 \tag{6.95}$$

przy tym dla stacji $\varkappa$, $\varkappa = K^{\eta-1,\mathfrak{L}} + 1,\ldots,k-1$ otrzymujemy:

$$\Omega_{\varkappa}^{N,l_2^o} = 0 \tag{6.96}$$

Z uwagi na (6.92), (6.93) i (6.94) od stanu P^{η,l_1} można zrealizować alokację operacji i monterów, która wynika z trajektorii lokalnie optymalnej $P^{\eta,l_2},\ldots,P^{N,l_2^o}$. A zatem otrzymamy trajektorię $P^{\eta,l_1},\ldots,P^{N,\lambda}$, dla której napiszemy:

$$v^{N,\lambda} = v^{\eta,l_1} + \Delta v^1 \tag{6.97}$$

Przy uwzględnieniu (6.96) otrzymamy:

$$v^{\eta,l_2} - v^{\eta,l_1} = 1 \tag{6.98a}$$

$$0 \leq \Delta v^1 - \Delta v^2 \leq 1 \tag{6.98b}$$

A zatem z (6.95), (6.96) i (6.98) dochodzimy do:

$$v^{N,l_1^o} \leq v^{N,\lambda} \leq v^{N,l_2^o} \tag{6.99}$$

ponieważ stan $P^{N,\lambda}$ na ogół nie jest stanem lokalnie optymalnym P^{N,l_1^o}. □

Twierdzenie 6.11 daje uzasadnienie dla stosowania podanych wcześniej procedur generowania stanów.

7. ZAKOŃCZENIE

W rozprawie przedstawiono wybrane deterministyczne modele jednowersyjnego procesu montażu. Dla sformułowanych problemów opracowano następujące algorytmy:

- BLM (z dodatkowymi parametrami operacji, z enklawami operacji, z frontami pracy oraz z uwzględnieniem kosztu montażu),
- AO na linii o szeregowych stanowiskach pracy (z ograniczeniami pozycyjnymi, z ograniczeniami czasowymi, z ograniczeniami zakresowymi oraz dla linii asymetrycznej),
- AO na linii o równoległych stanowiskach pracy (dla modelu podstawowego z relacją wykluczania oraz z relacją synchronizacji),
- AOiZ (z uwzględnieniem kontenerów, narzędzi oraz monterów).

Wszystkie algorytmy oparto na metodzie programowania wieloetapowego.

W DODATKU przedstawiono modele i algorytmy sterowania dynamicznym procesem montażu (jednowersyjnym oraz wielowersyjnym). Są to algorytmy symulacyjne oparte na równaniach stanu procesu montażu.

Modele i algorytmy BLM, AO oraz AOiZ dotyczą jednowersyjnego montażu na linii w przedziale $[0,C]$. Modele i algorytmy sterowania montażem pozwalają analizować przebieg procesu, gdy w przedziale $[0,C]$ montowane są obiekty różnych wersji.

Z teoretycznego punktu widzenia w rozdziałach: 2, 3, 4, 5 i 6 rozprawy rozważano problemy alokacji zasobów. BLM stanowi szczególny przypadek alokacji zasobów, w którym dla każdej operacji przydzielane jest jedno stanowisko pracy. Przy tym nie rozróżnia się stacji i stanowisk pracy. AO jest szczególnym przypadkiem alokacji zasobów, w którym każdej operacji przydzielone jest stanowisko pracy na określonej stacji montażowej. Stanowiska pracy stanowią podstawowy zasób w procesie montażu. Uwzględnienie zasobów dodatkowych (kontenerów, narzędzi oraz monterów) prowadzi do ogólnych modeli alokacji zasobów. Z powyższych względów problemy rozważane w rozprawie nazwano ogólnie alokacją operacji i zasobów.

Z praktycznego punktu widzenia w procesie montażu istotne znaczenie ma określenie stanowiska pracy, na którym ma być wykonana każda operacja. Dla montażu statycznego problem taki jest rozwiązywany jednokrotnie w przedziale $[0,C]$. W przypadku montażu dynamicznego problem alokacji operacji jest rozwiązywany dla każdego cyklu. Stąd też przedstawione w DODATKU modele i algorytmy sterowania montażem dotyczą w istocie alokacji operacji.

W rozprawie wprowadzono klasyfikację linii i procesu montażu. Wykorzystanie metody programowania wieloetapowego do konstrukcji wszystkich algorytmów ułatwia ocenę stopnia komplikacji rozwiązywanych problemów - dla wyróżnionych modeli. Poza tym łatwo można zauważyć pewne analogie, np. pomiędzy modelami z dodatkowymi zasobami a modelem uwzględniającym koszt montażu.

Problem AOiZ można uznać za wiodący w dziedzinie sterowania montażem na liniach. Wynika to wprost z charakterystyki ekonomicznej linii montażowej. Wymaganiom stawianym przez linie montażowe podporządkowuje się dostawy detali. Problemy harmonogramowania dostaw detali przedstawiono między innymi w pracach [16, 172, 173, 177, 178, 179, 194].

Rozwiązanie problemu AOiZ dla różnych cykli daje charakterystykę statyczną linii montażowej. W oparciu o taką charakterystykę można sterować montażem na liniach jednostrefowych lub wielostrefowych. Ponadto uwzględniając stanowiska podmontażowe i rozwiązując problemy alokacji operacji dla różnych podzbiorów operacji - otrzymuje się charakterystyki sterowania statycznego. Zagadnienia te były przedstawione w pracach [167, 186, 194].

W rozprawie przedstawiono problemy AOiZ dla montażu jednowersyjnego. Uogólnienia dla montażu wielowersyjnego można konstruować w oparciu o koncepcje przedstawione w pracach [289, 290, 291, 292]. Jednakże dla niektórych modeli (np. montaż przestrzenny lub montaż na liniach z równoległymi stanowiskami pracy) pojawiają się wówczas dodatkowe problemy, związane z kolejnością montażu obiektów różnych wersji.

Uogólnienia deterministycznych modeli i algorytmów przedstawionych w rozprawie, na przypadek probabilistyczny, można opracować wykorzystując koncepcje z pracy [116]. Metoda programowania wieloetapowego pozwala skonstruować odpowiednie algorytmy.

Metoda programowania wieloetapowego pozwala uzyskać rozwiązanie optymalne lub przybliżone z oszacowaniem dokładności. Zajętość pamięci komputerowej można ograniczyć do jednej trajektorii - stąd też istotne znaczenie ma jedynie czas obliczeń. W przypadku wykorzystania limitu czasu obliczeń otrzymuje się rozwiązanie przybliżone z oszacowaniem dokładności.

W rozprawie przedstawiono różne elementy mające wpływ na efektywność algorytmów programowania wieloetapowego. Najwięcej uwagi poświęcono regułom eliminacji stanów nieperspektywicznych, przytaczając liczne twierdzenia. Konstrukcja innych twierdzeń o eliminacji stanów nieperspektywicznych jest sprawę otwartą.

W rozprawie przedstawiono pewne wyspecyfikowane modele AOiZ. Perspektywę dalszych prac w tym kierunku (mogących mieć znaczenie praktyczne) stanowią modele kompleksowe. Modele takie są jednak znacznie bardziej złożone.

W związku z tendencją do rozwijania montażu wielowersyjnego na liniach z robotami przemysłowymi, istotnego znaczenia nabiera problematyka sterowania montażem dynamicznym. W tym przypadku w modelach montażu należy uwzględnić automatyczne transportery i magazyny dla detali.

DODATEK

MODELE MATEMATYCZNE I ALGORYTMY STEROWANIA MONTAŻEM DYNAMICZNYM

Streszczenie. W dodatku rozważana jest szeregowa linia montażowa oraz proces montażu - z dokładnością do operacji (detalu). Analizowany jest proces montażu w kolejnych cyklach na każdym stanowisku pracy (stacji montażowej).

Dla opisu procesu wprowadzono definicję stanu montażu. Wyróżniono: stan początkowy i końcowy oraz lewostronny i prawostronny warunek brzegowy. Ponadto przyjęto klasyfikację procesu montażu, rozróżniając proces: statyczny, dynamiczny oraz przejściowy. W analogiczny sposób zdefiniowano sterowanie montażem oraz przyjęto klasyfikację sterowania wyróżniając sterowanie: statyczne, dynamiczne oraz przejściowe. W oparciu o definicje stanu i sterowania wyprowadzono równania stanu procesu montażu. Równania te interpretuję zależności logiczne w montażu na kolejnych stanowiskach pracy w kolejnych cyklach.

W dodatku wyróżniono procesy montażu jednowersyjnego oraz wielowersyjnego. Szczegółowo przeanalizowano proces jednowersyjny. Wyprowadzone wzory uogólniono dla montażu wielowersyjnego. Rozważane modele matematyczne są oparte na równaniach stanu. Algorytmy sterowania wykorzystuję heurystyczne reguły decyzyjne oraz symulację procesu montażu.

Dla procesu jednowersyjnego przedstawiono deterministyczny oraz probabilistyczny model montażu. Ponadto rozważono deterministyczny model montażu i zapasów detali. Dla modelu deterministycznego przeanalizowano sterowanie: statyczne, pseudostatyczne, pseudodynamiczne oraz dynamiczne. Pokazano różne algorytmy sterowania dynamicznego.

W modelu wielowersyjnym istotne znaczenie ma lewostronny warunek brzegowy (ponieważ jest sterowalny). Interpretuje on kolejność obiektów (różnych wersji) wprowadzanych na linię montażową. Dla deterministycznego modelu montażu wielowersyjnego przedstawiono sterowanie dynamiczne.

Rozważane w dodatku modele matematyczne i algorytmy sterowania, oparte na równaniach stanu, są modelami typu symulacyjnego. Korzystając z nich można zatem wyznaczyć racjonalne sterowanie, bez gwarancji optymalności. Przedstawione modele mogą być wykorzystane na liniach montażowych z robotami przemysłowymi.

SPIS OZNACZEŃ

Ω - zbiór operacji

K - liczba stacji linii montażowej

k - numer stacji linii montażowej

L - liczba cykli (lub obiektów) w procesie montażu

l - numer cyklu w procesie montażu

M - liczba stanowisk pracy na linii montażowej

m - numer stanowiska pracy na linii montażowej

$\mathcal{K}_m$ - numer stacji, na której znajduje się m-te stanowisko pracy

N - liczba operacji montażowych

n - numer operacji montażowej

S - liczba wersji montowanych obiektów

s - numer wersji montowanego obiektu

Ω^s - zbiór operacji s-tej wersji

N^s - liczba operacji s-tej wersji

ω_n - n-ta operacja

Γ - macierz relacji kolejnościowej

$\gamma_{\nu,n}$ - element macierzy Γ

Γ^s - macierz relacji kolejnościowej s-tej wersji

τ_n - deterministyczny czas wykonania operacji ω_n

c - czas cyklu montażu

c^l - l-ty czas cyklu montażu

C - okres (horyzont) montażu

ζ_n - losowy czas wykonania operacji ω_n

A - wektor alokacji operacji na stanowiskach montażowych

a_n - numer stacji, na której jest wykonywana operacja ω_n

x^l - stan montażu na linii po l-tym cyklu

x^l_k - stan montażu na k-tej stacji po l-tym cyklu

$x^l_{k,n}$ - współrzędne wektora x^l_k (lub macierzy x^l)

x^0 - warunek początkowy montażu

x^L - warunek końcowy montażu

x^l_0 - lewostronny warunek brzegowy montażu

x^l_{K+1} - prawostronny warunek brzegowy montażu

A^l_m - sterowanie jednowersyjnym procesem montażu w l-tym cyklu na m-tym stanowisku pracy

$a^l_{m,n}$ - współrzędna wektora A^l_m

U^l - sterowanie procesem montażu na linii w l-tym cyklu

U^l_k - sterowanie procesem montażu w l-tym cyklu na k-tej stacji

$u^l_{k,n}$ - współrzędne wektora U^l_k

h - logiczna funkcja transformacji współrzędnej stanu $x^l_{k,n}$

H_k - logiczna funkcja transformacji stanu x^l_k

H - logiczna funkcja transformacji stanu x^l

Δx^l - wektor kompletności montażu obiektu schodzącego z linii w l-tym cyklu

$f_n(t)$ - funkcja gęstości rozkładu prawdopodobieństwa czasu wykonania operacji ω_n

$\sigma^l_{k,n-1}$ - suma losowych czasów operacji przydzielonych w l-tym cyklu do k-tej stacji przed operacją ω_n

χ^l - wektor pseudolosowych czasów operacji w l-tym cyklu

D^l - wektor dostaw detali w l-tym cyklu

d^l_n - dostawa n-tego detalu w l-tym cyklu

Z^l - stan zapasów detali po l-tym cyklu

z^l_n - zapas n-tego detalu po l-tym cyklu

F - funkcja transformacji zapasów detali

T^l - chwila uruchomienia transportera po l-tym cyklu

g_n - koszt wykonania operacji ω_n na linii montażowej

g'_n - dodatkowy koszt montażu operacji ω_n poza linię

$q^l_{k,n}$ - straty wynikające z braku wykonania operacji ω_n w l-tym cyklu na k-tej stacji

q^l_k - strata w l-tym cyklu na k-tej stacji

q_k - strata przejściowego procesu montażu w k-tej stacji

R_k - zadanie na k-tej stacji

$r_{k,n}$ - współrzędne wektora R_k

τ_k - czas wykonania zadania R_k

$\bar{U}^l$ - sterowanie statyczne

$\tilde{U}^l$ - składowa zmienna sterowania pseudodynamicznego

Ω_k - podzbiór operacji, które mogą być wykonane na k-tej stacji

c^l_k - czas wykonania najmniejszego podzbioru operacji na k-tej stacji

W - macierz przyporządkowania operacji ω_n do s-tej wersji

$w_{n,s}$ - element macierzy W

$u^{l}_{k,o}$ - numer wersji obiektu, który w l-tym cyklu znajduje się na k-tej stacji

$\hat{U}$ - sterowanie quasi-dynamiczne

1. WPROWADZENIE

Rozważane w rozprawie problemy alokacji zadań i zasobów mają podstawowe znaczenie dla montażu seryjnego. Jednakże w seryjnym procesie montażu występują pewne zaburzenia - na styku dwóch kolejnych okresów, dwóch kolejnych serii lub w przypadku zakłóceń (np. braku niektórych detali, losowych czasów operacji itp.). Efektem tych zaburzeń są niekompletnie zmontowane obiekty. Zagadnienia te będą przedmiotem rozważań w niniejszym dodatku.

W zmiennym procesie montażu zbierana jest (sporadycznie) duża liczba informacji [230]. Ponadto w zależności od charakteru zaburzeń montażu stosowane są różne modele dla sterowania [139, 146]. W ogólnym przypadku sterowanie to obejmuje nie tylko montaż, lecz również synchronizację dostaw detali [241].

Analiza zaburzeń w procesie montażu obiektów różnych wersji sprowadzana jest do tzw. problemu kolejnościowego [93, 128, 129, 314, 289, 290, 291, 292, 179, 205, 178]. Problem ten polega na określeniu najlepszej (w sensie pewnego kryterium) kolejności montażu obiektów. W opublikowanych dotąd pracach z tego zakresu proces montażu był analizowany z dokładnością do podzbioru operacji na stanowisku pracy.

Analiza zaburzeń procesu montażu na styku dwóch kolejnych okresów była przedstawiona w pracach [43, 44, 145, 271, 272, 176, 180, 194]. Istota zagadnienia wynika z różnej alokacji operacji w dwóch kolejnych okresach. A zatem w tym przypadku proces montażu jest analizowany z dokładnością do pojedynczej operacji.

Synchronizacja dostaw detali była przedstawiona w pracach [149, 241, 119, 154, 221, 164]. Ponieważ detal jest związany z operacją, zatem również w tym przypadku proces montażu jest analizowany z dokładnością do operacji.

Analiza zakłóceń procesu montażu, wynikających z losowych czasów operacji, była przedmiotem prac [228, 181]. Określenie kompletności montażu obiektów wymaga przeprowadzenia analizy z dokładnością do operacji.

Ewidencja (identyfikacja) oraz sterowanie procesu montażu z dokładnością do operacji wymagają określenia podzbiorów operacji, jakie muszą być wykonane w każdym cyklu na każdym stanowisku pracy. Realizacja montażu w takich warunkach jest możliwa, jeżeli operacje są wykonywane przez roboty przemysłowe [15, 49, 231, 234, 140, 143].

W przypadku gdy montaż jest realizowany przez roboty przemysłowe, program pracy robota może być różny w kolejnych cyklach. Dla grupy współpracujących robotów istotne znaczenie ma wybór programu pracy robota na każdym stanowisku pracy w każdym cyklu. Zagadnienia te były przedstawione w pracach [198, 215, 221].

W niniejszym dodatku zostaną przedstawione modele matematyczne i algorytmy sterowania montażem dynamicznym. Będą to modele i algorytmy symulacyjne, umożliwiające wybór racjonalnego wariantu sterowania.

2. ZAŁOŻENIA

W procesie montażu wyróżniamy obiekty, które formalnie są zbiorami określonych detali. Jeżeli każdy montowany obiekt składa się z detali tego samego typu, to montaż jest jednowersyjny. W przypadku przeciwnym montaż jest wielowersyjny. Rozważymy najpierw montaż jednowersyjny, a następnie otrzymane formuły uogólnimy dla montażu wielowersyjnego.

Model matematyczny procesu montażu zostanie przedstawiony w postaci równań stanu, które interpretują zależności logiczne. Na podstawie tych równań można symulować deterministyczny lub probabilistyczny proces montażu. Ponadto można również uwzględnić ograniczenia wynikające z dostaw detali do stanowisk pracy.

Celem modelowania i symulacji procesu montażu jest analiza efektów zmian i zakłóceń montażu. Ponadto metodą symulacji wyznacza się sterowanie montażem.

Do dalszych rozważań założymy, że linia montażowa składa się z transportera obiektów, wzdłuż którego rozmieszczone są stacje montażowe. Na pewnych stacjach montażowych mogą być zlokalizowane stanowiska pracy. Rozważymy linię szeregową, dla której na stacji znajduje się najwyżej jedno stanowisko pracy. Przyjmujemy, że linia montażowa składa się z K stacji (ponumerowanych: 1,...,k,...,K, zgodnie z kierunkiem ruchu obiektów). Punkt instalowania chassis znajduje się bezpośrednio przed pierwszą stacją montażową (tzn. na stacji zerowej). Ponadto punkt kontroli kompletności montażu znajduje się bezpośrednio po K-tej stacji (tzn. na K+1-szej stacji).

W trakcie montażu obiekt przesuwa się ruchem jednostajnym wzdłuż stanowiska pracy - lub pozostaje w spoczynku na środku tego stanowiska, a po zakończeniu operacji przesuwa się na następne stanowisko. W pierwszym przypadku operacje są wykonywane przez monterów, natomiast w drugim - przez roboty przemysłowe. W obu przypadkach obiekt jest dostępny na stanowisku pracy w ciągu cyklu c. Dalsze rozważania będą obejmowały oba przypadki.

Proces montażu obiektu polega na wykonaniu wzdłuż linii, na kolejnych stanowiskach pracy, zbioru Ω operacji ω_n, n = 1,...,N, gdzie: N - licz-

ba operacji. Czas pobytu obiektu na linii (przy stałym cyklu) wynosi K.c. W dalczym ciągu rozważymy montaż na linii w okresie [0,C]. Dla montażu przy stałym cyklu C = L.c, gdzie: L - liczba obiektów zmontowanych w okresie [0,C]. Kolejne cykle ponumerujemy: l = 1,...,L.

W procesie montażu musi być respektowana relacja kolejności dana macierzą $\Gamma = [\gamma_{\nu,n}]$, $\nu = 1,...,N$ oraz $n = 1,...,N$. Jeżeli $\gamma_{\nu,n} = 1$, to operacja ω_ν jest bezpośrednim poprzednikiem ω_n (w przypadku przeciwnym $\gamma_{\nu,n} = 0$). Załóżymy, że dane są deterministyczne (ϑ_n) lub probabilistyczne (ξ_n) czasy operacji ω_n, $n = 1,...,N$.

Do dalszej analizy załóżymy, że numeracja operacji spełnia warunek:

$$\bigwedge_{\nu}\bigwedge_{n} (\gamma_{\nu,n} = 1) \Longrightarrow (\nu < n) \qquad (1)$$

A zatem następnik ma wyższy numer niż poprzednik. Warunek (1) będzie miał istotne znaczenie w obliczeniach prowadzonych na podstawie równań stanu. Warunek ten nie ogranicza ogólności rozważań.

Przyjmiemy, że w rezultacie rozwiązania problemu alokacji operacji dana jest lokalizacja operacji na stacjach montażowych, tzn. dany jest wektor:

$$A = [a_n]_{n=1,...,N} \qquad (2)$$

gdzie: a_n - numer stacji montażowej, do której przydzielono operację ω_n.

Z wektora (2) można łatwo określić numer stacji $\mathcal{S}_m$ ($m = 1,...,M$ oraz $M \leq K$), na której zlokalizowano m-te stanowisko pracy.

3. PODSTAWOWE OKREŚLENIA

Rozważmy jednowersyjny proces montażu. W jednowersyjnym procesie montażu w każdym cyklu na każdym stanowisku pracy mogą być wykonywane jednakowe lub różne zbiory operacji. Balansowanie linii montażowej lub alokacja operacji określa zbiory operacji i ich lokalizację na stanowiskach pracy. Jednakże procedury te nie uwzględniają stanu zaawansowania montażu obiektów znajdujących się na linii. Stąd w procesie montażu jednowersyjnego w kolejnych cyklach na stanowisku pracy mogą być wykonywane różne operacje.

3.1. STAN PROCESU MONTAŻU

Załóżmy, że proces montażu będzie analizowany z dokładnością do operacji ω_n ($n = 1,...,N$). Elementarną jednostkę przestrzennego podziału linii jest k-ta stacja montażowa ($k = 1,...,K$). Ponadto wprowadzamy dyskretyzację czasu wyróżniając cykle c^l ($l = 1,...,L$). A zatem proces montażu na linii jest zależny od czasu (cyklu c^l) oraz przestrzeni (k-tej stacji).

Z punktu widzenia ewidencji (identyfikacji) procesu montażu istotne znaczenie mają chwile zakończenia kolejnych cykli. Można wówczas sprawdzić, które operacje zostały wykonane na k-tej stacji, a zatem jakie detale są zamontowane do obiektu, który wchodzi na k+1-szą stację.

Definicja 1a

Stan jednowersyjnego procesu montażu obiektu na k-tej stacji (k=1,...,K) po l-tym cyklu (l = 1,...,L) jest wektorem:

$$x_k^l = \left[x_{k,n}^l\right]_{n=1,...,N} \qquad (3)$$

Elementy tego wektora określamy następująco:

$$x_{k,n}^l = \begin{cases} 1, \text{ jeśli n-ty detal jest zamontowany w obiekcie znajdującym się na k-tej stacji po l-tym cyklu} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \qquad (3a)$$

Definicja 1b

Stan jednowersyjnego procesu montażu na linii po l-tym (l = 1,...,L) cyklu jest macierzę:

$$x^l = \left[x_k^l\right] = \left[x_{k,n}^l\right]_{\substack{k=1,...,K \\ n=1,...,N}} \qquad (4)$$

W uzupełnieniu definicji 1 przyjmiemy:

- warunek początkowy jednowersyjnego procesu montażu na linii

$$x^0 = \left[x_k^0\right] = \left[x_{k,n}^0\right] \qquad (5)$$

- warunek końcowy jednowersyjnego procesu montażu na linii

$$x^{L+1} = \left[x_k^{L+1}\right] = \left[x_{k,n}^{L+1}\right] \qquad (6)$$

- lewostronny warunek brzegowy jednowersyjnego procesu montażu na linii

$$x_0^l = 0 \quad \bigwedge_{0 \leq l \leq L} \qquad (7)$$

- prawostronny warunek brzegowy jednowersyjnego procesu montażu na linii

$$x^l_{K+1} = \left[x^l_{K+1,n}\right] \quad \bigwedge_{1 \le l \le L} \tag{8}$$

W szczególnym przypadku (mającym istotne znaczenie praktyczne) prawostronny warunek brzegowy jest warunkiem montażu kompletnego, czyli:

$$x^l_{K+1} = 1 \quad \bigwedge_{1 \le l \le L} \tag{8a}$$

Lewostronny warunek brzegowy przedstawia sytuację, w której obiekt wchodzący na pierwszą stację nie ma zamontowanego żadnego detalu. Stan początkowy x^0 określa stopień zaawansowania montażu obiektów znajdujących się na linii przed rozpoczęciem montażu w okresie $[0,c]$. Z kolei stan końcowy x^L określa stopień zaawansowania montażu obiektów na linii po zakończeniu okresu $[0,c]$.

W oparciu o definicję stanu 1 można wprowadzić następującą klasyfikację procesu montażu:

- Proces statyczny

Proces montażu nazwiemy statycznym, jeżeli jest spełniony warunek:

$$\bigwedge_{1 \le l \le L} x^l = x^{l-1} \tag{9}$$

- Proces dynamiczny

Proces montażu nazwiemy dynamicznym, jeżeli warunek (9) nie jest spełniony, tzn.:

$$\bigvee_{1 \le l \le L} x^l \neq x^{l-1} \tag{10}$$

- Proces przejściowy

W montażu jednowersyjnym wyróżnimy szczególny przypadek procesu dynamicznego, który nazwiemy procesem przejściowym. Proces przejściowy spełnia warunek:

$$\bigvee_{1 \le \lambda \le K} \bigwedge_{K < l \le L} (x^\lambda \neq x^{\lambda-1}) \wedge (x^l = x^{l-1}) \tag{11}$$

Proces przejściowy jest dynamiczny do K-tego cyklu, a następnie jest statyczny. Jest on charakterystyczny dla montażu jednowersyjnego przy przejściu pomiędzy dwoma kolejnymi okresami pracy (przy odmiennej alokacji operacji). Proces przejściowy zanika po zejściu z linii obiektów montowanych w dwóch różnych (kolejnych) okresach pracy.

3.2. STEROWANIE PROCESEM MONTAŻU

Sterowanie procesem montażu polega na określeniu podzbiorów operacji, które powinny być wykonane na m-tym stanowisku pracy ($m = 1,\dots,M$) w każdym cyklu l ($l = 1,\dots,L$).

Definicja 2a

Sterowanie jednowersyjnym procesem montażu w l-tym cyklu ($l = 1,\dots,L$) na m-tym stanowisku pracy ($m = 1,\dots,M$) jest wektorem:

$$A^l_m = \left[a^l_{m,n}\right]_{n=1,\dots,N} \tag{12}$$

Elementy tego wektora określamy następująco:

$$a^l_{m,n} = \begin{cases} 1, \text{ jeśli w l-tym cyklu na m-tym stanowisku pracy ma być wykonana n-ta operacja} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{12a}$$

Definicja 2b

Sterowanie jednowersyjnym procesem montażu w l-tym cyklu ($l = 1,\dots,L$) na k-tej stacji ($k = 1,\dots,K$) jest wektorem:

$$U^l_k = \left[u^l_{k,n}\right]_{n=1,\dots,N} \tag{13}$$

przy tym istotne znaczenie ma U^l_k dla:

$$k = \varkappa_m, \quad m = 1,\dots,M \tag{13a}$$

Sterowanie U^l_k dla $k \neq \varkappa_m$, $m = 1,\dots,M$, jest wektorem zerowym.

Elementy wektora (13) są określone następująco:

$$u^l_{k,n} = \begin{cases} 1, \text{ jeśli w l-tym cyklu na k-tej stacji należy wykonać n-tę operację} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{13b}$$

Znając lokalizację stanowisk pracy na stacjach montażowych można wyznaczyć U^l_k na podstawie A^l_k. Tak więc otrzymujemy:

$$u^l_{k,n} = \begin{cases} a^l_{m,n}, & \text{jeśli } \mathcal{M}_m = k \\ 0, & \text{w przypadku przeciwnym.} \end{cases} \tag{14}$$

Dla procesu statycznego sterowanie A^l można wyznaczyć z (2), bowiem:

$$a^l_{k,n} = \begin{cases} 1, \text{ jeśli } a_n = m \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{15}$$

Definicja 2c

Sterowanie jednowersyjnym procesem montażu na linii w l-tym cyklu ($l = 1,\dots,L$) jest macierzą:

$$U^l = \left[U^l_k\right] = \left[u^l_{k,n}\right]_{\substack{n=1,\dots,N \\ k=1,\dots,K}} \tag{16}$$

Analogicznie jak proces montażu można sklasyfikować sterowanie procesem. A zatem wyróżnimy:

- Sterowanie statyczne

Sterowanie procesem montażu nazwiemy statycznym, jeżeli jest spełniony warunek:

$$\bigwedge_{1 < l \leq L} U^l = U^{l-1} \tag{17}$$

- Sterowanie dynamiczne

Sterowanie procesem montażu nazwiemy dynamicznym, jeżeli warunek (17) nie jest spełniony, czyli:

$$\bigvee_{1 < l \leq L} U^l \neq U^{l-1} \tag{18}$$

- Sterowanie przejściowe

Sterowanie przejściowe jest szczególnym przypadkiem sterowania dynamicznego. Sterowanie to spełnia warunek:

$$\bigvee_{1 < \lambda \leq K} \bigwedge_{K < k \leq L} (U^\lambda \neq U^{\lambda-1}) \wedge (U^k = U^{k-1}) \tag{19}$$

Zauważmy, że sterowanie przejściowe ma charakter dynamiczny do K-tego cyklu oraz statyczny - po K-tym cyklu. Sterowanie statyczne będziemy oznaczali przez $\bar{U}^l$, natomiast sterowanie dynamiczne przez $\overline{\overline{U}}^l$.

Sterowanie statyczne $\bar{U}^l$ wynika z alokacji operacji na linii montażowej. Jeżeli $K < L$, to sterowanie statyczne prowadzi do statycznego procesu montażu począwszy od K-tego cyklu. Dla $l < K$ na linii występuje proces przejściowy.

4. RÓWNANIA STANU

W oparciu o wprowadzone założenia i określenia przedstawione zostaną modele matematyczne jednowersyjnego procesu montażu.

W deterministycznym modelu montażu przyjmujemy, że dany jest normatywny czas t_n wykonania operacji ω_n, $n = 1,\dots,N$. Aby wyprowadzić równania stanu rozważymy proces montażu na k-tej stacji w l-tym cyklu. Zatem dany jest stan:

$$x^{l-1}_{k-1} \quad \begin{matrix} l=1,\dots,L \\ k=1,\dots,K \end{matrix}$$

oraz sterowanie:

$$U^l_k \quad \begin{matrix} l=1,\dots,L \\ k=1,\dots,K \end{matrix}$$

Sterowanie to ma istotne znaczenie tylko na stanowiskach pracy, a zatem założymy, że $k = \mathcal{M}_m$, $m = 1,\dots,M$.
Przejście do stanu:

$$x^l_k \quad \begin{matrix} l=1,\dots,L \\ k=1,\dots,K \end{matrix}$$

wymaga wyznaczenia wszystkich współrzędnych $x^l_{k,n}$ na podstawie x^{l-1}_{k-1} i U^l_k. Ponadto dane są: stan początkowy X^0 oraz lewostronny warunek brzegowy x^{l-}_0. Załóżmy, że współrzędne $x^l_{k,n}$ są wyznaczane kolejno od 1 do N. A zatem zgodnie z (1) najpierw wyznaczamy współrzędną $x^l_{k,\nu}$ dla poprzednika ω_ν, a później $x^l_{k,n}$ dla następnika ω_n.

Analizując możliwość i konieczność wykonania w l-tym cyklu na k-tej stacji operacji ω_n, bierzemy pod uwagę: współrzędną stanu $x^{l-1}_{k-1,n}$, sterowanie $u^l_{k,n}$ oraz sprawdzamy, czy zostały wykonane poprzedniki ω_ν operacji ω_n. Mogą przy tym wystąpić sytuacje, w których operacja ω_n nie ma poprzedników, lub jej poprzedniki ω_ν są wykonane lub nie.

A zatem wyróżnimy następujące przypadki:

I. Jeżeli operacja ω_n została wykonana przed k-tę stacją montażową, to:

$$\bigwedge_n \; (x^{l-1}_{k-1,n} = 1) \Longrightarrow (x^{l}_{k,n} = 1) \tag{20}$$

II. Jeżeli operacja ω_n nie została wykonana przed k-tę stacją montażową i nie ma być wykonana na k-tej stacji, to:

$$\bigwedge_n \; (x^{l-1}_{k-1,n} = 0) \wedge (u^{l}_{k,n} = 0) \Longrightarrow (x^{l}_{k,n} = 0) \tag{21}$$

III. Jeżeli operacja ω_n (która nie została wykonana przed k-tę stację i ma być wykonana na k-tej stacji) nie ma bezpośrednich poprzedników, to:

$$\bigwedge_n \; \bigwedge_{\vartheta < n} \; (x^{l-1}_{k-1,n} = 0) \wedge (u^{l}_{k,n} = 1) \wedge (\varphi_{\vartheta,n} = 0) \Longrightarrow (x^{l}_{k,n} = 1) \tag{22}$$

IV. Jeżeli wszystkie bezpośrednie poprzedniki ω_ϑ operacji ω_n (która nie została wykonana przed k-tę stację i ma być wykonana na k-tej stacji) są wykonane, to:

$$\bigwedge_n \; \bigwedge_{\vartheta < n} \; (x^{l-1}_{k-1,n}=0) \wedge (u^{l}_{k,n}=1) \wedge \left[(\varphi_{\vartheta,n}=1) \Longrightarrow (x^{l}_{k,\vartheta}=1) \right] \Longrightarrow (x^{l}_{k,n}=1) \tag{23}$$

V. Jeżeli pewien bezpośredni poprzednik ω_ϑ operacji ω_n (która nie została wykonana przed k-tę stację i ma być wykonana na k-tej stacji) nie został wykonany, to:

$$\bigwedge_n \; \bigvee_{\vartheta < n} \; (x^{l-1}_{k-1,n}=0) \wedge (u^{l}_{k,n}=1) \wedge \left[(\varphi_{\vartheta,n}=1) \Longrightarrow (x^{l}_{k,\vartheta}=0) \right] \Longrightarrow (x^{l}_{k,n}=0) \tag{24}$$

Warto podkreślić, że sprawdzenia wykonania bezpośrednich poprzedników ω_ϑ operacji ω_n dokonuje się na podstawie początkowych n-1 współrzędnych stanu X^{l}_{k}. Jest to istotne, ponieważ bezpośrednie poprzedniki operacji ω_n mogą być wykonane również na k-tej stacji (wówczas $x^{l-1}_{k-1,\vartheta}=0$, lecz $x^{l}_{k,\vartheta}=1$). Z drugiej strony, jeżeli ϑ-ty detal zamontowano w l-1-szym cyklu na k-1-szej stacji (lub wcześniej), to jest on również zamontowany w obiekcie na k-tej stacji (tzn. jeżeli $x^{l-1}_{k-1,\vartheta}=1$, to $x^{l}_{k,\vartheta}=1$).

Tak więc wyznaczając stan X^{l}_{k} korzystamy z niektórych współrzędnych tego stanu. Jednakże wzory dla wyznaczania współrzędnych $x^{l}_{k,n}$ nie są uwi-

kłane, gdyż zawierają jedynie $x^{l}_{k,\vartheta}$, dla $\vartheta < n$, natomiast numeracja operacji spełnia warunek (1).

Z warunków (20÷24) wynika, że współrzędne stanu procesu montażu na k-tej stacji wyznaczamy z następujących równań rekurencyjnych:

$$\left.\begin{aligned}
&x^{l}_{k,1} = h(x^{l-1}_{k-1,1},\, u^{l}_{k,1}) \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&x^{l}_{k,n} = h(x^{l-1}_{k-1,n},\, x^{l}_{k,1},\ldots,x^{l}_{k,\vartheta},\ldots,x^{l}_{k,n-1},\, u^{l}_{k,n}) \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&x^{l}_{k,N} = h(x^{l-1}_{k-1,N},\, x^{l}_{k,1},\ldots,x^{l}_{k,\vartheta},\ldots,x^{l}_{k,N-1},\, u^{l}_{k,N})
\end{aligned}\right\} \tag{25}$$

gdzie: h - funkcja logiczna.

Przebieg obliczeń funkcji logicznej h pokazano na rys. D.1.

Z równań tych wynika, że wielkościami niezależnymi są x^{l-1}_{k-1} oraz U^{l}_{k}.

A zatem równania stanu procesu montażu na kolejnych stacjach możemy napisać w postaci:

$$\left.\begin{aligned}
&x^{l}_{1} = H_1(X^{l-1}_{0},\, U^{l}_{1}) \\
&\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots \\
&x^{l}_{k} = H_k(x^{l-1}_{k-1},\, U^{l}_{k}) \\
&\cdots\cdots\cdots\cdots\cdots \\
&\cdots\cdots\cdots\cdots\cdots \\
&x^{l}_{K} = H_K(x^{l-1}_{K-1},\, U^{l}_{K})
\end{aligned}\right\} \tag{26}$$

gdzie: H_k - funkcja logiczna dla k-tej stacji.

Funkcję H_k definiujemy za pomocą (25), jeżeli $k = \mathcal{K}_m$, $m = 1,\ldots,M$, tzn. na k-tej stacji znajduje się m-te stanowisko pracy. W przypadku przeciwnym otrzymujemy wprost:

$$x^{l}_{k} = x^{l-1}_{k-1} \qquad l=1,\ldots,L \tag{27}$$

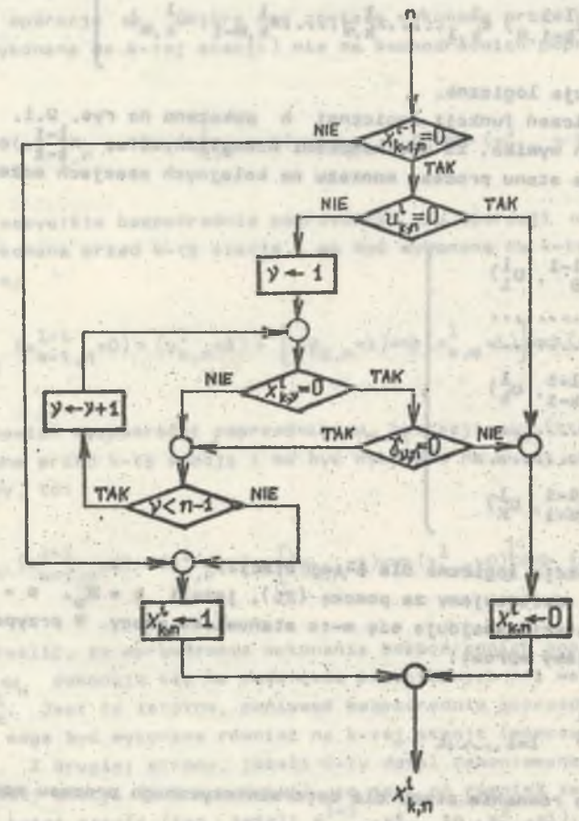
Tak więc ogólne równanie stanu dla deterministycznego procesu montażu ma postać:

$$x^l = H(x^{l-1}, u^l) \qquad (28)$$

gdzie: H - funkcja logiczna.

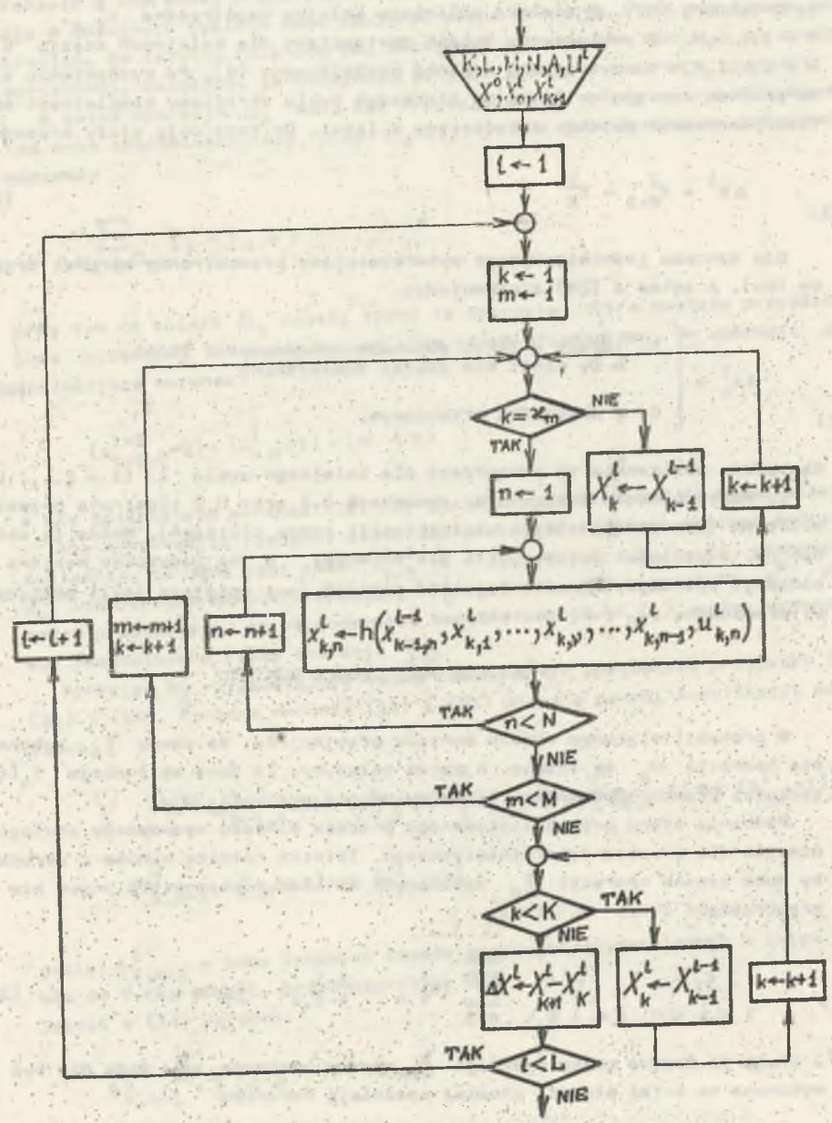
Zgodnie z przedstawionymi równaniami stanu obliczenia są prowadzone dla l cykli (l = 1,...,L), k stacji (k = 1,...,K) oraz n współrzędnych (n = 1,...,N). Ogólny schemat blokowy przebiegu obliczeń dla deterministycznego procesu montażu pokazano na rys. D.2. Na rysunku tym funkcję h oznaczono jako procedurę.

W pierwszym kroku obliczeniowym przyjmujemy l=1. Obliczenia rozpoczynamy od k=1, korzystając z lewostronnego warunku brzegowego (7). Jeżeli $\aleph_1 = 1$, to stosując funkcję H_1 wykorzystujemy dla kolejnych n



Rys. D.1. Schemat blokowy obliczeń dla funkcji logicznej h

Fig. D.1. Block diagram for computation of the logical function h



Rys. D.2. Ogólny schemat blokowy obliczeń dla deterministycznego procesu montażu

Fig. D.2. General block diagram of computation for deterministic assembly process

$(n = 1,\ldots,N)$ funkcję h. W przypadku przeciwnym (gdy $\mathcal{H}_1 > 1$), stosujemy procedurę (27). W efekcie obliczamy kolejne współrzędne $x^1_{1,n}$, $n = 1,\ldots,N$. W analogiczny sposób postępujemy dla kolejnych stacji k, $k = 2,\ldots,K$, korzystając z warunku początkowego (5). Po wyznaczeniu stanu procesu montażu na linii po pierwszym cyklu określamy kompletność montażu pierwszego obiektu schodzącego z linii. Do tego celu służy procedura:

$$\Delta x^1 = x^1_{K+1} - x^1_K \tag{29}$$

Dla montażu jednowrzeciennego wykorzystujemy prawostronny warunek brzegowy (8a). A zatem z (29) otrzymujemy:

$$\Delta x^1_n = \begin{cases} 1, \text{ jeżeli w l-tym obiekcie schodzącym z linii,} \\ \quad n\text{-ty detal nie został zamontowany} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{29a}$$

Następnie obliczenia są powtarzane dla kolejnego cyklu l $(l = 1,\ldots,L)$.

Schematy blokowe pokazane na rysunkach D.1 oraz D.2 ilustrują przebieg obliczeń (nie uwzględniając minimalizacji czasu obliczeń). Można to osiągnąć np. określając zakres pętli dla zmiennej n na podstawie wektora A alokacji operacji. Ponadto zajętość pamięci jest mniejsza jeśli obliczenia są prowadzone od $k=K$ do $k=1$.

5. PROBABILISTYCZNY MODEL MONTAŻU

W probabilistycznym modelu montażu przyjmujemy, że czasy ξ_n wykonania operacji ω_n są losowe. A zatem założymy, że dane są funkcje $f_n(t)$ gęstości prawdopodobieństwa czasu wykonania operacji ω_n.

Równania stanu probabilistycznego procesu montażu wyznaczamy analogicznie jak dla procesu deterministycznego. Istotna różnica wynika z warunku, by suma czasów operacji ξ_n wykonanych na każdym stanowisku pracy nie przekraczała cyklu c, tzn.

$$\bigwedge_{1 \le k \le K} \bigwedge_{1 \le l \le L} \sum_{n=1}^{n=N} (x^1_{k,n} - x^{l-1}_{k-1,n}) \cdot \xi_n \le c \tag{30}$$

Z uwagi na losowe czasy operacji ξ_n pewne operacje ω_n mogą nie być wykonane na k-tej stacji, chociaż spełniają warunek:

$$\bigwedge_{\vartheta < n} (x^{l-1}_{k-1,n}=0) \wedge (u^l_{k,n}=1) \wedge \left[(\vartheta_{\vartheta,n}=1) \Longrightarrow (x^l_{k,\vartheta}=1)\right] \tag{31}$$

Ponadto z tej samej przyczyny podzbiór operacji wykonywanych na k-tej stacji w kolejnych cyklach może zawierać różne operacje. Poza tym trzeba podkreślić, że istotną rolę odgrywa kolejność wykonywania operacji na stacji montażowej. Założymy, że kolejność ta jest zgodna z numeracją operacji.

A zatem operacja ω_n może być przydzielona do k-tej stacji montażowej (na m-te stanowisko pracy, przy $\mathcal{H}_m=k$), jeżeli jest spełniony dodatkowy warunek:

$$\sum_{\omega_i \in \Omega_k} \xi_i + \xi_n \le c \tag{32}$$

przy tym do zbioru Ω_k należą tylko te operacje, które zostały przydzielone wcześniej do k-tej stacji. Tym samym przyjmujemy, że operacja ω_ϑ spełniająca warunek:

$$(x^{l-1}_{k-1,\vartheta}=0) \wedge (u^l_{k,\vartheta}=1) \wedge (\vartheta < n) \tag{33}$$

i nie spełniająca warunku (32) nie zostanie przydzielona do k-tej stacji.

Aby wyprowadzić równanie stanu dla probabilistycznego procesu montażu, zakładamy, że dany jest stan x^{l-1}_{k-1} $(l = 1,\ldots,L,\ k = 1,\ldots,K)$, sterowanie u^l_k, lewostronny warunek brzegowy x^l_0 oraz warunek początkowy x^0_k. Ponadto przyjmujemy, że na podstawie danych funkcji $f_n(t)$ można wygenerować pseudolosowe czasy operacji ξ_n.

Funkcja h_ξ wyznaczająca współrzędne $x^l_{k,n}$ uwzględnia przypadki (2D) (21) i (24). Ponadto warunki (22) i (23) ulegają pewnej modyfikacji do postaci:

$$\bigwedge_n \bigwedge_{\vartheta < n} (x^{l-1}_{k-1,n}=0) \wedge (u^l_{k,n}=1) \wedge \left\{(\vartheta_{\vartheta,n}=0) \vee \left[(\vartheta_{\vartheta,n}=1) \Longrightarrow (x^l_{k,\vartheta}=1)\right]\right\} \wedge$$

$$\wedge (\sigma^l_{k,n-1} + \xi_n \le c) \Longrightarrow (x^l_{k,n}=1) \tag{34}$$

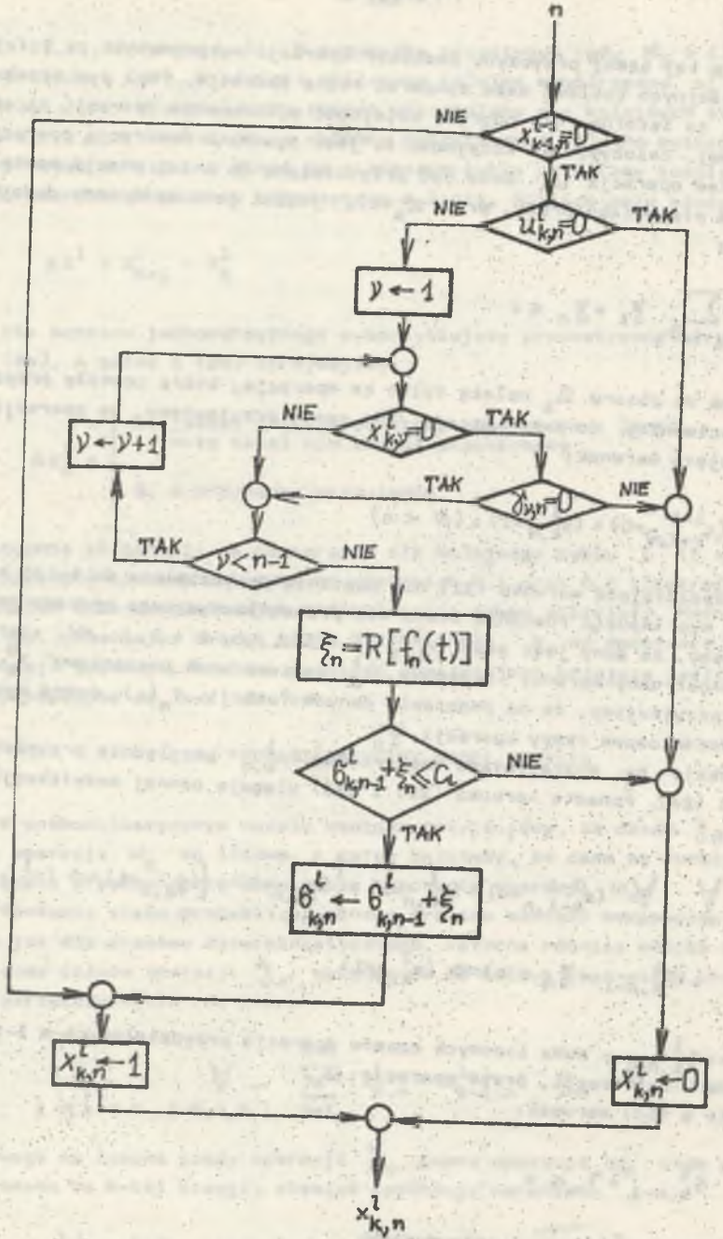
gdzie: $\sigma^l_{k,n-1}$ - suma losowych czasów operacji przydzielonych w l-tym cyklu do k-tej stacji, przed operacją ω_n.

Jeżeli w (34) warunek:

$$\sigma^l_{k,n-1} + \xi_n \le c \tag{34a}$$

nie jest spełniony, to otrzymujemy:

$$x^l_{k,n} = 0 \tag{34b}$$



Rys. D.3. Schemat blokowy obliczeń dla funkcji logicznej h_{ξ} (R - generator liczb pseudolosowych)

Fig. D.3. Block diagram for computation of the logical function h_{ξ} (R - pseudorandom numbers generator)

Schemat blokowy obliczeń dla funkcji h_{ξ} pokazano na rys. D.3. Na schemacie tym uwidoczniono procedurę R generowania pseudolosowych czasów operacji ξ_n.

Dla każdego stanowiska pracy przyjmujemy $G^l_{k,0} = 0$, co należy uwzględnić na rysunku D.2 (bezpośrednio po bloku decyzyjnym $k = \varkappa_m$).

Dla probabilistycznego modelu montażu współrzędne stanu wyznaczamy z następujących równań rekurencyjnych:

$$\left.\begin{aligned}
x^l_{k,1} &= h_{\xi}(x^{l-1}_{k-1,1},\ u^l_{k,1},\ \xi_1)\\
&\cdots\cdots\cdots\cdots\\
&\cdots\cdots\cdots\cdots\\
x^l_{k,n} &= h_{\xi}(x^{l-1}_{k-1,n},\ x^l_{k,1},\ldots,x^l_{k,n-1},\ u^l_{k,n},\ \xi_n,\ G^l_{k,n-1})\\
&\cdots\cdots\cdots\cdots\\
&\cdots\cdots\cdots\cdots\\
x^l_{k,N} &= h_{\xi}(x^{l-1}_{k-1,N},\ x^l_{k,N},\ldots,x^l_{k,N-1},\ u^l_{k,N},\ \xi_N,\ G^l_{k,N-1})
\end{aligned}\right\} \qquad (35)$$

Sumy $G^l_{k,n-1}$ są zależne od czasów ξ_ν, $\nu < n$ na każdej stacji. A zatem równania stanu probabilistycznego procesu montażu na kolejnych stacjach możemy napisać w postaci:

$$\left.\begin{aligned}
x^l_1 &= H_1(x^{l-1}_0,\ u^l_1,\ \xi_1,\ldots,\xi_N)\\
&\cdots\cdots\cdots\cdots\\
&\cdots\cdots\cdots\cdots\\
x^l_k &= H_k(x^{l-1}_{k-1},\ u^l_k,\ \xi_1,\ldots,\xi_N)\\
&\cdots\cdots\cdots\cdots\\
&\cdots\cdots\cdots\cdots\\
x^l_K &= H_K(x^{l-1}_{K-1},\ u^l_K,\ \xi_1,\ldots,\xi_N)
\end{aligned}\right\} \qquad (36)$$

Tak więc ogólne równanie stanu dla probabilistycznego procesu montażu ma postać:

$$x^l = H(x^{l-1},\ u^l,\ \chi^l) \qquad (37)$$

gdzie: χ^l - wektor pseudolosowych czasów operacji w l-tym cyklu.

Przebieg obliczeń dla probabilistycznego procesu montażu zilustrowano na rys. D.2, przy uwzględnieniu procedury h_{ξ} z rys. D.3 oraz warunku $G^l_{k,0} = 0$ przed rozpoczęciem przydzielania operacji na m-tym stanowisku pracy ($\varkappa_m = k$).

Analogicznie jak dla modelu deterministycznego można rejestrować kompletność montażu obiektów schodzących z linii w kolejnych cyklach.

Probabilistyczny model montażu pozwala oszacować efekty montażu dla różnych czasów cyklu. Można w ten sposób wybrać cykl, przy którym efekty montażu są najlepsze. Przy krótkim cyklu liczba obiektów zmontowanych w okresie $[0,C]$ jest duża, lecz prawdopodobieństwo montażu niekompletnego jest również duże. Przy długim cyklu liczba zmontowanych obiektów jest mała, lecz prawdopodobieństwo montażu kompletnego - większe.

6. MODEL MONTAŻU I ZAPASÓW DETALI

Rozważymy obecnie deterministyczny proces montażu uwzględniając zapasy detali. Założymy, że transporter montowanych obiektów nie zatrzymuje się w okresie czasu $[0,C]$, zatem liczba obiektów, które zejdą z linii, wynosi L. Jeżeli w trakcie montażu zabraknie pewnego detalu (tzn. jego zapas będzie zerowy), to pewne obiekty nie zostaną zmontowane kompletnie. Analiza tego zjawiska może być przeprowadzona na podstawie równań stanu.

Załóżmy, że dostawy detali w l-tym cyklu są określone wektorem:

$$D^l = \left[d_n^l\right]_{n=1,\dots,N} \tag{38}$$

gdzie: d_n^l - liczba detali n-tego typu, które są dostarczone do linii montażowej w l-tym cyklu.

A zatem dostawy detali w okresie $[0,C]$ można przedstawić za pomocą macierzy D o L kolumnach.

Przyjmujemy, że w każdym obiekcie należy zamontować jeden detal każdego typu. Ponadto, jeżeli n-ty detal ma być zamontowany w l-tym cyklu, to zapas tego detalu w l-1-azym cyklu nie może być zerowy.

W dalszym ciągu rozważań przyjmujemy wszystkie założenia poczynione dla deterministycznego procesu montażu.

Definicja 3

Stan zapasów detali po l-tym cyklu jest wektorem:

$$z^l = \left[z_n^l\right]_{n=1,\dots,N} \tag{39}$$

gdzie: z_n^l - liczba detali n-tego typu, które po l-tym cyklu znajdują się na stanowiskach pracy linii montażowej.

Zakładamy, że stan początkowy z^0 jest dany.

Przy powyższych założeniach można wyprowadzić równania stanu procesu montażu i zapasów detali, wyróżniając analogiczne przypadki jak dla deterministycznego modelu montażu. Istotna różnica polega jedynie na sprawdzeniu dodatkowego warunku dla zapasów detali:

$$z_n^{l-1} = 0 \tag{40}$$

A zatem przypadek (20) nie ulega zmianie. Przypadek (21) zmodyfikujemy do postaci:

$$\bigwedge_n (x_{k-1,n}^{l-1} = 0) \wedge \left[(u_{k,n}^l = 0) \vee (z_n^{l-1} = 0)\right] \Rightarrow (x_{k,n}^l = 0) \tag{41}$$

gdyż operacja ω_n nie zostanie wykonana z powodu braku n-tego detalu.

W przypadkach (22), (23) i (24) wystarczy jedynie dodatkowo przyjąć, że warunek (40) nie jest spełniony.

Funkcję logiczną, za pomocą której wyznaczamy współrzędne $x_{k,n}^l$, oznaczymy przez h_d. Schemat blokowy przebiegu obliczeń dla tej funkcji pokazano na rys. D.4.

Jeżeli w l-tym cyklu następuje dostawa detali lub zostanie wykonana operacja ω_n, to zapas n-tego detalu ulega zmianie. Modyfikacje zapasów detali opisuje wzór:

$$z_n^l \leftarrow \begin{cases} z_n^{l-1} + d_n^{l-1}, & \text{jeżeli } (x_{k-1,n}^{l-1}=0) \wedge (x_{k,n}^l=1) \\ z_n^{l-1} + d_n^l, & \text{w przypadku przeciwnym} \end{cases} \tag{42}$$

Modyfikacje te pokazano na rys. D.4.

Stosując funkcję h_d można równania stanu procesu montażu zapisać w postaci:

$$\left.\begin{aligned} x_{k,1}^l &= h_d(x_{k-1,1}^{l-1}, u_{k,1}^l, z_1^{l-1}) \\ &\dots\dots\dots\dots \\ &\dots\dots\dots\dots \\ x_{k,n}^l &= h_d(x_{k-1,n}^{l-1}, x_{k,1}^l, \dots, x_{k,n-1}^l, u_{k,n}^l, z_n^{l-1}) \\ &\dots\dots\dots\dots \\ &\dots\dots\dots\dots \\ x_{k,N}^l &= h_d(x_{k-1,N}^{l-1}, x_{k,1}^l, \dots, x_{k,N-1}^l, u_{k,N}^l, z_N^{l-1}) \end{aligned}\right\} \tag{43}$$

Równania te uzupełniamy o równanie (42) określające stan zapasów oraz warunki początkowe x^0, z^0 i lewostronny warunek brzegowy x_0^l. Tak więc dla k-tej stacji ($k = 1,\dots,K$) otrzymamy:

$$\left.\begin{aligned} x_1^l &= H_1(x_0^{l-1}, u_1^l, z^{l-1}) \\ &\cdots\cdots\cdots\cdots \\ &\cdots\cdots\cdots\cdots \\ x_k^l &= H_k(x_{k-1}^{l-1}, u_k^l, z^{l-1}) \\ &\cdots\cdots\cdots\cdots \\ &\cdots\cdots\cdots\cdots \\ x_K^l &= H_K(x_{K-1}^{l-1}, u_K^l, z^{l-1}) \end{aligned}\right\} \qquad (44)$$

Funkcje H_k mają takie samo znaczenie jak w deterministycznym modelu montażu. A zatem działanie tej funkcji ilustruje schemat blokowy przedstawiony na rys. D.2. Na schemacie tym zamiast procedury dla funkcji h należy uwzględnić procedurę dla funkcji h_d, którą pokazano na rys. D.4.

Równania stanu procesu montażu i zapasów detali na linii mają postać:

$$x^l = H(x^{l-1}, u^l, z^{l-1}) \qquad (45a)$$

oraz

$$z^l = F(z^{l-1}, x^{l-1}, u^l, D^l) \qquad (45b)$$

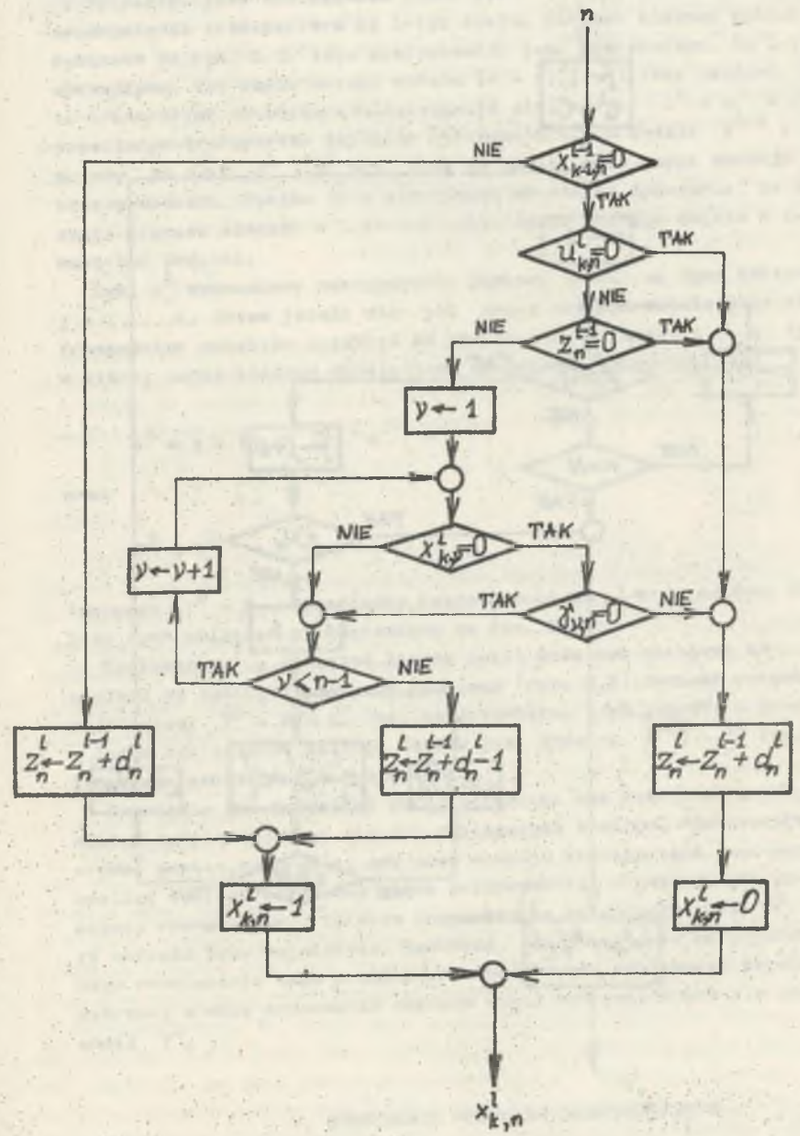
gdzie: F - funkcja transformacji zapasów, oparta na formule(42), przy warunkach x^0, z^0 oraz x_0^l.

Przy zadanym prawostronnym warunku brzegowym x_{K+1}^l (np. (8a)) na podstawie równań stanu można analizować efekty montażu (kompletność) przy braku niektórych detali. Odpowiednią procedurę (dla Δx^l) pokazano na rys. D.2.

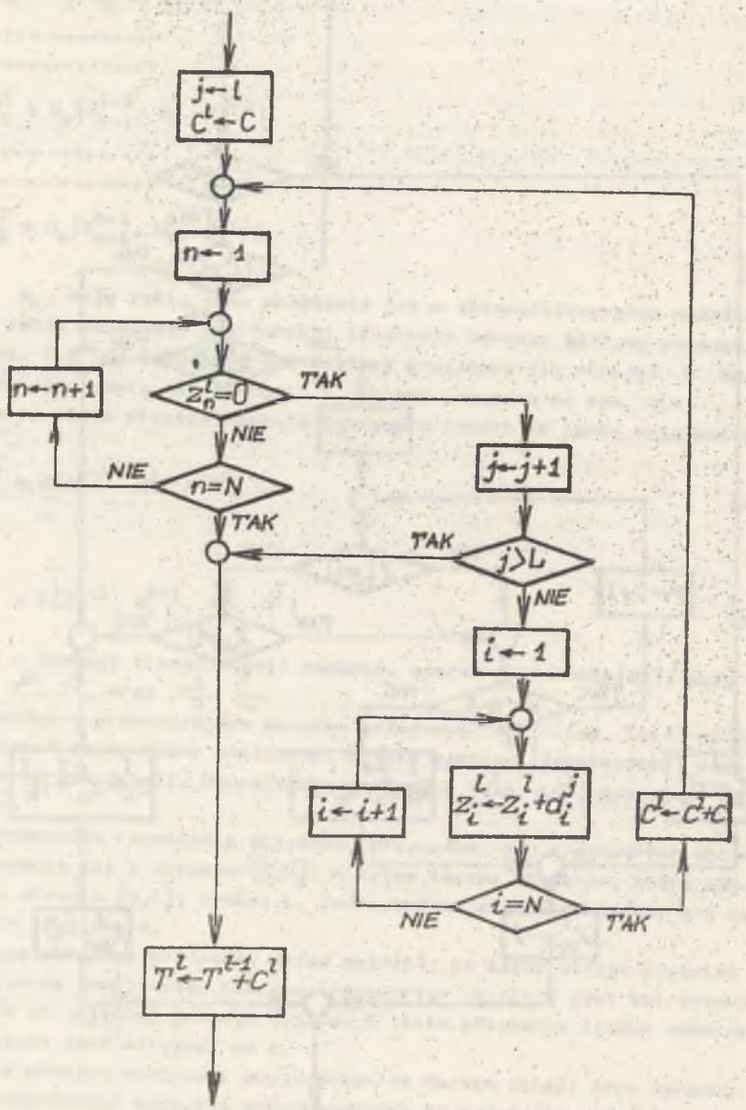
Przeprowadzone rozważania dotyczyły przypadku, gdy transporter obiektów nie zatrzymuje się w okresie $[0,C]$. A zatem liczba obiektów, które zejdą z linii w okresie $[0,C]$, wynosi L. Jednocześnie niektóre obiekty nie są zmontowane kompletnie.

W drugim skrajnym przypadku można założyć, że każdy obiekt powinien być zmontowany kompletnie. Wówczas transporter obiektów jest zatrzymywany przy braku co najmniej jednego detalu. W takim przypadku liczba zmontowanych obiektów jest mniejsza od L.

Analiza procesu montażu z uwzględnieniem dostaw detali przy warunku montażu kompletnego może być przeprowadzona na podstawie równań (45). A zatem obliczenia można prowadzić wg schematu blokowego przedstawionego na rys. D.2, wprowadzając proste modyfikacje. Zamiast procedury dla funkcji h trzeba wprowadzić procedurę dla funkcji h_d z rys. D.4. W procedurze h_d pomijamy warunek $z_n^{l-1} = 0$, ponieważ w montażu kompletnym nie może on być spełniony. Ponadto dla montażu kompletnego $\Delta x^l = 0$, a zatem



Rys. D.4. Schemat blokowy obliczeń dla funkcji logicznej h_d
Fig. D.4. Block diagram for computation of the logical function h_d



Rys. D.5. Schemat blokowy obliczeń czasów cykli c^l oraz chwil T^l uruchamiania transportera głównego

Fig. D.5. Block diagram for computation of the cycle-time c^l and the moment T^l to start main conveyor

odpowiednia procedura na rys. D.2 nie ma znaczenia. Dlatego też procedurę tę zastępujemy obliczeniem czasu cyklu c^l lub inaczej - chwili T^l uruchomienia transportera po l-tym cyklu. Schemat blokowy takich obliczeń pokazano na rys. D.5. Idea postępowania jest następująca. Po l-tym cyklu sprawdzamy, czy zapas n-tego detalu ($n = 1,\ldots,N$) jest dodatni. Jeśli tak, to transporter obiektów nie zatrzymuje się, a cykl $c^l = c$. W przypadku przeciwnym transporter obiektów zatrzymuje się w chwili $T^{l-1} + c$. Przyjmujemy, że cykl c^l ($c^l > c$) trwa do chwili, gdy zapas każdego detalu będzie dodatni. Wynika to z przyjętego wcześniej założenia, że dla realizacji procesu montażu w l+1-szym cyklu zapas każdego detalu w l-tym cyklu musi być dodatni.

Cykl c^l wyznaczamy następująco. Dostawy detali są dane wektorami D^j, $j = 1,\ldots,L$. Zatem jeżeli dla $j=l$ zapas pewnego detalu jest zerowy, to transporter obiektów oczekuje od chwili l.c do chwili $j_1 . c$ ($j_1 > l$), w której zapas każdego detalu jest dodatni. Stąd wynika, że:

$$c^l = c + (j_1 - l) . c \tag{46a}$$

oraz

$$T^l = T^{l-1} + c^l \tag{46b}$$

(warunek $T^0 = 0$ umieszczamy bezpośrednio po $l \leftarrow l$ na rys. D.2). Przebieg tych obliczeń przedstawiono na rys. D.5.

Ponieważ $c^l \geq c$, zatem liczba cykli może być mniejsza od L. Z tego względu na ogólnym schemacie blokowym (rys. D.2) zamiast warunku $l < L$ podstawiamy $T^l + c \leq C$. Jest to oczywiste, gdyż jedynie w przypadku gdy w l-tym cyklu zapas każdego detalu jest dodatni i $C - T^l \geq c$, można analogicznie analizować l+1-szy cykl.

Rozważone dwa przypadki skrajne (montaż bez zatrzymań transportera oraz montaż kompletny) mogą nie dać najlepszych efektów. Wówczas trzeba rozpatrywać pewien kompromis, gdy czas postoju transportera jest ograniczony. Analizę takiego przypadku można przeprowadzić w analogiczny sposób, na podstawie równań stanu. Istotne znaczenie ma określenie chwil T^l, tak by efekty montażu były największe. Jednakże, jak dotąd, nie ma algorytmów optymalnego rozwiązania tego problemu. Można jedynie analizować metodą symulacji cyfrowej efekty stosowania różnych reguł heurystycznych dla określenia chwil T^l.

7. STEROWANIE MONTAŻEM JEDNOWERSYJNYM

Dla przedstawienia problemów sterowania montażem jednowersyjnym rozważymy deterministyczny model montażu. Wyróżnimy sterowania: statyczne, pseudostatyczne, pseudodynamiczne i dynamiczne.

7.1. STEROWANIE STATYCZNE

Ze względów organizacyjnych i technicznych najprostszym rozwiązaniem jest sterowanie statyczne. Wówczas w chwili t=0, czyli na początku okresu $[0,C]$, ustala się podzbiory operacji, które należy wykonywać na stanowiskach pracy. W każdym cyklu podzbiory te są takie same - bowiem są one rezultatem alokacji operacji. Jednakże sterowanie statyczne nie uwzględnia efektów procesu przejściowego, dlatego powoduje dodatkowe straty w montażu.

Załóżmy, że dany jest koszt wykonania operacji ω_n na linii montażowej. Koszt ten może być zależny od montera, który wykonuje tę operację. A zatem przyjmiemy, że dany jest wektor kosztów:

$$G = [g_n] \tag{47a}$$

gdzie: g_n - koszt wykonania operacji ω_n na linii montażowej.

Jeżeli operacja ω_n jest wykonywana poza linią montażową, to koszt jej wykonania jest wyższy. A zatem przyjmiemy, że dany jest wektor dodatkowych kosztów:

$$G' = [g'_n] \tag{47b}$$

gdzie: g'_n - dodatkowy koszt montażu operacji ω_n poza linię.

Straty w przejściowym procesie montażu wynikają z faktu, że monterzy nie wykonują (z przyczyn od nich niezależnych) pewnych operacji. Przypadek taki występuje wtedy, gdy operacja przydzielona monterowi została wykonana wcześniej lub gdy monter nie może wykonać operacji z uwagi na ograniczenia kolejnościowe. Zakładamy przy tym, że płace montera jest zależna od przydzielonych (a nie wykonywanych) operacji. A zatem jeżeli monter nie wykonuje przydzielonych mu operacji, to występuje strata. Jeżeli pominięte operacje muszę być wykonane poza linią, to występuje dodatkowa strata. Oznaczmy przez $q^l_{k,n}$ - stratę związaną z operacją ω_n, która powinna być wykonana na k-tej stacji w l-tym cyklu. Zatem możemy napisać:

$$q^l_{k,n} = \begin{cases} g_n, \text{ jeśli } (u^l_{k,n} = 1) \wedge \left[(x^{l-1}_{k-1,n} = 1) \vee (x^l_{k,n} = 0)\right] \\ 0, \quad \text{w przypadku przeciwnym.} \end{cases} \tag{48}$$

Strata q^l_k dla k-tej stacji w l-tym cyklu:

$$q^l_k = \sum_{n=1}^{n=N} q^l_{k,n} \tag{49}$$

Proces przejściowy na k-tej stacji trwa k-1 cykli, a zatem strata q_k dla k-tej stacji:

$$q_k = \sum_{l=1}^{l=k-1} q^l_k \qquad k=2,\ldots,K \tag{50}$$

Stąd strata w procesie przejściowym wynosi:

$$Q = \sum_{k=2}^{k=K} q_k \tag{51}$$

Analizując obiekty schodzęce z linii w procesie przejściowym, wyznaczamy straty dodatkowe wynikające z montażu niekompletnego:

$$Q' = \sum_{l=1}^{l=K-1} \sum_{n=1}^{n=N} (x^l_{K+1,n} - x^l_{K,n}) \cdot g'_n \tag{52}$$

Jak wynika ze wzorów (48) i (52), straty sumaryczne w procesie przejściowym można wyznaczyć symulując proces montażu na podstawie równań stanu (przyjmując sterowanie U_k wynikające z alokacji operacji).

7.2. STEROWANIE PSEUDOSTATYCZNE

Sterowanie procesem przejściowym nazwiemy pseudostatycznym, jeżeli spełnia warunek:

$$\bigwedge_{2 \le l \le L} (U^l \neq U^{l-1}) \wedge (U^{l-1} = U^l) \tag{53}$$

Tak więc począwszy od drugiego cyklu sterowanie jest statyczne. Nazwa sterowanie pseudostatyczne wiąże się z tym, że pierwszy cykl jest umowny. W trakcie tego cyklu transporter jest nieruchomy, a monterzy wykonują zadania na różnych stacjach. Ponadto w trakcie sterowania pseudostatycznego proces montażu może mieć przebieg dynamiczny.

Rozważmy pracę M monterów w pierwszym cyklu. Monterzy wykonują K zadań. Każde k-te zadanie jest podzbiorem operacji montażowych na k-tej stacji. Załóżmy, że dany jest stan początkowy X^0 oraz stan końcowy X^{L+1} (dla $K < L$). Ponadto przyjmujemy lewostronny warunek brzegowy $x^1_0 = 0$ oraz prawostronny warunek brzegowy $x^1_{K+1} = 1$ (jako warunek montażu kompletnego).

Zadanie określamy za pomocą wektora:

$$R_k = \left[r_{k,n}\right]_{n=1,\ldots,N} \tag{54}$$

Elementy tego wektora definiujemy następująco:

$$r_{k,n} = \begin{cases} 1, \text{ jeśli } x^{L+1}_{k,n} > x^{O}_{k,n} \\ O, \text{ w przypadku przeciwnym.} \end{cases} \tag{55}$$

Czasy τ_k wykonania zadań wyznaczamy jako:

$$\tau_k = \sum_{n=1}^{n=N} r_{k,n} \cdot \vartheta_n \tag{56}$$

Przyjmując, że monterzy realizują zadania na sąsiednich stacjach, otrzymujemy sekwencję zadań od k=1 do k=K. Dla wyznaczenia zadań, które w cyklu c^1 ma wykonać m-ty monter, należy rozwiązać prosty problem balansowania. Tak więc m-temu monterowi przydzielamy zadania o numerach od $k_{m-1}+1$ do k_m, przy tym musi być spełniony warunek:

$$\sum_{k=k_{m-1}+1}^{k=k_m} \tau_k \leq c^1 < \sum_{k=k_{m-1}+1}^{k=k_m+1} \tau_k \qquad \begin{matrix} m=1,\ldots,M \\ k_O=O \end{matrix} \tag{57}$$

Cykl c^1 jest najkrótszym czasem potrzebnym do wykonania wszystkich zadań przez M monterów. Tak więc:

$$c^1 \geq \frac{1}{M} \sum_{k=1}^{k=K} \tau_k \tag{58}$$

oraz przyjmuje się, że c^1 jest wielokrotnością cyklu c, tzn.:

$$c^1 = l_1 \cdot c \tag{59}$$

gdzie: l_1 - liczba naturalna.

W wyniku realizacji procesu montażu w pierwszym cyklu przechodzimy ze stanu X^O do stanu X^1. Ponieważ transporter był nieruchomy, zatem stosujemy wzór:

$$X^1 = X^O + R \tag{60}$$

Prosta suma algebraiczna w (60) wynika z faktu, że R jest macierzą zadań (operacji różnicowych), których nie ma w X^O. Po uruchomieniu transportera proces montażu opisują równania stanu, przy warunku początkowym X^1.

Sterowanie pseudostatyczne pozwala wyeliminować straty wynikające z niewykonywania przez monterów pewnych operacji, które w konsekwencji prowadzą do montażu niekompletnego. A zatem składowa strat dodatkowych (52) jest zerem. Ponadto składowa strat (48) ulega zmniejszeniu, bowiem:

$$(u^1_{k,n} = 1) \Longrightarrow (x^1_{k,n} = 1) \tag{61}$$

A zatem otrzymamy:

$$Q = \sum_{k=2}^{k=K} \sum_{l=1}^{l=k-1} \sum_{n=1}^{n=N} q^1_{k,n} \tag{62}$$

przy tym:

$$q^1_{k,n} = \begin{cases} g_n, \text{ jeśli } (u^1_{k,n}=1) \wedge (x^{l-1}_{k-1,n}=1) \\ O, \text{ w przypadku przeciwnym.} \end{cases} \tag{62a}$$

Ponieważ w trakcie sterowania pseudostatycznego transporter jest zatrzymany przez l_1 cykli c, zatem w przedziale czasu $[O,C]$ z linii zejdzie tylko $L-l_1$ obiektów. Stąd drugi składnik strat wynosi:

$$Q^* = l_1 \cdot g \tag{63}$$

gdzie: g - strata linii wynikające z braku jednego obiektu.

Sterowanie pseudostatyczne pozwala montować obiekty kompletne, jednakże liczba tych obiektów jest mniejsza niż L.

7.3. STEROWANIE PSEUDODYNAMICZNE

Sterowanie pseudodynamiczne można przedstawić w postaci:

$$U^1 = \overline{U}^1 + \vec{U}^1 \tag{64}$$

Składowa zmienna $\vec{U}^1$ występuje tylko w procesie przejściowym, tzn. dla $1 \leq l < K$. Dla cykli $K \leq l \leq L$ sterowanie $\vec{U}^1 = O$. Składowa stała $\overline{U}^1$ jest sterowaniem statycznym wyznaczonym w wyniku rozwiązania problemu alokacji operacji (AO).

Celem sterowania pseudodynamicznego jest zapewnienie montażu kompletnego w przedziale $[0,C]$. Aby ten cel zrealizować, w kolejnych cyklach $1 \leq l < K$ na stanowiskach pracy wyznaczane są operacje, które mogą i muszą być wykonane. Zatem, jeżeli z A0 wynika, że na k-tym stanowisku pracy ma być wykonana operacja ω_n, lecz operacja ta została wykonana wcześniej (przed przedziałem $[0,C]$), to odpowiednia korekta sterowania jest zawarta w $\tilde{U}^l$ ($\tilde{u}^l_{k,n} = -1$). Analogicznie, jeśli operacja ω_ν miała być wykonana przed k-tą stację, lecz nie została wykonana, to $\tilde{u}^l_{k,\nu} = 1$. W pozostałych przypadkach $\tilde{u}^l_{k,n} = 0$.

Ze sterowania pseudodynamicznego wynika, że w kolejnych cyklach l, $(1 \leq l < K)$, podzbiory operacji, jakie powinny być wykonane na stanowisku pracy, mogą być różne. A zatem również cykle c^l mogą być różne. Efekt sterowania pseudodynamicznego można ocenić na podstawie liczby obiektów zmontowanych w przedziale $[0,C]$. Jednakże począwszy od K-tego cyklu proces montażu jest statyczny. A zatem problem sprowadza się do minimalizacji czasu trwania procesu przejściowego.

Załóżmy, że dany jest stan początkowy x^0, stan końcowy x^{L+1} (dla $K < L$) oraz sterowanie statyczne $\bar{U}^l$. Sterowanie dynamiczne $\tilde{U}^l$ wyznaczamy dla kolejnych cykli l, $1 \leq l < K$, następująco:

Krok 1: dla $l=1$ otrzymujemy:

$$\tilde{U}^1 = x^{L+1} - x^0 \tag{65}$$

oraz

$$U^1 = \bar{U}^1 + \tilde{U}^1 \tag{66}$$

$$c^1 = \max_{1 \leq k \leq K} \sum_{n=1}^{n=N} u^1_{k,n} \cdot \vartheta_n \tag{67}$$

Na podstawie U^1 z równań stanu wyznaczamy x^1.

Krok l: dla $1 < l < K$ otrzymujemy:

$$\tilde{U}^l = x^{L+1} - x^{l-1} \tag{68}$$

oraz

$$U^l = \bar{U}^l + \tilde{U}^l \tag{69}$$

Na podstawie U^l z równań stanu wyznaczamy x^l.

W każdym cyklu l, $1 \leq l < K$ wyznaczamy czas jego trwania:

$$c^l = \max_{1 \leq k \leq K} \sum_{n=1}^{n=N} u^l_{k,n} \cdot \vartheta_n \tag{70}$$

Ponadto:

$$Q = \sum_{l=1}^{l=K-1} c_l \tag{71}$$

7.4. STEROWANIE DYNAMICZNE

W sterowaniu dynamicznym zakłada się, że musi być spełniony warunek montażu kompletnego (8a). Ponadto liczba obiektów zmontowanych w okresie $[0,C]$ powinna być maksymalna. Z tych względów w każdym cyklu l, $l=1,\ldots,L$ na każdym stanowisku pracy mogą być wykonywane różne podzbiory operacji. Sterowanie dynamiczne U^l nie jest dane dla $l = 1,\ldots,L$ lecz należy je sukcesywnie wyznaczać.

Załóżmy, że dany jest stan początkowy x^0 i prawostronny warunek brzegowy (8a), a ponadto $\aleph_M = K$. Ogólny algorytm dla sterowania dynamicznego ma następującą postać:

Krok 1: dla $l=1$ otrzymujemy:

$$u^1_K = 1 - x^0_{K-1} \tag{72}$$

oraz

$$c^1 = \sum_{n=1}^{n=N} u^1_{K-1,n} \cdot \vartheta_n \tag{73}$$

Ponadto dla $k = 1,\ldots,K-1$ wyznaczamy u^1_k tak, by spełniony był warunek:

$$\sum_{n=1}^{n=N} u^1_{k,n} \cdot \vartheta_n \leq c^1 \tag{74}$$

Dla stacji, na których nie są zlokalizowane stanowiska pracy, sterowanie $u^1_k = 0$.

Krok l: dla $1 < l \leq L$ otrzymujemy:

$$u^l_K = 1 - x^{l-1}_{K-1} \tag{75}$$

oraz

$$c^l = \sum_{n=1}^{n=N} u^l_{K,n} \cdot \vartheta_n \tag{76}$$

Ponadto dla stacji k, $(k = \mathcal{K}_m)$, na których są zlokalizowane stanowiska pracy, wyznaczamy U_k^l tak, by spełniony był warunek:

$$\sum_{n=1}^{n=N} u_{k,n}^l \cdot \vartheta_n \leq c^l \tag{77}$$

Obliczenia kończymy, gdy:

$$\sum_{l=1}^{l=L} c^l \leq C < \sum_{l=1}^{l=L+1} c^l \tag{78}$$

W każdym cyklu $l = 1,\ldots,L$ można w różny sposób wyznaczyć U_k^l, dla $k = 1,\ldots,K-1$. Jeżeli kolejność wykonywania operacji jest ustalona (od 1 do N), to operacje, które mają być wykonane w l-tym cyklu na k-tej stacji $(k = \mathcal{K}_m,\ m = 1,\ldots,M-1)$ wyznaczamy następująco. Oznaczmy przez n_{k-1} numer ostatniej operacji wykonywanej w stanie x_{k-1}^{l-1} (dla x_0^{l-1}, $n_0=0$). Zatem n_k wyznaczamy z warunku:

$$\sum_{n=n_{k-1}+1}^{n=n_k} \vartheta_n \leq c^l < \sum_{n=n_{k-1}+1}^{n=n_k+1} \vartheta_n \tag{79}$$

oraz

$$n_k \leq N \tag{79a}$$

W tym przypadku sterowanie U_k^l ma postać:

$$u_{k,n}^l = \begin{cases} 1, \text{ jeśli } n_{k-1} < n \leq n_k \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{79b}$$

W innych przypadkach sterowanie U_k^l wyznacza się za pomocą bardziej złożonych algorytmów. Jeżeli kolejność wykonywania operacji nie jest określona, a jedynie musi być respektowana relacja Γ, to można maksymalizować sumę czasów operacji, które mają być wykonywane w l-tym cyklu na k-tej stacji. Wówczas sterowanie U_k^l wyznaczamy za pomocą algorytmów programowania wieloetapowego.

Jeżeli podzbiór operacji Ω_k $(k = 1,\ldots,K)$, które mogą być wykonane na k-tej stacji jest ograniczony, to cykl c^l wynika nie tylko z ostatniego stanowiska pracy. Załóżmy, że:

$$\bigcup_{k=1}^{k=K} \Omega_k = \Omega \tag{80a}$$

$$\Omega_k \cap \Omega_{k+1} \neq 0 \qquad k = 1,\ldots,K-1 \tag{80b}$$

$$\Omega_k \cap \Omega_{k+\mathcal{K}} = 0 \qquad \mathcal{K} > 1 \tag{80c}$$

Przy tych założeniach na k-tej stacji można wykonać najwyżej wszystkie operacje ze zbioru Ω_k. Ponadto na k-tej stacji muszą być wykonane co najmniej te operacje, które nie zostały wykonane na stacjach $j < k$ oraz nie należą do zbioru Ω_{k+1}. Powiemy, że operacje te tworzą najmniejszy podzbiór zbioru Ω_k. W przeciwnym przypadku warunek montażu kompletnego nie byłby spełniony (jeżeli operacja ω_n nie należy do zbioru $\Omega_k \setminus \Omega_{k+1}$, to ω_n nie należy również do zbioru $\Omega_k \setminus \Omega_{k+\mathcal{K}}$, $\mathcal{K} > 1$). A zatem cykl c^l wyznaczamy jako:

$$c^l = \max_{1 \leq k \leq K} c_k^l \tag{81}$$

gdzie: c_k^l - czas wykonania najmniejszego podzbioru operacji na k-tej stacji w l-tym cyklu.

Dla tak wyznaczonego cyklu c^l można określić sterowania U_k^l w sposób pokazany wyżej.

8. MODEL MONTAŻU WIELOWERSYJNEGO

Montaż wielowersyjny stanowi uogólnienie montażu jednowersyjnego. Wersja obiektu jest określona przez zbiór detali, z których składa się ten obiekt. Załóżmy, że danych jest S wersji. Montaż obiektu s-tej wersji wymaga wykonania wszystkich operacji ze zbioru Ω^s. Załóżmy, że liczba tych operacji wynosi N^s.

W dalszym ciągu rozważać będziemy zbiór wszystkich operacji, tzn.:

$$\Omega = \bigcup_{s=1}^{s=S} \Omega^s \tag{82}$$

Przyjmiemy, że

$$\bigcap_{s=1}^{s=S} \Omega^s \neq 0 \tag{83}$$

Liczbę operacji zbioru Ω oznaczymy przez N.

Załóżmy, że na podstawie relacji Γ^s $(s = 1,\ldots,S)$ określona jest relacja Γ dla operacji ze zbioru Ω. Przez ϑ_n $(n = 1,\ldots,N)$ oznaczymy czas wykonania operacji ω_n. Przyjmiemy, że numeracja operacji spełnia warunek (1).

Załóżymy, że przyporządkowanie operacji ω_n do s-tej wersji jest określone macierzą:

$$W = \left[w_{n,s}\right]_{\substack{n=1,\ldots,N \\ s=1,\ldots,S}} \tag{84}$$

Elementy tej macierzy mają następujące znaczenie:

$$w_{n,s} = \begin{cases} 1, \text{ jeśli operacja } \omega_n \text{ należy do s-tej wersji} \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{84a}$$

Dla wielowersyjnego procesu montażu stan zdefiniujemy następująco.

Definicja 4

Stan wielowersyjnego procesu montażu na k-tej stacji, $(k = 1,\ldots,K)$ po l-tym cyklu $(l = 1,\ldots,L)$ jest wektorem:

$$x_k^l = \left[x_{k,n}^l\right]_{n=0,\ldots,N} \tag{85}$$

Element $x_{k,0}^l$ jest numerem s, $(s = 1,\ldots,S)$ wersji obiektu montowanego w l-tym cyklu na k-tej stacji. Pozostałe elementy (dla $n = 1,\ldots,N$) są określone w (3a).

A zatem stan wielowersyjnego procesu montażu na linii po l-tym $(l=1,\ldots,L)$ cyklu jest macierzą - analogiczną do (4). Tym samym w analogiczny sposób jak w (5) i (6) określamy warunki początkowy i końcowy dla montażu wielowersyjnego.

W lewostronnym warunku brzegowym dla montażu wielowersyjnego istotne znaczenie ma współrzędna $x_{0,0}^l$, $(l = 1,\ldots,L)$. Współrzędna ta określa wersję obiektu, który w l+1-szym cyklu znajdzie się na pierwszej stacji. Pozostałe elementy warunku x_0^l (dla $n = 1,\ldots,N$), analogicznie jak w (7), są zerowe.

Prawostronny warunek brzegowy x_{K+1}^l ma istotne znaczenie dla montażu kompletnego. Współrzędna $x_{K+1,0}^l$ określa wersję obiektu, który w l-tym cyklu powinien być na K+1-szej stacji. Jeżeli:

$$x_{K+1,0}^l = s \tag{86}$$

to warunek montażu kompletnego dla l-tego cyklu przyjmie postać:

$$\bigvee_{1 \le n \le N} (w_{n,s}=1) \Rightarrow (x_{K+1,n}^l=1) \tag{87}$$

A zatem w obiekcie znajdującym się po l-tym cyklu na K+1-szej stacji muszą być wykonane wszystkie operacje $\omega_n \in \Omega^s$.

Przyjmując definicję stanu (85) zauważymy, że montaż wielowersyjny jest procesem dynamicznym - w sensie definicji (10).

Definicja 5

Sterowanie wielowersyjnym procesem montażu w l-tym cyklu $(l = 1,\ldots,L)$, na k-tej stacji $(k = 1,\ldots,K)$ jest wektorem:

$$u_k^l = \left[u_{k,n}^l\right]_{n=0,1,\ldots,N} \tag{88}$$

Element $u_{k,0}^l$ jest numerem s $(s = 1,\ldots,S)$ wersji obiektu montowanego w l-tym cyklu na k-tej stacji $(k = \aleph_m, m = 1,\ldots,M)$. Pozostałe elementy (dla $n = 1,\ldots,N$) są określone w (13b). A zatem analogicznie jak w (16) sterowanie wielowersyjnym procesem montażu w l-tym cyklu można przedstawić w postaci macierzy. Ponadto w sensie (18) jest to sterowanie dynamiczne.

W montażu wielowersyjnym zachodzi:

$$u_{k,0}^l = u_{k-1,0}^{l-1} \quad \begin{matrix} k=1,\ldots,K+1 \\ l=1,\ldots,L \end{matrix} \tag{89}$$

natomiast współrzędna $u_{0,0}^{l-1}$ jest sterowalna. Kolejność wersji obiektów podawanych na pierwszą stację ma wpływ na efekty montażu wielowersyjnego.

W równaniach stanu montażu wielowersyjnego uwzględniamy (89), a poza tym wyróżniamy przypadki (20) ÷ (24) - jak dla montażu jednowersyjnego.

Tak więc równania stanu dla montażu wielowersyjnego, w l-tym cyklu na k-tej stacji, stanowią formuły (89) i (26). Tym samym ogólne równanie stanu dla montażu wielowersyjnego ma postać (28). Jednakże dla montażu wielowersyjnego istotne znaczenie ma lewostronny warunek brzegowy x_0^l, poprzez który sterowalny jest proces montażu.

Na schemacie blokowym przedstawionym na rys. D.2 dla montażu wielowersyjnego trzeba wprowadzić pewne modyfikacje. Formułę (89) $(u^{l}_{k,0} = u^{l-1}_{k-1,0})$ należy umieścić bezpośrednio przed blokiem dla $n = 1$. Ponadto w procedurze Δx^{l} analizy kompletności montażu przyjmujemy:

$$\Delta x^{l}_{0} = x^{l-1}_{K,0} = s \tag{90a}$$

oraz

$$\Delta x^{l}_{n} = \begin{cases} x^{l}_{K+1,n} - x^{l}_{K,n}, \text{ jeśli } w_{n,s}=1 \\ 0, \text{ w przypadku przeciwnym.} \end{cases} \tag{90b}$$

Ponadto sterowanie U^{l} na ogół nie jest dane, gdyż $u^{l}_{k,0}$ jest zależne od sterowalnego lewostronnego warunku brzegowego dla $u^{l}_{0,0}$.

Dla montażu wielowersyjnego można, analogicznie jak dla montażu jednowersyjnego, przedstawić inne modele: probabilistyczny oraz uwzględniający dostawy detali. Jednakże zasadnicze znaczenie ma sterowanie dynamiczne procesem deterministycznym.

9. STEROWANIE DYNAMICZNE MONTAŻEM WIELOWERSYJNYM

Montaż wielowersyjny jest procesem dynamicznym, wymagającym sterowania dynamicznego. Rozważymy dwa warianty takiego sterowania, które nazwiemy quasi-dynamicznym oraz dynamicznym.

Definicja 6

Sterowanie wielowersyjnym procesem montażu jest quasi-dynamiczne, jeżeli spełnia warunek:

$$\bigvee_{1\le k\le K} \bigvee_{1\le l\le L} \bigvee_{1\le \mathfrak{N}\le L} (u^{l}_{k,0} = u^{\mathfrak{N}}_{k,0}) \Rightarrow (u^{l}_{k} = u^{\mathfrak{N}}_{k}) \tag{91}$$

Sterowanie quasi-dynamiczne oznaczymy przez $\hat{U}^{l}$.
A zatem podzbiór operacji, jakie należy wykonać w l-tym cyklu na k-tej stacji, jest zależny tylko od wersji montowanego obiektu. Takie proste sterowanie dynamiczne jest stosowane, gdy operacje są wykonywane przez monterów.

Przyjmiemy, że sterowanie dynamiczne nie spełnia warunku (91). Tym samym na pewnej stacji przy montażu obiektu tej samej wersji w różnych cyklach są wykonywane różne podzbiory operacji. Ten złożony sposób sterowania można rozpatrywać, gdy operacje są wykonywane przez roboty przemysłowe.

W przypadku sterowania quasi-dynamicznego zakładamy, że dany jest wektor (2) alokacji operacji do stacji montażowych. Ponadto m-te stanowisko pracy, $(m = 1,\ldots,M)$ znajduje się na stacji $\mathfrak{K}_{m}$ niezależnie do wersji montowanego obiektu, w całym okresie $[0,C]$.

Załóżmy, że dane są: x^{l}_{0}, x^{0}, x^{l}_{K+1}. Sterowanie quasi-dynamiczne u^{l}_{k} w l-tym cyklu, $(l = 1,\ldots,L)$ na k-tej stacji, $(k = 1,\ldots,K)$ określamy następująco:

$$\hat{u}^{l}_{k,n} = \begin{cases} 1, \text{ jeśli } (a_{n}=m) \wedge (\mathfrak{K}_{m}=k) \wedge (\hat{u}^{l}_{k,0}=s) \wedge (w_{n,s}=1) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \quad n=1,\ldots,N. \tag{92}$$

Współrzędną $\hat{u}^{l}_{k,0}$ wyznaczamy z (89).
Czas cyklu c^{l} określamy jako:

$$c^{l} = \max_{1\le k\le K} \sum_{n=1}^{n=N} \hat{u}^{l}_{k,n} \cdot \vartheta^{\nu}_{n} \tag{93}$$

Algorytm quasi-dynamicznego sterowania montażem wielowersyjnym jest oparty na równaniach stanu. W l-tym cyklu $(l = 1,\ldots,L)$ wyznaczamy $\hat{U}^{l}_{k}$ $(k = 1,\ldots,K)$ zgodnie z (92) oraz c^{l} na podstawie (93). Następnie mając stan X^{l} i sterowanie $\hat{U}^{l}$ wyznaczamy - z równań stanu - kolejny stan X^{l+1}. Na podstawie stanu X^{l+1} otrzymujemy sterowanie $\hat{U}^{l+1}$ itd.

Jako kryterium sterowania quasi-dynamicznego przyjmuje się maksymalizację liczby L obiektów zmontowanych w okresie $[0,C]$. A zatem obliczenia kończymy, gdy spełniony jest warunek (78).

Sterowanie quasi-dynamiczne jest szeroko stosowane w praktyce. Z tego względu poczynimy kilka uwag krytycznych. Przede wszystkim należy podkreślić, że przy sterowaniu quasi-dynamicznym na ogół nie jest spełniony warunek montażu kompletnego dla początkowych K obiektów schodzących z linii. Przez analogię do montażu jednowersyjnego można powiedzieć, że w początkowych K cyklach proces montażu jest przejściowy. Proces przejściowy nie występuje (a montaż jest kompletny) tylko w przypadku, gdy sterowanie quasi-dynamiczne w dwóch kolejnych okresach $[0,C]$ jest identyczne.

Liczba obiektów zmontowanych w okresie $[0,C]$ jest zależna od kolejności wprowadzania na pierwszą stację obiektów różnej wersji. Dlatego też w montażu wielowersyjnym istotne znaczenie ma tzw. problem kolejnościowy. Polega on na określeniu najlepszej kolejności montażu obiektów różnej wersji. W dwóch skrajnych przypadkach wyróżnia się montaż seryjny oraz montaż mieszany. W montażu seryjnym obiekty należące do jednej wersji są po-

dawane na linię kolejno jeden po drugim. Na ogół liczba obiektów w serii jest większa niż K. W takim przypadku problem kolejnościowy upraszcza się, bowiem liczba serii S jest znacznie mniejsza od liczby wszystkich obiektów. Ponadto istotne znaczenie mają jedynie te cykle, w których na linii znajdują się obiekty dwóch różnych wersji. W przeciwnym przypadku mamy do czynienia ze statycznym procesem jednowersyjnym.

W przypadku montażu mieszanego dokładne rozwiązanie problemu kolejnościowego nie jest możliwe z uwagi na długi czas obliczeń. Z tego względu stosowane są algorytmy heurystyczne. Algorytmy te są oparte na symulacji procesu montażu.

Sterowanie dynamiczne montażem wielowersyjnym można przedstawić analogicznie jak dla montażu jednowersyjnego. Tak więc przyjmujemy warunek montażu kompletnego (87). Czas cyklu c^l wyznaczamy na podstawie stanu procesu montażu na ostatnim stanowisku pracy (niech $\mathcal{K}_M = K$). Sterowanie dynamiczne U^l nie jest dane (na podstawie wektora (2)), lecz należy je sukcesywnie wyznaczać dla kolejnych cykli $l = 1,\ldots,L$.

Załóżmy, że dany jest stan początkowy X^O, lewostronny warunek brzegowy x^l_O oraz prawostronny warunek montażu kompletnego (87). Ogólny algorytm sterowania dynamicznego ma następującą postać:

Krok l: dla $l = 1,\ldots,L$

- jeżeli $x^{l-1}_{K-1,O} = s$, to

$$u^l_{K,O} = s \tag{94}$$

- dla $n = 1,\ldots,N$ otrzymujemy

$$u^l_{K,n} = \begin{cases} 1, \text{ jeśli } (x^{l-1}_{K-1,n}=0) \wedge (w_{n,s}=1) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{95}$$

- czas cyklu c^l określamy jako

$$c^l = \sum_{n=1}^{n=N} u^l_{K,n} \cdot \vartheta_n \tag{96}$$

- dla stacji $k = \mathcal{K}_m$, $m = 1,\ldots,M-1$

$$u^l_{k,O} = u^{l-1}_{k-1,O} = s \tag{97}$$

- dla kolejnych $n = 1,\ldots,N$ uwzględniając (96) i (97) otrzymujemy

$$u^l_{k,n} = \begin{cases} 1, \text{ jeśli } (w_{n,s}=1) \wedge (x^{l-1}_{k-1,n}=0) \wedge \left(\sum_{\vartheta=1}^{\vartheta=n} u^l_{k,\vartheta} \cdot \vartheta_\vartheta \leq c^l\right) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{98}$$

- dla stacji $k \neq \mathcal{K}_m$ otrzymujemy

$$x^l_k = x^{l-1}_{k-1} \tag{99}$$

Obliczenia kończymy, gdy spełniony jest warunek (78). W algorytmie tym założono, że dla każdej wersji operacje są wykonywane w kolejności zgodnej z numeracją.

10. ZAKOŃCZENIE

W przeprowadzonych rozważaniach przedstawiono modele matematyczne montażu dynamicznego. Z wprowadzonej klasyfikacji wynika, że alokacja zadań i zasobów, rozpatrywana w rozprawie, jest formą sterowania statycznego montażem.

Stosowanie sterowania statycznego jest uzasadnione dla montażu seryjnego, gdyż zaburzenia występują jedynie w procesie przejściowym. Ponadto w zależności od efektów tych zaburzeń można stosować różne formy sterowania. Sterowanie procesem przejściowym wymaga ewidencji wykonanych operacji w każdym cyklu, na każdym stanowisku pracy. Dlatego też jest to możliwe w przypadku wykorzystania robotów przemysłowych oraz mikroprocesorowego systemu sterowania.

Przedstawione modele matematyczne montażu dynamicznego są oparte na równaniach stanu. Ten sposób modelowania procesów dyskretnych daje pewne analogie do procesów ciągłych. Ponadto umożliwia jasną klasyfikację procesów i sterowań.

Modele matematyczne oparte na równaniach stanu są modelami typu symulacyjnego. Można za ich pomocą wyznaczyć racjonalne sterowanie, bez gwarancji optymalności.

dawane na linię kolejno jeden po drugim. Na ogół liczba obiektów w serii jest większa niż K. W takim przypadku problem kolejnościowy upraszcza się, bowiem liczba serii S jest znacznie mniejsza od liczby wszystkich obiektów. Ponadto istotne znaczenie mają jedynie te cykle, w których na linii znajdują się obiekty dwóch różnych wersji. W przeciwnym przypadku mamy do czynienia ze statycznym procesem jednowersyjnym.

W przypadku montażu mieszanego dokładne rozwiązanie problemu kolejnościowego nie jest możliwe z uwagi na długi czas obliczeń. Z tego względu stosowane są algorytmy heurystyczne. Algorytmy te są oparte na symulacji procesu montażu.

Sterowanie dynamiczne montażem wielowersyjnym można przedstawić analogicznie jak dla montażu jednowersyjnego. Tak więc przyjmujemy warunek montażu kompletnego (87). Czas cyklu c^l wyznaczamy na podstawie stanu procesu montażu na ostatnim stanowisku pracy (niech $\mathcal{K}_M = K$). Sterowanie dynamiczne U^l nie jest dane (na podstawie wektora (2)), lecz należy je sukcesywnie wyznaczać dla kolejnych cykli $l = 1,\ldots,L$.

Załóżmy, że dany jest stan początkowy X^0, lewostronny warunek brzegowy x_0^l oraz prawostronny warunek montażu kompletnego (87). Ogólny algorytm sterowania dynamicznego ma następującą postać:

Krok l: dla $l = 1,\ldots,L$

- jeżeli $x^{l-1}_{K-1,0} = s$, to

$$u^l_{K,0} = s \tag{94}$$

- dla $n = 1,\ldots,N$ otrzymujemy

$$u^l_{K,n} = \begin{cases} 1, \text{ jeśli } (x^{l-1}_{K-1,n}=0) \wedge (w_{n,s}=1) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{95}$$

- czas cyklu c^l określamy jako

$$c^l = \sum_{n=1}^{n=N} u^l_{K,n} \cdot \vartheta_n \tag{96}$$

- dla stacji $k = \mathcal{K}_m$, $m = 1,\ldots,M-1$

$$u^l_{k,0} = u^{l-1}_{k-1,0} = s \tag{97}$$

- dla kolejnych $n = 1,\ldots,N$ uwzględniając (96) i (97) otrzymujemy

$$u^l_{k,n} = \begin{cases} 1, \text{ jeśli } (w_{n,s}=1) \wedge (x^{l-1}_{k-1,n}=0) \wedge \left(\sum_{\vartheta=1}^{\vartheta=n} u^l_{k,\vartheta} \cdot \vartheta_\vartheta \le c^l\right) \\ 0, \text{ w przypadku przeciwnym} \end{cases} \tag{98}$$

- dla stacji $k \neq \mathcal{K}_m$ otrzymujemy

$$x^l_k = x^{l-1}_{k-1} \tag{99}$$

Obliczenia kończymy, gdy spełniony jest warunek (78). W algorytmie tym założono, że dla każdej wersji operacje są wykonywane w kolejności zgodnej z numeracją.

10. ZAKOŃCZENIE

W przeprowadzonych rozważaniach przedstawiono modele matematyczne montażu dynamicznego. Z wprowadzonej klasyfikacji wynika, że alokacja zadań i zasobów, rozpatrywana w rozprawie, jest formę sterowania statycznego montażem.

Stosowanie sterowania statycznego jest uzasadnione dla montażu seryjnego, gdyż zaburzenia występują jedynie w procesie przejściowym. Ponadto w zależności od efektów tych zaburzeń można stosować różne formy sterowania. Sterowanie procesem przejściowym wymaga ewidencji wykonanych operacji w każdym cyklu, na każdym stanowisku pracy. Dlatego też jest to możliwe w przypadku wykorzystania robotów przemysłowych oraz mikroprocesorowego systemu sterowania.

Przedstawione modele matematyczne montażu dynamicznego są oparte na równaniach stanu. Ten sposób modelowania procesów dyskretnych daje pewne analogie do procesów ciągłych. Ponadto umożliwia jasną klasyfikację procesów i sterowań.

Modele matematyczne oparte na równaniach stanu są modelami typu symulacyjnego. Można za ich pomocą wyznaczyć racjonalne sterowanie, bez gwarancji optymalności.

Д О П О Л Н Е Н И Е

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ УПРАВЛЕНИЯ ДИНАМИЧЕСКОЙ СБОРКОЙ

Р е з ю м е

В дополнении рассматривается последовательная сборочная линия а также процесс сборки с точностью в одну операцию (одну деталь). Рассматривается процесс сборки в очередных циклах сборки на каждом рабочем месте сборочной станции. Для описания процесса введено определение состояния сборки. В частности определены: начальное и конечное состояния а также левостронный и правостронный краевые условия. Кроме этого принята классификация процесса сборки, выделяя процессы: статический, динамический и переходной. Аналогичным образом определено управление сборкой а также принята классификация управления с выделением управлений: статического, динамического и переходного. На основании определения состояния и управления, выделены уравнения состояния процесса сборки. Уравнения эти показывают логические зависимости при сборке на очередных рабочих местах в очередных циклах.

В дополнении выделены одноверсионные и многоверсионные процессы сборки. Детально проанализирован одноверсионный процесс. Выделенные формулы обобщены для случая многоверсионной сборки.

Рассматриваемые математические модели основываются на уравнениях состояния. Алгоритмы уравнения используют эвристические решающие правила а также симуляцию сборочного процесса.

Для одноверсионного процесса представлена детерминистическая и вероятностная модели сборки. Кроме этого рассмотрена детерминистическая модель сборки и запаса деталей. Для детерминистической модели дан анализ уравнений статического, псевдостатического, псевдодинамического и динамического. Приведены различные алгоритмы динамического управления.

В многоверсионной модели существенное значение имеет левостороннее краевое условие. Определяет оно очерёдность объектов (различных версий) введённых на сборочную линию. Для детерминистической многоверсионной модели сборки представлено динамическое управление.

Рассмотренные в дополнении математические модели и алгоритмы управления, основанные на уравнениях состояния, являются моделями симуляционного типа. Используя эти модели можно таким образом определить рациональное управление однако без гарантированной оптимальности. Представленные моделм могут быть использованы на сборочных линиях с промышленными роботами.

A P P E N D I X

MATHEMATICAL MODELS AND ALGORITHMS FOR CONTROL OF DYNAMIC ASSEMBLY

S u m m a r y

In the appendix a sequential assembly line and an assembly process is considered - with precision to the operation (assembly element). The assembly process is analysed in the succesive cycles on each work place (line station).

For description of the process the definition of an assembly state is introduced. The initial and final states end boundary conditions (the left and the right hand side) are distinguished. Futhermore the classification of the assembly process has been introduced. According to the clasification the static, dynamic and transient assembly processes are distinguished. In analogious way the control of the assembly process is defined. The state equations of the assembly process are found out basing on the appropriate definitions. The equations interpret logical dependences at the assembly on sequential line stations during succesive cycles.

In the appendix the single-version of assembly and its multi-version are distinguished. The single-version assembly process is analysed in details. The worked out formulas have been generalized for the multi-version assembly process.

The mathematical models considered in the appendix are based on the logical state equations. Control algorithms use heuristic decision rules and a simulation of the assembly process.

For the single-version process deterministic and stochastic models of the assembly are shown. Futhermore a deterministic model of the assembly and assembly element stocks has been considered. For the deterministic model the following controls have been analysed: static, pseudostatic, pseudodynamic and dynamic.

At the multi-version process has the left hand side boundary condition has got a key because it is controlable. The left hand side boundary condition interprets the sequence of various version object which are introduced on the assembly line. For the multi-version deterministic model of assembly the quasi-dynamic and dynamic controls are presented.

The mathematical models and control algorithms considered in the appendix are based on the state equations. These are the simulation systems which give feasible solutions without warrant of the optimality.

LITERATURA

[1] Adrabiński A., Wodecki M., Grabowski J.: Algorytm optymalizacji zagadnień sekwencyjnych w dyskretnych procesach produkcyjnych. ZN Pol. Śl., s. Automatyka, z. 46, Gliwice 1978, ss. 90-99.

[2] Agin N.: Optimum Seeking with Branch and Bound. Management Science, V. 13, No 4, 1966, pp. 176-185.

[3] Ameljańczyk A.: Metoda punktu idealnego w rozwiązywaniu zadań polioptymalizacji. ZN Pol. Śl., s. Automatyka, z. 67, Gliwice 1983, ss. 7-16.

[4] Ameljańczyk A.: Ogólne własności rozwiązań dominujących i niezdominowanych. ZN Pol. Śl., s. Automatyka, z. 67, Gliwice 1983, ss. 17-28.

[5] Ameljańczyk A.: Optymalizacja kolektywna. XXIII Sympozjon nt. "Modelowanie w mechanice", PTMTiS, Gliwice 1984, ss. 483-494.

[6] Ameljańczyk A.: Metoda szeregowania zbioru rozwiązań niezdominowanych w zadaniach ze skończonym zbiorem rozwiązań dopuszczalnych. XXIII Sympozjon nt. "Modelowanie w mechanice". PTMTiS, Gliwice 1984, ss. 495-508.

[7] Ameljańczyk A., Oleksiuk W.: Wpływ przyjętej metody polioptymalizacji na wybór rozwiązania konstrukcyjnego. XXIII Sympozjon nt. "Modelowanie w mechanice", PTMTiS, Gliwice 1984, ss. 509-520.

[8] Arcus A.L.: Assembly Line Balancing by Computer. Graduate School of Business, University of California, Berkeley 1962.

[9] Arcus A.L.: An Analysis of a Computer Method of Sequencing Assembly Line Operations. Ph.D. dissertation, University of California, Berkeley 1963.

[10] Arcus A.L.: COMSOAL - A Computer Method of Sequencing Operations for Assembly Lines. The International Journal of Production Research, V. 4, No 4, 1966 (również w Buffa E.S.: Readings in Production and Operations Management. New York-London-Sydney, John Wiley and Sons, 1966, pp. 259-277).

[11] Automatyzacja dyskretnych procesów przemysłowych. Materiały I Krajowej Konferencji ADDPP, ZN Pol. Śl., s. Automatyka z. 43-46, Gliwice 1978.

[12] Automatyzacja dyskretnych procesów przemysłowych. Materiały II Krajowej Konferencji ADPP, ZN Pol. Śl., s. Automatyka z. 54-56, Gliwice 1980.

[13] Automatyzacja dyskretnych procesów przemysłowych. Materiały III Krajowej Konferencji ADPP, ZN Pol. Śl., s. Automatyka z. 63, 64, Gliwice 1982.

[14] Automatyzacja dyskretnych procesów przemysłowych. Materiały IV Krajowej Konferencji ADPP, ZN Pol. Śl., s. Automatyka, z. 74-76, Gliwice 1984.

[15] Buda J., Kovač M.: Zastosowanie robotów przemysłowych. WNT, Warszawa 1979.

[16] Assembly Automation, Proceedings of the 5th International Conference organized by IFS, Paris, France, 1984.

[17] Balas E.: An Additive Algorithm for Solving Linear Programs with Zero-One Variables. Operations Research, V. 13, No 4, 1965, pp.517-546.

[18] Balas E.: Discrete Programming by the Filter Method. Operations Research, V. 15, No 5, 1967, pp. 915-957.

[19] Balas E.: Machine Sequencing via Disjunctive Graphs: An Implicite Enumeration Algorithm. Operations Research, V. 17, No 6, 1969, pp. 941-957.

[20] Bellman R.: Adaptacyjne procesy sterowania. PWN, Warszawa 1965, ss. 80-92.

[21] Bellman R., Dreyfus S.E.: Applied Dynamic Prgramming, Princeton University Press, 1962.

[22] Beck H.G.: The Influence of Assembly Line Organization on Output, Quality and Morale. Occupational Psychology, V. 38, 1984.

[23] Bielecki S.A., Bibik E.W.: Dwuparametriczeskaja zadacza bałansirowanija sborocznoj linii. Prieprint 75-30, Institut Kibiernietiki AN USSR, Kijew 1975.

[24] Bielecki S.A., Bibik E.W.: K woprosu ob optimalnom bałansje sborocznoj linii. Uprawlajuszczije sistiemy i masziny, No 4, Kijew 1975, ss. 117-121.

[25] Biolik L.: Opracowanie i zbadanie efektywności algorytmów harmonogramowania pracy wielu brygad utrzymania ruchu w wielu oddziałach wydobywczych KWK. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki Politechniki Śląskiej, Gliwice 1983.

[26] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Algorytmy sterowania rozdziałem zadań i zasobów w kompleksie operacji. WPP, Poznań 1978.

[27] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Badania operacyjne dla informatyków. WNT, Warszawa 1983.

[28] Bołtianski W.G.: Sterowanie optymalne układami dyskretnymi. WNT, Warszawa 1978.

[29] Bonney M.C., Schofield N.A.: Developments in the SAMMIE/AUTOMAT Computerised Work Study System. The MTM Journal, V. 1, No 1, 1974.

[30] Bowman E.M.: Assembly Line Balancing by Linear Programming. Operations Research, 1960, V. 8, No 3, pp. 385-389.

[31] Bryton B.: Balancing of a Continuous Production Line. Unpublished M.S. Thesis, Northwestern University, Evanston, Illinois 1954.

[32] Brennecke D.: Two Parameters Assembly Line Balancing Model (in: Models and Analysis for Production Management). International Textbook, Soranton, Pensylvania 1968.

[33] Bubnicki Z.: Problemy optymalnego sterowania kompleksami operacji. Konferencja nt. "Współczesne problemy automatyki i informatyki", Politechnika Śląska, Gliwice 1973.

[34] Bubnicki Z.: Identyfikacja obiektów sterowania. PWN, Warszawa 1974.

[35] Bubnicki Z.: Optymalizacja kompleksów operacji w sterowaniu dyskretnymi procesami produkcyjnymi. VII KKA, Rzeszów 1977, t. III, ss. 37-44.

[36] Burgeson J.W., Daum T.E.: Production Line Balancing. File No 10.3.002 IBM, Akron, Ohio 1958.

[37] Burkow W.N., Łowiecki S.E.: Mietody rieszenija ekstriemalnych kombinatornych zadacz. Tiechniczeskaja Kibiernietika, No 4, Moskwa 1968, ss. 82-93.

[38] Buxey G.M.: Assembly Line Balancing with Multiple Stations. Management Science, V. 20, No 6, 1974, pp. 1010-1021.

[39] Caruso F.R.: Assembly Line Balancing for Improved Profits. Automation 1965, V. 12, No 1, pp. 48-52.

[40] Cauley J.M.: A Review of Assembly Line Balancing Algorithms. Proceedings of 19-th Annual Conference, AIIE, New York 1968.

[41] Coffman E.G. jr (red.): Teoria szeregowania zadań. WNT, Warszawa 1980.

[42] Czernowa T.N.: Ob odnom mietodie sinchronizacji sborocznych potocznych linii. Ekonomika i matiematiczeskije mietody, Moskwa 1970, V. 6, No 4.

[43] Ćwik A.: Projekt wstępny struktury programowej i urządzeniowej systemu sterowania montażem mechanizmów samochodu małolitrażowego FIAT 126p. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Gliwice 1977.

[44] Ćwik A., Marecki F.: Algorytm optymalizacji procesu montażu silnika samochodu małolitrażowego FIAT 126p. Prace Studenckich Obozów Naukowo-Badawczych, Politechnika Śląska, Gliwice 1978, ss. 9-13.

[45] Christowa N.P.: Sinchronizacja sborocznogo processa. Primienienije matiematiki w ekonomikie, L.G.U., 1973, No 8.

[46] Christowa N.P.: Płanirowanije racjonalnoj organizacji sistiemnych processow sborocznogo tipa. Dissiertacja kandidata ekonomiczeskich nauk, L.F.E.U., Leningrad 1973.

[47] Dantzig G.B.: On the Shortest Route Through a Network. Management Science, 1960, V. 6, pp. 187-190.

[48] David F., Berry M.: Equilibrate des Chaines de Montage. Automatisme, 1969, V. 14, No 9, pp. 403-407.

[49] Dubina A.: Zastosowanie robotów przemysłowych w procesie montażu. ZN Pol. Śl., s. Automatyka, nr 56, Gliwice 1980, ss. 109-114.

[50] Duda J.: Dobór parametrów wybranego dyskretnego procesu przemysłowego metodą modelowania cyfrowego. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 11-18.

[51] Duda J.: PRZEMONT - system optymalnego przydziału zadań dla monterów na linii montażowej. ZN Pol. Śl., s. Automatyka, nr 55, Gliwice 1980, ss. 129-137.

[52] Dusza K., Kowalowski H., Marecki F.: Sterowanie dyspozytorskie obsługą robót spawalniczych na KWK. Sympozjum nt. "Systemy zarządzania i sterowania kopalniami", KG PAN, Szklarska Poręba 1979, ss. 164-174.

[53] Dusza K., Marecki F.: Automatisiertes Leitsystem für Schweissarbeiten im Schacht. ARS 81, Ostrava 1981.

[54] Dusza K.: Harmonogramowanie zadań remontowo-konserwacyjnych na wielu oddziałach KWK z różnymi priorytetami ich realizacji. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 29-37.

[55] Findeisen W., Szymanowski J., Wierzbicki A.: Teoria i metody obliczeniowe optymalizacji. PWN, Warszawa 1977.

[56] Ford L.R., Fulkerson D.R.: Przepływy w sieciach. PWN, Warszawa 1969.

[57] Garfinkel R.S., Nemhauser G.L.: Programowanie całkowito-liczbowe, PWN, Warszawa 1978.

[58] Geoffrion A.M.: Integral Programming by Implicite Enumeration and Balas Method. SIAM Review 1967, V. 9, No 2, pp. 178-190.

[59] Geoffrion A.M.: An Improved Implicit Enumeration Approach for Integer Programming. Operations Research, V. 8, No 3, pp. 385-389.

[60] Gertych J.: Analiza pracy węzła napraw na montażu głównym samochodów. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 17-28.

[61] Glover P.C., Norman J.T.: Assembly Line Balancing. Assembly Engineering, 1965, V. 8, No 10.

[62] Gosiewski A.: Teoria sterowania. VII KKA, Rzeszów 1977, t. III, ss. 50-54.

[63] Goetz J.: Modelowanie na grafie dysjunktywnym ogólnych zagadnień przydziału zasobów i szeregowania operacji. ZN Pol. Śl., s. Automatyka, nr 54, Gliwice 1980, ss. 33-46.

[64] Gościński A.: Modelowanie złożonych dyskretnych procesów produkcyjnych w oparciu o teorię automatów. Archiwum Automatyki i Telemechaniki, 1974, nr 4.

[65] Gościński A.: Przykłady modelowania dyskretnego procesu produkcyjnego w oparciu o teorię automatów. Archiwum Automatyki i Telemechaniki, 1975, nr 2.

[66] Gościński A.: Sterowanie złożonych dyskretnych procesów produkcyjnych w oparciu o system współdziałających automatów. Archiwum Automatyki i Telemechaniki, 1975, nr 2.

[67] Gościński A., Mysona-Byrska E., Nawracji E.: Ewolucyjny system sterowania dyskretnymi procesami produkcyjnymi, ZN Pol. Śl., seria: Automatyka, Z. 46, 1978, ss. 115-126.

[68] Górecki H.: Badania systemowe. VII KKA, Rzeszów 1977, t. III, ss. 193-194.

[69] Górecki H.: Problem of choise of an optimal solution in a multicriterion space. Preprinta of VIII World IFAC Congress, Kyoto 1981, X-106.

[70] Grabowski J.: Modele matematyczne i optymalizacja zagadnień sekwencyjnych w dyskretnych procesach produkcyjnych. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 29-38.

[71] Grabowski J.: Zagadnienia kolejnościowe w optymalizacji ciągów krytycznych w dyskretnych systemach produkcyjnych. ZN Pol. Śl., s. Automatyka, nr 54, Gliwice 1980, ss. 47-56.

[72] Gutjahr A.L.: An Algorithm for the Assembly Line Balancing Problem. Master's Thesis, The Johns Hopkins University, 1963.

[73] Gutjahr A.L., Nemhauser G.L.: An Algorithm for the Line Balancing Problem. Management Science, 1964, V. 11, No 2, p. 308-315.

[74] Hahn R., Lutz L., Roschmann K.: Die Bandabgleichung-ein Problem bie Fliessfertigung. Ablauf und Planungsforschung, 1963, No 4.

[75] Hajewski M.: Poszukiwanie efektywnego algorytmu harmonogramowania zadań w systemach o strukturze równoległej. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1984.

[76] Held M., Karp R.M.: A Dynamic Programming Approach to Sequencing Problems. SIAM Journal on Applied Mathematics, 1962, V. 10, No 1, pp. 192-210.

[77] Held M., Karp R.M., Sharesian R.: Assembly Line Balancing-Dynamic Programming with Precedence Constraints. Operations Research, 1963, V. 10, No 3, pp. 442-460.

[78] Held M., Karp R.M.: The Construction of Discrete Dynamic Programming Algorithms. IBM Systems Journal, 1965, V. 4, No 2, pp. 136-147.

[79] Helgeson W.P., Kwo T.T.: Letter to the Editor. Management Science, 1956, V. 3, No 1.

[80] Helgeson W.P., Birnie D.P.: Assembly Line Balancing Using the Ranked Positional Wieght Technique. The Journal of Industrial Engineering, 1971, V. 12, No 6, pp. 394-398.

[81] Heskiaoff H.: A Heuristic Method for Balancing Assembly Lines. Western Electric Engineer, 1968, V. 12, No 3, pp. 9-16.

[82] Hoffman T.R.: Permutations and Precedence Matrices with Automatic Computer Applications to Industrial Problems. Ph. D. dissertation, University of Wisconsin, Madison, 1959.

[83] Hoffman T.R.: Optimizing Criteria and Assembly Line Balancing by Digital Computer. 7-th Annual International Meeting of the Institute of Management Science, 1960.

[84] Hoffman T.R.: Assembly Line Balancing with a Precedence Matrix. Management Science, 1963, V. 9, No 4, pp. 551-563.

[85] Hu T.C.: Parallel Sequencing and Assembly Lines Problems. Operations Research, 1961, V. 9, No 6, pp. 841-848.

[86] Ibaraki T.: Theoretical Comparisons of Search Strategies in Branch and Bound Algorithms. International Journal of Computer and Information Science, 1976, No 4, pp. 315-344.

[87] Ignall E.J.: A review of Assembly Line Balancing. Journal of Industrial Engineering, 1965, V. 16, No 4, pp. 244-254.

[88] Ignasiak E.: Programowanie sieciowe. PWE, Warszawa 1975.

[89] Jackson J.R.: A Computing Procedure for a Line Balancing Problem. Management Science, 1956, V. 2, No 3, pp. 261-272.

[90] Jackson J.R.: A Computing Procedure for a Line Balancing Problem. Management Science, 1963, V. 9, pp. 551-563.

[91] Jaeschke G.: Eine Algemaine Methode Zur Lösung Kombinatorischer Probleme. Ablauf und Planungsforschung, 1964, V. 5, pp. 626-638.

[92] Jamroż L.: O pewnym zastosowaniu metody podziału i ograniczeń do rozwiązywania problemu szeregowania zadań. ZN Pol. Śl., s. Automatyka nr 63, Gliwice 1982, ss. 19-28.

[93] Janasz-Kusidło G.: Optymalizacja kolejności montażu różnych wersji samochodów FIAT 126p. Praca dyplomowa magisterska (niepublikowana). Instytut Automatyki, Pol. Śl., Gliwice 1980.

[94] Jankowska-Zorychta Z.: Modele sekwencyjne i ich zastosowania w planowaniu optymalnej organizacji w dyskretnych procesach produkcyjnych. PWN, Warszawa 1973.

[95] Jońca A., Trybuś G.: Heurystyczna metoda synchronizacji czasów operacji w liniach montażowych. Mechanik, No 3, 1978, ss. 137-139.

[96] Jońca A.: Koncentracja i operacyjny podział robót w projektowaniu systemów organizacyjnych montażu. Prace naukowe AE, s. Organizacja produkcji i pracy, nr 248, Wrocław 1983, ss. 5-15.

[97] Jońca A., Bednarz L.: Złożone przypadki synchronizacji operacji w potokowych liniach montażowych. Prace naukowe AE, s. Organizacja Produkcji i Pracy, nr 248, Wrocław 1983, ss. 17-38.

[98] Jońca A.: Kompleksowo-systemowe projektowanie rozwiązań organizacyjnych montażu, Przegląd Organizacji No 3, 1979.

[99] Jermoliew J., Melnik I.M.: Ekstremalnyje zadaczi na grafach. Naukowa dumka, Kijew 1968, ss. 142-144.

[100] Jarosz H.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemach o strukturze drzewa i antydrzewa. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1983.

[101] Jurczyk Z., Marecki F.: Harmonogramowanie procesu wytłaczania blach karoseryjnych. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 26-33.

[102] Jurczyk Z.: Model procesu wytłaczania blach karoseryjnych i koncepcja heurystycznego algorytmu harmonogramowania. ZN Pol. Śl., s. Automatyka, nr 35, Gliwice 1980, ss. 153-162.

[103] Jurczyk Z., Szendzielorz K.: System harmonogramowania procesu wytłaczania blach karoseryjnych. VIII KKA, Szczecin 1980.

[104] Jurczyk Z.: Kaliendarnoje płanirowanije roboty sztampowocznoj linii. ARS'81, Ostrava 1981.

[105] Jurczyk Z.: Harmonogramowanie zadań na linii technologicznej. Konferencja nt.: "Metody i środki projektowania automatycznego", Instytut Podstaw Budowy Maszyn Pol. Warszawskiej, Warszawa 1981.

[106] Jurczyk Z., Marecki F.: Oiscrete Process Scheduling in a System with Series Structure. 4-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981.

[107] Jurczyk Z.: Kaliendarnoje płanirowanije roboty sztampowocznoj linii s uczotom fiksirowannych riesursnych ograniczenij. 5-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981.

[108] Jurczyk Z.: Optymalne harmonogramowanie procesu wytłaczania blach karoseryjnych. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 35-44.

[109] Kałuski J. i inni: Laboratorium rachunku prawdopodobieństwa i statystyki matematycznej. Skrypt nr 1172, Gliwice 1984.

[110] Kałuski J.: Niezawodność harmonogramowania zadań na agregacie. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 53-64.

[111] Kałuski J.: Nadiożnost dieterministiczeskich grafikow razpisanija robot s uczotom słuczajnych pomiech. ARS'84, Ostrava 1984.

[112] Kałuski J.: Ocena harmonogramów ze względu na kryterium niezawodności. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 45-52.

[113] Kałuski J., Marecki F.: Kryterium niezawodności w polioptymalnym harmonogramowaniu, 24 Sympozjon "Modelowanie w Mechanice", PTMTiS, Gliwice 1985, ss. 597-603.

[114] Kałuski J., Marecki F.: Niezawodność jako kryterium polioptymalizacji procesów. Konferencja nt.: "X dni jakości i niezawodności", NOT, Gliwice 1984.

[115] Kaczmarski S.: Przegląd systemów sterowania produkcją u wybranych producentów samochodów. Forum Automatyki '78, FSO, Warszawa 1978, t. I, ss. 87-102.

[116] Kao E.P.C.: A Preference Order Dynamic Program for stochastic Assembly Line Balancing. Management Science, 1976, V. 22, No 10, pp. 164-171.

[117] Kao E.P.C.: Comments on "Shortest Route Models for Allocation of Inspection Effort on a Production Line", by L.S. White. Management Science, 1976, V. 22, No 10, p. 1168.

[118] Kempny B.: Algorytm harmonogramowania produkcji struktur półprzewodnikowych. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1982.

[119] Kiecka M.: Koncepcja sterowania transportem elementów karoserii na linię montażu ciągnika MF. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1983.

[120] Kilbridge M.D., Wester L.: The Assembly Line Problem. Proceedings of the 2-rd International Conference on Operational Research, London, English Universities Press, 1961, pp. 213-224.

[121] Kilbridge M.D., Wester L.: A Heuristic Method of Assembly Line Balancing. The Journal of Industrial Engineering, 1961, V. 12, No 4, pp. 292-298.

[122] Kilbridge M.D., Wester L.: The Balans Delay Problem. Management Science, 1961, V. 8, No 1, pp. 69-84.

[123] Kilbridge M.D.: A Model for Industrial Learning Costs. Management Science, 1962.

[124] Kilbridge M.D.: Non Productive Work as a Factor in the Economic Division of Labour. The Journal of Industrial Engineering, 1961, V.12, No 3, pp. 155-159.

[125] Kilbridge M.D., Wester L.: Analitic Systems of Line Balancing. 20-th National Meeting of the Operations Research Society of America, San Francisco, 1961.

[126] Kilbridge M.D., Wester L.: A Review of Analitical Systems of Line Balancing. Operations Research, 1962, V. 10, No 5, pp. 626-638.

[127] Kilbridge M.D., Wester L.: An Economic Model for the Division of Labour. Management Science, 1966, V. 12, No 6, pp. 255-269.

[128] Kimmel M.: Analiza i modelowanie nieustalonego procesu montażu taśmowego. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 45-52.

[129] Kimmel M.: Zastosowanie metody typu Monte Carlo do optymalizacji sekwencji w problemie kolejnościowym. ZN Pol. Śl., s. Automatyka, nr 50, Gliwice 1980.

[130] Kimmel M.: Zastosowanie teorii procesów punktowych do projektowania węzła montażu. ZN Pol. Śl., s. Automatyka, nr 54, Gliwice 1980, ss. 89-98.

[131] Kimmel M.: Zastosowanie teorii procesów Markowa do badania granicznych procesów dyskretnych. ZN POl. Śl., s. Automatyka, nr 63, Gliwice 1982, ss. 41-50.

[132] Kislicyn S.S.: Czastliczno uporiadoczennyje mnożestwa i K-rozszczeplajemost. Sybirskij Matiematiczeskij Żurnał, 1967, V. 8, No 5, ss. 1197-1201.

[133] Kislicyn S.S.: Koniecznyje czastliczno uporiadoczennyje mnożestwa i sootwietstwujuszczije im mnożestwa pieriestanowok. Matiematiczeskije zamietki, 1968, V. 4, No 5.

[134] Klein M.: On Assembly Line Balancing. Operations Research, 1963, V. 11, No 2, pp. 274-281.

[135] Klein M., Thakamori H.: Parallel Line Assignment Problems. Management Science, 1972, V. 19, No 1, pp. 1-10.

[136] Kohler H.W., Steiglitz K.: Przeglądowe i iteracyjne metody obliczeniowe. W "Teoria szeregowania zadań" - red. Coffman E.G., jr, WNT, Warszawa 1980, ss. 241-301.

[137] Korbut A.A., Finkelsztejn J.J.: Programowanie dyskretne. PWN, Warszawa 1974.

[138] Kottas J.F., Lau H.S.: A Cost-Oriented Approach to Stochastic Line Balancing. AIIE Transactions, 1973, V. 5, No 2, pp. 164-171.

[139] Kowalowski H.: Modelowanie w operatywnym sterowaniu produkcją dyskretnych procesów przemysłu maszynowego. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 53-59.

[140] Kowalowski H.: Ważniejsze zasady projektowania i wdrażania automatycznych systemów sterowania dyskretnymi procesami przemysłowymi wspomaganymi komputerowo. ZN Pol. Śl., s. Automatyka, nr 55, ss. 163-174.

[141] Kowalowski H. i inni: System sterowania SILMONT. Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1977.

[142] Kowalowski H. i inni: System sterowania PRZEMONT. Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1980.

[143] Kowalowski H.: Osobliwości elastycznej automatyzacji dyskretnych procesów przemysłowych drogą robotyzacji. ZN Pol. Śl., s. Automatyka nr 64, Gliwice 1982, ss. 201-216.

[144] Kowalowski H., Marecki F., Starzyczny L. Dusza K.: An Algorithm of Independent Tasks Scheduling in a System with Dispersed Parameters. 4-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981, V. III, pp. 8-16.

[145] Kowalowski H. i inni: Sterowanie operatywne montażem ciągników ciężkich. Raport z pracy n-b, Instytut Automatyki, Pol. Śl., Gliwice 1982.

[146] Kowalowski H. i inni: Automatyzacja dyskretnych procesów przemysłowych. WNT, Warszawa 1984.

[147] Kowalowski H.: Naucznaja problematika robotizacji kak nieobchodimost dopołnienija awtomatizacji diskretnych promyszliennych processow. ARS'84, Ostrava 1984.

[148] Kowalowski H., Marecki F.: Informatyczny system kierowania linii montażowej samochodu FIAT 126p. Konferencja "INFOGRYF'78", TNOiK, Kołobrzeg 1978, ss. 168-180.

[149] Krowiak A.: Synchronizacja dostaw silników wersyjnych do linii montażu samochodów PF 126. ZN Pol. Śl., s. Automatyka, nr 56, Gliwice 1980, ss. 47-51.

[150] Ksyta J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych realizowanych w jednym agregacie. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1983.

[151] Kućmierz M.: Analiza procesu montażu samochodu małolitrażowego FIAT 126p. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1978.

[152] Kulikowski R.: Sterowanie w wielkich systemach. WNT, Warszawa 1970.

[153] Kurnal J. (red.): Teoria organizacji i zarządzania. PWE, Warszawa 1979.

[154] Lach H.: Modele harmonogramowania i sterowania montażem podwozi ciągników MF w ZM URSUS. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1981.

[155] Land A.H., Doig A.G.: An Automatic Method of Solving Discrete Programming Problems. Econometrics, 1960, V. 28, No 3, pp. 497-520.

[156] Lehman M.L.: What's Going on in Products Assembly. The Journal of Industrial Engineering, 1966, No 4, pp. 41-45.

[157] Mann A.S.: On the Job-Shop Scheduling Problem. Operations Research, 1960, V. 8, No 2, pp. 219-223.

[158] Magazine M.J., Wee T.S.: The generation of Bin-Packing Heuristics to the Line Balancing Problem. Department of Management Sciences, University of Waterloo, Ontario, Working Paper No 128, 1979.

[159] Magazine M.J., Wee T.S.: An Efficient Branch and Bound Algorithm for the Assembly Line Balancing Problem. Department of Management Science, University of Waterloo, Ontario, Working Paper, 1979.

[160] Mansoor E.M.: Assembly Line Balancing - An Improvement on the Ranked Positional Weight Technique. The Journal of Industrial Engineering, 1964, V. 15, No 2.

[161] Mansoor E.M.: Assembly Line Balancing - Extension and Discusion. The Journal of Industrial Engineering, 1964, V. 15, No 6.

[162] Mansor E.M.: Improvement on Gutjahr and Nemhauser's Algorithm for the Line Balancing Problem. Management Science, 1967, V. 14, No 3, pp. 250-254.

[163] Mansoor E.M.: MALB-A Heuristic Technique for Balancing Large Single-Model Assembly Lines. AIIE Transactions, 1973, pp. 343-356.

[164] Marecki F.: Modelowanie symulacyjne linii montażowej samochodu małolitrażowego. Informatyka, No 7-8, 1975, ss. 25-28.

[165] ,-,: Analiza możliwości optymalizacji pracy linii montażowej silnika samochodu małolitrażowego FIAT 126p przy zastosowaniu S/7 IBM. Konferencja nt.: "Sterowanie procesem produkcyjnym przy zastosowaniu IBM S/7", FSM Bielsko-Biała, Jaszowiec, 1976, ss. 51-63.

[166] ,-,: Algorytm sterowania jednostrefowych linii montażowych ze stanowiskami podmontaży. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 69-78.

[167] ,-,: Assembly Line Control. ARS'78, Ostrava 1978, V. 1, pp. 122-130.

[168] ,-,: Automatyzacja planowania i sterowania procesu montażu na liniach. Konferencja nt.: "Automatyzacja planowania i zarządzania przebiegiem procesu produkcyjnego", FSS Kielce, Ameliówka, 1978.

[169] ,-,: System sterowania linii montażu silnika samochodu FIAT 126p. Konferencja SYSTER'79, NOT, Katowice 1979, cz. I, ss. 109-120.

[170] ,-,: Projektowanie procesu montażu na liniach. Konferencja nt.: "Zastosowanie elektronicznych maszyn cyfrowych w pracach inżynierskich", NOT, Katowice 1979, ss. 5-13.

[171] ,-,: Metoda balansowania wielostrefowej linii montażowej. Konferencja nt.: "Metody i środki projektowania automatycznego", Instytut Podstaw Budowy Maszyn, Politechnika Warszawska, Warszawa 1979, ss. 213-221.

[172] ,-,: Harmonogramowanie dostaw detali w procesie montażu. Konferencja nt.: "Cybernetyka w gospodarce morskiej", WSM, Gdynia 1979, cz. II, ss. 59-68.

[173] ,-,: Harmonogramowanie procesu montażu wielowersyjnego. Konferencja nt.: "VIII dni jakości i niezawodności", NOT, Gliwice 1979.

[174] ,-,: Probabilistyczna analiza jakości procesu montażu na liniach. Konferencja nt.: "VIII dni jakości i niezawodności", NOT, Gliwice 1979.

[175] ,-,: O uprawlienii kaczestwom processa sborki na liniach. 3-rd International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1979, V. III, pp. 917-939.

[176] ,-,: Symulacja cyfrowa przejściowego procesu montażu na linii. VIII KKA, Szczecin 1980, t.I.

[177] ,-,: Harmonogramowanie dostaw detali na linię montażową. ZN Pol. Śl., s. Automatyka, nr 55, Gliwice 1980, ss. 51-61.

[178] ,-,: Problemy sterowania procesem montażu na liniach. Konferencja nt.: "Nowe metody organizacji produkcji i pracy w montażu wyrobów", SIMP, Wrocław 1980.

[179] ,-,: Problem kolejnościowy w montażu wielowersyjnym. ZN Pol. Śl., s. Automatyka, nr 54, Gliwice 1980, ss. 119-128.

[180] ,-,: Simulation Model of the Assembly Line. SIMULA'80, HAS, Budapest 1980.

[181] ,-,: Modelowanie cyfrowe procesu montażu silnika samochodu FIAT 126p. Konferencja nt.: "Zastosowanie komputerów w przemyśle", NOT, Szczecin 1978, t. 4, ss. 29-42.

[182] ,-,: Static Control of Tasks Allocation on Assembly Line. 4-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1981, V. III, pp. 17-23.

[183] ,-,: Sterowanie obsługą statków w porcie. Konferencja nt.: "Cybernetyka w gospodarce morskiej", WSM, Gdynia 1981, t. 2, ss. 85-96.

[184] Marecki F., Starzyczny L.: Uprawlienije processami dobyczi na podziemnoj szachtie. Konferencja nt.: "Awtomatizacja gornodobywajuszczijej promyszliennosti", Nauczno-Tiechniczeskije Sojuzy Bałgarii, Warna 1981.

[185] Marecki F., Dusza K.: Centralizowannoje dispietczerskoje uprawlienije riemontnymi i konserwacjonnymi robotami. Konferencja nt.: "Awtomatizacja gornodobywajuszczijej promyszliennosti", Nauczno-Tiechniczeskije Sojuzy Bałgarii, Warna 1981.

[186] Marecki F.: Leitung der Quellenverteilung im System Technologisher Streken. ARS'81, Ostrava 1981.

[187] ,-,: Assembly Line Balancing Problem. Prace Naukowe Instytutu Cybernetyki Technicznej, nr 63, seria: Konferencje, nr 22, System Science VII, Politechnika Wrocławska, Wrocław 1981, ss. 136-138.

[188] ,-,: An Algorithm of Dependent Tasks Scheduling in a System with Dispersed Parameters. Prace Naukowe Instytutu Cybernetyki Technicznej, nr 63, seria: Konferencje, nr 22, System Science VII, Politechnika Wrocławska, Wrocław 1981, ss. 132-135.

[189] ,-,: Balansowanie szeregowej linii montażowej z ograniczeniami wykluczenia operacji. ZN Pol. Śl., s. Automatyka, nr 63, Gliwice 1982, ss. 81-90.

[190] ,-,: Harmonogramowanie załadunku kontenerów w przemysłowym magazynie wysokiego składowania. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 69-82.

[191] Marecki F., Ślesińska I.: Problem harmonogramowania procesu lakierowania karoserii. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 83-94.

[192] Marecki F., Zielińska-Król E.: Model matematyczny i algorytm harmonogramowania procesu regeneracji walców. ZN Pol. Śl., s. Automatyka nr 64, Gliwice 1982, ss. 95-108.

[193] Marecki F.: Control of Discrete Process. 5-th International Conference on "Control System and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[194] Marecki F.: Modele matematyczne i algorytmy sterowania dyskretnymi procesami przemysłowymi na liniach montażowych. Instytut Automatyki (niepublikowane), Politechnika Śląska, Gliwice 1979.

[195] ,-,: Model sterowania przeładunkiem statków w porcie. Konferencja nt.: "Cybernetyka w gospodarce morskiej", WSM, Gdynia 1983, t. II, ss. 104-112.

[196] ,-,: Wielokryterialne balansowanie linii montażowej. Sympozjon nt.: "Modelowanie w mechanice", PTMTiS, Gliwice 1984, ss. 555-563.

[197] ,-,: Mnogokriterialnoje bałansirowanije sborocznoj linii. ARS'84, Ostrava 1984.

[198] ,-,: Model matematyczny mikrokomputerowego systemu sterowania grupę robotów przemysłowych na linii montażowej. Konferencja nt.: "Mikrokomputery w automatyce i technice systemów", Instytut Sterowania i Techniki Systemów, Politechnika Wrocławska, Wrocław 1984.

[199] ,-,: Algorytmy programowania wieloetapowego. Konferencja nt.: Zastosowanie matematyki", IM PAN, Sielpia 1984, ss. 46-47.

[200] ,-,: Metoda programowania wieloetapowego. ZN Pol. Śl., s. Automatyka, nr 74, ss. 159-178.

[201] ,-,: Balansowanie linii montażowej z enklawami operacji. ZN Pol. Śl., s. Automatyka, nr 73, Gliwice 1984.

[202] ,-,: Harmonogramowanie procesu lakierowania karoserii samochodowych. ZN Pol. Śl., s. Automatyka, nr 73, Gliwice 1984.

[203] ,-,: Alokacja operacji na równoległej linii montażowej. ZN Pol. Śl., s. Automatyka, nr 74, Gliwice 1984, ss. 179-194.

[204] ,-,: Harmonogramowanie ruchu statków w porcie z lokalnymi basenami buforowymi, Konferencja nt.: "Cybernetyka w gospodarce morskiej", WSM, Gdynia 1985.

[205] ,-,: Dwukryterialne sterowanie procesem dyskretnym realizowanym na jednym agregacie. ZN Pol. Śl., s. Automatyka, nr 76 Gliwice 1984, ss. 75-86.

[206] ,-,: Balansowanie linii montażowej z ograniczeniami czasowymi. ZN Pol. Śl., s. Automatyka (w druku).

[207] ,-,: Uogólniony model balansowania linii montażowej. Matematyka Stosowana (złożone w redakcji).

[208] ,-,: Balansowanie linii montażowej z frontami pracy. Archiwum Automatyki i Telemechaniki (złożone w redakcji).

[209] ,-,: Polioptymalna alokacja zasobów linii montażowej. Postępy Cybernetyki, t. 3, 1985.

[210] ,-,: Balansowanie asymetrycznej linii montażowej. Podstawy Sterowania, t. 15, z. 3.

[211] ,-,: Alokacja operacji i monterów na linii montażowej, ZN Pol. Śl., s. Automatyka, (złożone w redakcji).

[212] ,-,: Balansowanie linii montażowej z ograniczeniami pozycyjnymi. Przegląd Statystyczny (złożone w redakcji).

[213] ,-,: Generalized Assembly Line Balancing Problem. Conference on: Optimization - Theory, methods, applications, CVS, Praha, 1985.

[214] ,-,: Harmonogramowanie dyskretnych procesów przemysłowych. Skrypt Uczelniany Politechniki Śląskiej (w przygotowaniu).

[215] ,-,: Mikroprocesornaja sistema uprawlienija technołogiczeskoj liniej osnaszczennoj promyszliennymi robotami. ARS'84, Ostrava 1984.

[216] Mariotti J.: Four Approaches to Manuel Assembly Line Balancing. The Journal of Industrial Engineering, 1970, V. 2, No 6, pp. 35-40.

[217] Masłow W., Kalinin E., Muchin A., Koriukin E.: Normirowanije truda na konwiejernych liniach sborki s ispolzowanijem matiematiczeskich mietodow i ECWM. Socjalisticzeskij trud, 1970, No 10, ss. 106-112.

[218] Mastor A.A.: An Experimental Investigation and Comparative Evalution of Production Line Balancing Techniques. Ph. D. dissertation, University of California, Los Angeles, 1966.

[219] Mastor A.A.: An Experimental Investigation and Comparative Evalution of Production Line Balancing Techniques. Management Science, 1970, V. 16, No 11, pp. 728-746.

[220] Martens P.: Fliessbandabstimmung mit dem Verfahren der begrenzten Enumeration nach MÜller-Merbach. Ablauf und Planungforschung, 1967, V. 8, H. 4, ss. 429-433.

[221] Mikucka B.: Modele i algorytmy sterowania magazynami typu karuzelowego. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1984.

[222] Mitchell J.: A Computational Procedure for Balancing Zoned Assembly Lines. Research Report 6-94801-1-R3, Westinghouse Research Laboratories, Pittsburgh, Pensylvania, 1956.

[223] Moodie C.L.: A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work-Element Times. Ph.D. dissertation, Purdue University, 1964.

[224] Moodie C.L., Young H.H.: A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times. The Journal of Industrial Engineering, 1965, V. 16, No 1, pp. 23-29.

[225] Moodie C.L., Mandeville D.E.: Project Resource Balancing by Assembly Line Balancing Techniques. The Journal of Industrial Engineering 1966, V. 17, pp. 377-383.

[226] Moodie C.L.: Customized Assembly Line Balancing. The Journal of Industrial Engineering, 1973 (Aug), pp. 10-13.

[227] Muther R.: Production Line Technique. New York, N.Y., Mc Graw-Hill Book Company, 1944.

[228] Misiewicz B.: Analiza projektowa jakości montażu na liniach. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1980.

[229] Nevis A.J.: Assembly Line Balancing using Best Bud Search. Management Science, V. 18, 1972, pp. 529-539.

[230] Niederliński A.: Systemy cyfrowe automatyki przemysłowej. WNT, Warszawa 1977.

[231] Niederliński A.: Roboty przemysłowe. WSziP, Warszawa 1981.

[232] Niederliński A.: Komputerowe systemy sterowania magazynów wysokościowych. ZN Pol. Śl., s. Automatyka, nr 45, Gliwice 1978, ss. 96-103.

[233] Niederliński A.: Harmonogramowanie produkcji a wielopoziomowe wielowymiarowe dyskretne układy regulacji nadężnej. ZN Pol. Śl., s. Automatyka, nr 55, Gliwice 1980, ss. 63-70.

[234] Niederliński A.: Wielowymiarowe sterowanie ruchu ramion redundantnych manipulatorów w robotach generacji 2 i generacji wyższych. ZN Pol. Śl., s. Automatyka, nr 56, Gliwice 1980, ss. 147-153.

[235] Niepokój J.: Analiza algorytmów problemu przydziału z ograniczeniami czasowymi. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1984.

[236] Patterson J.H., Albracht J.J.: Assembly Line Balancing: Zero-One Programming with Fibonacci Search. Operations Research, 1975, V. 23, No 1, pp. 166-172.

[237] Pawlik S., Marecki F.: Algorytm rozdziału i obsługi zadań na linii montażowej. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 49-54.

[238] Peschel M., Riedel C.: Polioptymalizacja. WNT, Warszawa 1979.

[239] Piasecki S.: Optymalizacja procesów produkcyjnych. W "Teorie i zastosowania wielkich systemów", Wrocław 1972.

[240] Piasecki S.: Badania operacyjne. VII KKA, Rzeszów 1977, t. III, ss. 189-192.

[241] Piasecki S.: Operatywne kierowanie pracę linii montażowych. Prace IBS PAN, Warszawa 1981, ss. 3-20.

[242] Popow G.W.: Osobiennosti primienienija promyszliennych robotow pri awtomatizecji sborki. MONTEX-76, Bratysława 1976, pp. 246-251.

[243] Pogorzelski W.A.: Klasyczny rachunek kwantyfikatorów - zarys teorii. PWN, Warszawa 1981.

[244] Potrzebowski S.: Model i algorytmy szeregowania robót montażowych w budownictwie. Prace IBS PAN, Warszawa 1981, ss. 21-30.

[245] Prejsnar J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemach o strukturze równoległej. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki Pol. Śl., Gliwice 1983.

[246] Przegędza P.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemie o strukturze szeregowo-równoległej. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1984.

[247] Przybysz T.: Balansowanie szeregowej linii montażowej. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1981.

[248] Przybysz T., Jurczyk Z., Pawlik S., Marecki F.: System sterowania montażem podwozi ciągnika licencyjnego. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 109-114.

[249] Przybysz T.: Industrial High Storage Control. 5-th International Conference on: "Control System and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[250] Przybysz T.: Polioptymalne harmonogramowanie w magazynach wysokiego składowania. XXIII Sympozjon nt: "Modelowanie w mechanice", PTMTiS, Gliwice 1984, ss. 593-600.

[251] Przybysz T.: Kaliendarnoje planirowanije robot promyszliennogo wysokoetażnogo magazina. ARS'84, Ostrava 1984.

[252] Przybysz T., Kałuski J.: Analiza modelu niezewodności harmonogramowania zadań na agregacie. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 101-110.

[253] Przybysz T.: Harmonogramowanie transportu wewnątrzzakładowego, na przykładzie tłoczni. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 111-124.

[254] Pyszny H.: Sterowanie kompleksem operacji wykonywanych przez wiele brygad na jednym oddziale KWK. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki Pol. Śl., Gliwice 1982.

[255] Rasiowa H.: Wstęp do matematyki współczesnej. PWN, Warszawa 1975.

[256] Reeve N.R., Thomas W.H.: Balancing Stochastic Assembly Lines. AIIE Transactions, V. 5, No 3, 1973, pp. 223-229.

[257] Ross G.T., Soland R.M.: A Branch and Bound Algorithm for Generalized Assignment Problem. Mathematical Programming, 1975, V. 8, No 1, pp. 91-103.

[258] Salveson M.E.: The Assembly Line Balancing Problem. The Journal of Industrial Engineering, 1955, V. 6, pp. 18-25.

[259] Salveson M.E.: The Assembly Line Balancing Problem. Transactions of the American Society of Mechanical Engineers, 1955, V. 77, No 16, pp. 939-947.

[260] Sawczuk J.: Sterowanie jakością detali dopływających do linii montażowej przy wielowersyjnym montażu samochodu FIAT 126p. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1980.

[261] Sawik T.: Sterowanie optymalne harmonogramem produkcji o wielu urządzeniach i wielu wyrobach. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 63-70.

[262] Sawyer J.H.F.: Line Balancing. Machinery Publishing Co., Brighton, 1970.

[263] Serafin J.: Ocena efektywności wybranych algorytmów harmonogramowania na przykładzie tłoczni blach karoseryjnych. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1982.

[264] Słomiński L.: Zadania kombinatoryczne z minimaksowym kryterium. ZN Pol. Śl., s. Automatyka nr 46, Gliwice 1978, ss. 143-150.

[265] Sołtys W.: Osnownyje woprosy tiechniki integrirowanija processa pitanija sborocznych systiem. MONTEX-76, Bratysława 1976, pp. 81-86.

[266] Stanicki T.: Zagadnienia optymalizacji dyskretnych systemów produkcyjnych. VII KKA, Rzeszów, 1977, t. III, ss. 127-140.

[267] Słowiński R.: Metody wielokryterialnego programowania liniowego - próba syntezy. XXI Sympozjon nt.: "Modelowanie w mechanice", PTMTiS Gliwice 1982, ss. 389-424.

[268] Słowiński R.: Modele preferencji w wielokryterialnych problemach decyzyjnych. ZN Pol. Śl., s. Automatyka, nr 67, Gliwice 1983, ss. 95-112.

[269] Starzyczny L.: Problem harmonogramowania pracy brygad utrzymania ruchu na jednym oddziale KWK. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 129-135.

[270] Staszewski A.: Harmonogramowanie procesu walcowania ciągłego z uwzględnieniem współpracy z systemem walcowni finalnych. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1983.

[271] Staszulonek A.: Wybrane zagadnienia analizy procesu przejściowego w linii montażowej. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 71-78.

[272] Staszulonek A.: Algorytm eliminowania procesu przejściowego w linii montażowej. ZN Pol. Śl., s. Automatyka, nr Gliwice 1986, ss.

[273] Stokłosa J.: Harmonogramowanie pracy przemysłowego magazynu wysokiego składowania. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1982.

[274] Strykowski A.: Zagadnienie wdrożenia systemu sterowania pracą taśmy montażowej silników na przykładzie FSM. ZN Pol. Śl., s. Automatyka nr 46, Gliwice 1978, ss. 74-79.

[275] Szendzielorz K., Marecki F.: Koncepcja harmonogramowania pracy krajalni w FSM Tychy. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 79-86.

[276] Szendzielorz K.: Kalienc rnoje płanirowanije riezanija karoserijnych listow. ARS'81, Ostrava, 1981.

[277] Szendzielorz K.: Harmonogramowanie zadań na maszynach równoległych. Konferencja nt.: "Metody i środki projektowanie automatycznego". Instytut Podstaw Budowy Maszyn, Pol. Warszawska, Warszawa 1981.

[278] Szkurba W.W., Bieleckij S.A.: Czisliennyje mietody w rieszenii zadaczi bałansirowanija sborocznoj linii. Kibiernetika, 1977, No 1, ANUSSR, ss. 96-108.

[279] Szkurba W.W., Bibik E.W., Bieleckij S.A.: Projektorowanije organizacji konwiejernych linii w sborocznom proizwodstwie. MONTEX-76, Bratysława 1976.

[280] Szwarc J.: Model matematyczny i algorytm harmonogramowania brygad utrzymania ruchu w wielu oddziałach wydobywczych KWK. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki Pol. Śl., Gliwice 1982.

[281] Szymura J.: Optymalizacja harmonogramu produkcji linii produkcyjnej. Podstawy Sterowania, 1979, t. 9, z. 2, ss. 177-188.

[282] Szymura J., Pietraszek W.: Komputerowy system harmonogramowania produkcji zakładu karoserii FSM-02. ZN Pol. Śl., s. Automatyka, nr 55, Gliwice 1980, ss. 111-118.

[283] Ślesińska I., Marecki F.: Niezawodność funkcjonowania systemu transportu szynowego. Konferencja nt.: "X dni jakości i niezawodności", NOT, Gliwice 1981.

[284] Talbot B.F.: An Integer Programming Algorithm for the Single Model Assembly Line Balancing Problem. University of Michigan, 1978, Working Paper No 171.

[285] Talbot B.F., Patterson J.H.: An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems. Management Science, 1978, V. 24, No 11, ss. 1163-1174.

[286] Tanajew W.S., Szkurba W.W.: Wwiedienije w tieoriju rospisanij. Nauka, Moskwa 1975.

[287] Thangavelu S.R., Shetty C.M.: Assembly Line Balancing by Zero-One Integer Programming. AIIE Transactions, 1971, V. 3, pp. 61-68.

[288] Thangavelu S.R.: Assembly Line Balancing by 0-1 Integer Programming. Masters Thesis, Georgia Institute of Technology, 1970.

[289] Thomopoulos N.T.: Sequencing Procedure for Multi-Model Assembly Lines. Doctoral Thesis, Industrial Engineering Departament, Illinois Institute of Technology, 1966.

[290] Thomopoulos N.T.: Line Balancing-Sequencing for Mixed-Model Assembly Management Science, 1967, V. 14, No 2.

[291] Thomopoulos N.T.: Some Analitical Approaches to Assembly Line Problems. The Production Engineer, 1968.

[292] Thomopoulos N.T.: Mixed-Model Balancing with Smothed Station Assignments. Management Science, 1970, V. 16, No 9.

[293] Tonge F.M.: An Assembly Line Balancing Procedure. Management Science, 1960, V. 7, No 1, pp. 21-42.

[294] Tonge F.M.: A Heuristic Program of Assembly Line Balancing. Englewood Cliffs, Prentice-Hall, New Jersey, 1961.

[295] Tonge F.M.: The Use of Heuristic Programming in Management Science. Management Science, 1961, V. 7, No 3.

[296] Tonge F.M.: Krótki opis heurystycznych metod bilansowania linii montażowej; maszyny matematyczne i myślenie (pod redakcją J. Feldmana i E.A. Feigenbauma), PWN, Warszawa 1968, ss. 177-203.

[297] Tonge F.M.: Assembly Line Balancing Using, Probabilistic Combinations of Heuristic. Management Science, 1965, V. 11, No 7.

[298] Tonge F.M.: Summary of a Heuristic Line Balancing Procedure. Management Science, 1960, V. 1, No 1, pp. 21-42.

[299] Torońska-Łuczyńska M.: Możliwości optymalizacji dyskretnych technologii wytwarzania metodą symulacji cyfrowej. Praca doktorska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1980.

[300] Torońska-Łuczyńska M.: Wykorzystanie modelowania cyfrowego do sterowania procesem kucia. ZN Pol. Śl., s. Automatyka, nr 43, Gliwice 1978, ss. 61-68.

[301] Torońska-Łuczyńska M.: Harmonogramowanie procesu kucia metodą symulacji cyfrowej. ZN Pol. Śl., s. Automatyka, nr 64, Gliwice 1982, ss. 137-153.

[302] Torońska-Łuczyńska M.: Awtomatizacja uprawlienija proizwodstwom w kuznieczno m processie, Międzynarodowa Konferencja ARS'84, CSVTS, Ostrava, 1984.

[303] Torońska-Łuczyńska M.: Sistiema uprawlienija proizwodstwom w kuznieczno m processie. 5-th International Conference on: "Control System and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[304] Torońska-Łuczyńska M.: Zastosowanie programowania wieloetapowego do harmonogramowania produkcji w procesie kuźniczym. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 143-156.

[305] Torońska-Łuczyńska M.: Awtomatizacja uprawlienija proizwodstwom w kuznieczno m processie. ARS'84, Ostrawa 1984.

[306] Trhlik J.: Program vyvazowania montažnych liniek kombinačnom metodou. MONTEX-76, Bratysława 1976, pp. 41-45.

[307] Trybuś E.: Metody porządkowania. Matematyka stosowana, seria III, PWN, Warszawa 1975, ss. 95-117.

[308] Turzański J.: Analiza algorytmów harmonogramowania dyskretnych procesów przemysłowych w systemie o strukturze szeregowej. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1983.

[309] Van Tetterode L.M.: Work Measurement Line Balance and Simulation. The MTM Journal, 1971, V. 16, No 5.

[310] Van Beek H.G.: The Influence of Assembly Line Organization on Output, Quality and Morale. Occupational Psychology, 1964, V. 38, pp. 161-172.

[311] Wala K.: Metody symulacyjne sterowania dyskretnymi procesami produkcyjnymi,

[312] Wagner H.M.: Badania operacyjne. PWE, Warszawa 1980.

[313] Wadekind H.: Ein Linearer Programmansatz für das Fliessbandproblem. Ablauf und Planungsforschung, 1983, No 4.

[314] Wester L., Kilbridge M.D.: The Assembly Line Model-Mix Sequencing Problem. The Proceedings of the 3-rd International Conference on Operations Research, Paris 1964.

[315] Wester L., Kilbridge M.L.: Heuristic Line Balancing-A Case. The Journal of Industrial Engineering, 1962, V. 13, No 3.

[316] Wester L., Kolbridge M.D.: A Review of Analitical Systems of Line Balancing. Operations Research, 1962, V. 10, No 5.

[317] White W.W.: Comments on a Paper by Bowman. Operations Research, 1961, V. 9, No 2, pp. 274-276.

[318] Węglarz J.: Złożoność obliczeniowa problemów szeregowania zadań w celu minimalizacji maksymalnego opóźnienia. ZN Pol. Śl., s. Automatyka, nr 46, Gliwice 1978, ss. 160-167.

[319] Węglarz J.: Złożoność obliczeniowa problemów szeregowania zadań na równoległych maszynach z uwzględnieniem dodatkowych zasobów. ZN Pol. Śl., s. Automatyka, nr 56, Gliwice 1980, ss. 83-90.

[320] Węgrzyn S.: Podstawy automatyki. PWN, Warszawa 1972.

[321] Węgrzyn S.: Systemy automatyki kompleksowej - Teoria i doświadczenie. Podstawy sterowania, 1975, z. 1.

[322] Węgrzyn S.: Podstawy informatyki. PWN, Warszawa 1982.

[323] Wikierska I.: Uprawlienije processom okraski kuzow. 5-th International Conference on: "Control System and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[324] Wirtz N.: Algorytmy + struktury danych = programy. PWN, Warszawa 1981.

[325] Wołany D.: Harmonogramowanie procesu wykrawania blach karoseryjnych. Praca dyplomowa magisterska (niepublikowana), Instytut Automatyki, Pol. Śl., Gliwice 1982.

[326] Woszczyk J.: Harmonogramowanie procesu montażu karoserii samochodowych. Praca dyplomowa magisterska (niepublikowana). Instytut Automatyki Pol. Śl., Gliwice 1982.

[327] Zapałowicz W.: Sterowanie procesami sekwencyjnymi na przykładzie walcowni. ZN Pol. Śl., s. Automatyka, nr 54, Gliwice 1980, ss. 183-188.

[328] Zielińska-Król E. Marecki F.: Harmonogramowanie procesu walcowania na WCK. ZN Pol. Śl., s. Automatyka, nr 44, Gliwice 1978, ss. 123-134.

[329] Zielińska-Król E.: Problemy iskliuczenija sostojanij pri kaliendarnom płanirowanii processa regeneracji wałkow. 5-th Internationale Conference on: "Control and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

[330] Zielińska-Król E.: Zastosowanie metody programowania wieloetapowego do harmonogramowania procesu walcowania ciągłego. ZN Pol. Śl., s. Automatyka, nr 75, Gliwice 1984, ss. 157-168.

[331] Zielińska-Król E.: Problemy uprawlienija processom prokata. ARS'84, Ostrava 1984.

MODELE I ALGORYTMY ALOKACJI OPERACJI I ZASOBÓW
NA LINII MONTAŻOWEJ

S t r e s z c z e n i e

Rozprawa składa się z sześciu podstawowych rozdziałów oraz dodatku. W części zasadniczej przedstawiono deterministyczne modele matematyczne i algorytmy alokacji operacji i zasobów, w jednowersyjnym procesie montażu na liniach o szeregowych oraz równoległych stanowiskach pracy. Dodatek zawiera deterministyczne i probabilistyczne modele matematyczne i algorytmy sterowania jednowersyjnym oraz wielowersyjnym montażem dynamicznym.

Rozdział pierwszy stanowi wprowadzenie do rozważanej problematyki. Na tle ogólnych zagadnień związanych z dyskretnymi procesami przemysłowymi przedstawiono opis linii i procesu montażu oraz scharakteryzowano podstawowe problemy. Określono klasyczny problem balansowania linii montażowej (BLM) i dokonano przeglądu dokładnych oraz przybliżonych algorytmów jego rozwiązania. Ponadto wprowadzono koncepcję metody programowania wieloetapowego.

W rozdziale drugim rozwiązano klasyczny problem BLM za pomocą różnych algorytmów (binarny, szeregowania, przydziału oraz harmonogramowania), opartych na metodzie programowania wieloetapowego. Algorytmy te zilustrowano przykładami obliczeniowymi. Na podstawie testów komputerowych oraz ogólnych sformułowań przeanalizowano efektywność algorytmów programowania wieloetapowego. Do dalszej analizy wybrano algorytm harmonogramowania.

W rozdziale trzecim sformułowano uogólnione modele BLM. Wyróżniono modele z: dodatkowymi parametrami operacji, enklawami operacji (podzielnymi oraz niepodzielnymi), frontami pracy oraz kosztem montażu. Dla każdego modelu przedstawiono algorytm oparty na metodzie programowania wieloetapowego. W algorytmach tych szczególną uwagę zwrócono na eliminowanie stanów nieperspektywicznych.

Rozdział czwarty dotyczy alokacji operacji na liniach z szeregowymi stanowiskami pracy. Sformułowano modele z ograniczeniami: pozycyjnymi, czasowymi oraz zakresowymi. Ponadto rozważano linię asymetryczną. Dla każdego modelu przedstawiono algorytm programowania wieloetapowego. W rozdziale tym wprowadzono rozróżnienie stacji montażowej i stanowiska pracy.

W rozdziale piątym rozważano alokację operacji na linii z równoległymi stanowiskami pracy (dwa stanowiska pracy na jednej stacji montażowej). Sformułowano model podstawowy i modele z relacją wykluczania oraz synchronizacji. Dla każdego modelu przedstawiono algorytm programowania wieloetapowego. W modelu podstawowym szczegółowo przedyskutowano eliminację stanów nieperspektywicznych.

W rozdziale szóstym przedstawione są problemy alokacji operacji i zasobów na linii z szeregowymi stanowiskami pracy. Wyróżniono modele alokacji: operacji i kontenerów, operacji i narzędzi oraz operacji i monterów. Dla każdego modelu przedstawiono algorytm programowania wieloetapowego. W ogólnym przypadku dodatkowe zasoby w procesie montażu są substytucyjne i podwójnie ograniczone.

W dodatku proces montażu jest rozważany z dokładnością do operacji (detalu) na każdym stanowisku pracy. Wprowadzono definicje stanu procesu montażu oraz sterowania montażem. Na tej podstawie wyróżniono montaż (oraz sterowanie) statyczny, dynamiczny oraz przejściowy. Wyprowadzono równania stanu procesu montażu i zapasów detali (deterministyczne oraz probabilistyczne). W oparciu o te równania stanu można symulować proces montażu przy różnych (heurystycznych) regułach sterowania. Dla modelu jednowersyjnego przeanalizowano sterowanie: statyczne, pseudostatyczne, pseudodynamiczne oraz dynamiczne. Dla montażu wielowersyjnego przedstawiono algorytmy sterowania dynamicznego.

Algorytmy alokacji operacji i zasobów pozwalają wyznaczyć rozwiązanie optymalne lub przybliżone (jeżeli wyczerpany zostaje czas obliczeń komputerowych). Algorytmy sterowania montażem dynamicznym są oparte na równaniach stanu i heurystycznych regułach decyzyjnych, dlatego nie gwarantują rozwiązań optymalnych.

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ АЛЛОКАЦИИ ОПЕРАЦИЙ И РЕСУРСОВ В СБОРОЧНОЙ ЛИНИИ

Р е з ю м е

Работа состоит из шести основных глав и дополнения. В основной части работы представлены детерминистические и вероятностные математические модели а также алгоритмы управления одно и многоверсионной динамической сборкой.

Первая глава является вводной в рассматриваемую проблематику. На фоне общих вопросов, касающихся дискретных промышленных процессов, дано описание линии и процесса сборки а также охарактеризованы основные проблемы. Определена классическая проблема балансирования сборочной линией (БСЛ) а также дан обзор точных и приближенных алгоритмов её решения. Кроме того введена концепция метода многоэтапного программирования.

Во второй главе решена классическая проблема БСЛ при помощи различных алгоритмов (бинарного, упорядочения, приписания а также календарного планирования), основанных на методе многоэтапного программирования. Алгоритмы эти проиллюстрированы расчётными примерами. На основе компьютерных тестов а также общих определений дан анализ эффективности алгоритмов многоэтапного программирования. Для дальнейшего анализа выбран алгоритм календарного планирования.

В третьей главе сформулированы общие модели БСЛ. В частности показаны следующие модели: с добавочными параметрами по операциям, с энклавами операций (делимыми и неделимыми), с фронтами работ и стоимостью сборки. Для каждой модели представлен алгоритм на основе метода многоэтапного программирования. В этих алгоритмах особое внимание обращено на возможность исключения неперспективных состояний.

Четвёртая глава касается аллокации операций на линиях с последовательными местами работы. Сформулированы модели с ограничениями: позиционными, временными, а также диапазонными. Кроме этого рассмотрена ассиметрическая линия. Для каждой модели дан алгоритм многоэтапного программирования. В этой главе введены отличия сборочной станции и мест работы.

В пятой главе рассмотрена аллокация операций на линии с параллельными рабочими местами (два рабочие места на одной сборочной станции). Сформулированы основная модель и модели с отношениями исключения и синхронизации. Для каждой модели представлен алгоритм многоэтапного программирования. В основной модели подробно оговорено исключение неперспективных состояний.

В шестой главе представлены проблемы аллокации операций и ресурсов на линии с последовательными рабочими местами. В частности показаны следующие

модели аллокации: операций и контейнеров, операций и рабочего инструмента а также операций и монтёров. Для каждой модели дан алгоритм многоэтапного программирования. В общим случае добавочные ресурсы в процессе сборки являются взаимозаменяемыми и двухкратно ограниченными.

MATHEMATICAL MODELS AND ALGORITHMS FOR RESOURCE AND OPERATION ALLOCATION ON AN ASSEMBLY LINE

Summary

INTRODUCTION

A subject of consideration in the idssertation is an assembly process realized on a line, shown on the figure 1.1. The assembly line consists of: a main conveyor, assembly stations and fields, at which tools and assembly elements are placed. The main conveyor transports an assembly object along the line (with uniform or step motion). Chassis of the assembled objects are installed on the main conveyor in equal distances. The assembly stations are distinguished fields, which are placed sequentially along the line. On the assembly stations work places are localized. The work place is a place on the assembly station, at which only one operator can occur (an assembler or industrial robot). An assembly elements is a part of the assembly object. The tools and containers with assembly elements are placed on the fields beside the main conveyor. The assembly objects remain on each assembly station during time called the cycle-time. During the cycle-time on each station only one object is placed.

In the dissertation the assembly process is analysed at the period of time $[0,C]$, where C is a multiple of the cycle-time c. An initial condition of the assembly process is defined by degree of assembly objects completeness on the line at the moment t=0. A final condition of the assembly process is defined respectively for the moment t=C. At the assembly process on the line, boundary conditions are also distinguished (see fig. 1.1). The conditions are defined by the type of assembly objects which enter the first station or respectively leave the last station of the line.

If at each assembly station no more than one work place is arranged, assembly process is sequential and the line consists of sequential work places. Otherwise the line with parallel work places is considered. In the dissertation on more than two work places at each assembly station are assumed (placed at different sides of the main conveyor). If operation parameters (for example performance time) are equal for the different sides of the main conveyor, then assembly line is symmetrical; otherwise the assembly line is asymmetrical. The assembly process is considered as spatial (three-dimensional) if transport times between the points at which assembly elements are placed can not be omitted; otherwise we call it point assembly process. Assembly stations are basic resources of the assembly process. Additional resources are: operators, tools and containers with the assembly elements. In the dissertation it is displayed that the additional resources are double-constrained and substitutional.

In assembly process versions of objects are distinguished. The version is defined by the assembly elements of which the assembly object consists. At single-version of assembly process each object consists of the same assembly elements. At multiversion assembly objects may consist of various assembly elements. An assembly is called sequential if all the objects of the same version form the sequence without objects of other version; otherwise mixed assembly is considered. In the dissertation a detailed description of the sequential single-version assembly is given.

An assembly process is called static if at each cycle in the period $[0,C]$ the same operations are performed on the work places, otherwise the assembly is dynamic. A transient assembly process is a particular case of the dynamic one. The transient assembly process is dynamic during initial K cycles only then the process is the static one, (K is the number of stations). The transient assembly process occurs during a change of the object version which are assembled on the line. A detailed description of the dynamic assembly process is given in the appendix.

The dissertation displays that for single-version of assembly essential meaning have the following problems:

i) Assembly line balancing (ALB)
ii) Allocation of operations (AO), for the lines with sequential and parallel work places
iii) Resource and operation allocation (ROA)

The problems are considered in the chapters 2, 3, 4, 5 and 6 of the dissertation.

MULTI-STAGE PROGRAMMING METHOD

To solve the problems formulated in the dissertation, multi-stage programming method is proposed. The method is based on the ideas of the branch and bound and the dynamic programming methods. Fundamental construction elements of multi-stage programming algorithms are: a state of decision process, a value of the state, state generation procedures and rules for elimination unperspective states. An idea of the algorithms depends on generation of feasible trajectories (sequence of states) and an elimination of unperspective trajectories, which does not lead to the optimal solution.

In the dissertation the N-stage decision processes are considered, (N is the number of assembly operations). The numbers of the decision stages are denoted by η, however the numbers of states at the η-th stage

are denoted by l, (L_η means a feasible number of states at the η-th stage).

The state P of decision process interprets a situation at the system after an appropriate decision. In the dissertation the state is a vector for the ALB and AO problems, however for the ROA problem the state is a matrix with two columns. At the multi-stage programming algoritms the following states are distinguished: $P^{0,1}$ - initial state (specifies initial conditions of the process on the line), $P^{N,1}$ - final state (gives a feasible solution of the problem), $P^{\eta,1}$ - active state (enables to generate next states), P^a - actually the best final state, $P^{N,1^0}$ - local optimal state (the best final state obtained from the state $P^{\eta,1}$), P^0 - global optimal state (the best final state).

Each state $P^{\eta,1}$ is associated with its value $V^{\eta,1}$. The value $V^{\eta,1}$ is an estimation of the state $P^{\eta,1}$ from a given criterion point of view. The final state with the extremal value gives the optimal solution. In general case the value $V^{\eta,1}$ is a vector. Elements of the vector are calculated with aid of the apropriate state value function or recursively - during generation of the state $P^{\eta,1}$ from the state $P^{\eta-1,\lambda}$.

State generation consists in a selection of a certain active state $P^{\eta-1,\lambda}$ and a decision to perform the n-th operation, what gives the next state $P^{\eta,1}$. The active states are placed in lexicographically ordered sets $\mathcal{L}_\eta$. The number of elements in the set $\mathcal{L}_\eta$ can be limited to L_η, where $L_\eta \geq 1$. For selection of the active state various rules (for example FIFO, LIFO, LLB and so on) are used. The branch rule is called complete, if starting from the state $P^{\eta-1,\lambda}$ all its immediate successors $P^{\eta,1}$ are generated; otherwise the branching is partial. For the complete branching the state $P^{\eta-1,\lambda}$ is selected only once because after state generation it is no active; contrary for the partial branching the state $P^{\eta-1,\lambda}$ is selected many times. In this case with each state $P^{\eta,1}$ the index $I^{\eta,1}$ is associated. The index $I^{\eta,1}$ indicates which states have already been generated from the state $P^{\eta,1}$. In this way each state is generated only once and futhermore there are no omitted states. The application of the state indices gives possibility to limit the number of states in the sets $\mathcal{L}_\eta$.

A state generation procedure is a logical sentence defining conditions, which must be satisfied to generate new states. The state generation starts from the initial state $P^{0,1}$ and is continued until the sets $\mathcal{L}_\eta$ (for $\eta = 0,1,\ldots,N-1$) are not empty. The calculations are finished if each set $\mathcal{L}_\eta$ is empty.

During calculations unperspective states are eliminated. For this purpose the elimination rules called: exhausting, fathoming and domination are applied. The exhausting rule eliminates the state from which there is no possibilities to get feasible solution of the problem. The fathoming rule eliminates the state $P^{\eta,1}$ if appropriate local optimal state $P^{N,1^0}$

is worse than the given state P^a. The domination rule permits to eliminate one of the two states $P^{\eta,1^1}$ or $P^{\eta,1^2}$, which has the worse appropriate local optimal state.

In the dissertation all the formulated problems are solved with aid of the multi-stage programming algorithms. For this purpose the state, the state value, the state generation procedures and the unperspective state elimination rules are defined at each algorith. The proofs for the elimination rules are also given. Futhermore for illustration of the multi-stage programming algorithms in the dissertation there are examples and computational tests.

STANDARD ALB PROBLEM

In the second chapter a single version assembly process on a line with sequential work stations is considered. The standard ALB problem is formulated in the following way. There is a set Ω of the assembly operations. Performance times of the operations and a sequencing relation are given. For the given cycle-time c the minimum number of operation subsets, which form work stations, should be calculated. For the given cycle-time c the number M of work places can not be greater than the number K of assembly stations ($M \leq K$). Futhermore the following subsets of operations $\Omega_1,\ldots,\Omega_m,\ldots,\Omega_M$ are located to the succeeding assembly stations from the first one to the k-th one ($k \leq K$).

The standard ALB problem was solved with the aid of four multi-stage programming algorithms. The algorithms are called: binary, sequencing, allocation and scheduling one. In the algorithms a state is a vector, elements of which are defined in a different way. In the binary algorithm the vector element is equal to 1, if the decision about performing the appropriate operation has been taken, (0 - otherwise). In the sequencing algorithm the vector element is equal to s (where s is the position number of the operation in the sequence), if the decision about performing the appropriate operation has been taken, (0 - otherwise). In the allocation algorithm the element of the state vector is equal to m, if the appropriate operation is allocated to the subset Ω_m, (0 - otherwise). In the scheduling algorithm the element of the state vector is equal to t_n, (where t_n is the moment at which performing of the operation is finished), if the decision about performing the appropriate operation has been taken, (0 - otherwise). Calculation examples and computer tests of the algorithms also have been given. For considerations in the following chapters the scheduling algorithm has been selected.

GENERALIZED ALB MODELS

In the third chapter the generalized ALB models are considered. In the models the following additional restrictions have been taken into account:

i) exclusion of certain subsets of the operations at one work place
ii) inclusion of certain operations to subset called an enclave
iii) spacial localization of operations at assembly object

Futhermore at the fourth model more general criterion of ALB, namely assembly cost, is considered.

To solve the formulated problems, the scheduling algorithm of the multi-stage programming method is used. The state of the decision process is defined in the same way for each problem. For the problem of the assembly cost the state value is defined more general than for other problems. State generation procedures and the rules for elimination of the unperspective states are different at each problem. In the algorithms proofs of the elimination rules have been given.

ALLOCATION OF OPERATIONS. SEQUENTIAL WORK STATIONS

In the fourth chapter the models of the operation allocation on a line with sequential work places are given. In general case for the operation allocation problems, subsets Ω_m are not located to the succeeding stations k of the assembly line. In the chapter the following models are distinguished:

i) The line with positions for work places
ii) The assembly process with time restrictions of operations
iii) The line with station ranges for operations
iv) An asymmetrical line

In each model the operations are performed sequencialy and at each station no more than one work place can be arranged. To solve the formulated problems the scheduling algorithms of multi-stage programming have been used. In the problems of operation allocation on symmetrical lines the state of the decision process is defined in the same way in the standard ALB problem. For the asymmetrical line the state of the decision process is defined differently. If the n-th operation is performed on the left hand side of the main conveyor, then the appropriate element of the state vector is negative ($-t_n$); (otherwise the element is equal + t_n). In the problems of the operations allocation the state value is defined more generally than in the ALB ones. In this chapter state generation procedures and rules for elimination of unperspective states are given for each problem. The rules have been proved.

ALLOCATION OF OPERATIONS. PARALLEL WORK STATIONS

Solution of the ALB and AO problems can be not possible on the line with sequential work stations, if the cycle-time is respectively short. In this case the problem of operation allocation on the line with parallel work stations is considered. In the fifth chapter the problem is analysed with the assumption that no more than two work places on each line

station are present. The parallel work stations are placed at both sides of the main conveyor. On the lines with parallel work stations the operations can be performed parallely.
In the chapter the following models are distinguished:

i) The basic model (with the sequencing relation)
ii) The model with the sequencing and exclusion relation
iii) The model with the sequencing and synchronization relation

The formulated problems are solved with aid of the scheduling algorithms of the multi-stage programming method. The decision process state is defined in the same way as for AO problem on the asymmetrical line with sequential work places. The state value determines the sum of work places at the both sides of the main conveyor. For each problem particular state generation procedures are introduced. Futhermore unperspective states elimination rules have been given.

RESOURCE AND OPERATION ALLOCATION

In the problems of ALB and AO only work places as a basic resource at the assembly process were considered. In the sixth chapter some additional resources such as: containers with assembly elements, assembly tools and operators (assemblers or industrial robots) are taken into consideration. In the chapter the following models are distinguish:

i) Operation and containers allocation
ii) Operation and tools allocation
iii) Operation and assemblers allocation

In the models the assembly line with the sequential work places is assumed. In general case the discussed additional resources are substitutional and double-constrained.

The formulated problems are solved with aid of scheduling algorithms of the multi-stage programming method. The decision process state is defined as a matrix with two columnes. The first column is analogious to the decision process state in the problems of ALB and AO. In the second column an information about the resource used to performing the operation has been written. The state value is a vector. The first element of the vector gives the number of work places, however other elements described numbers of used additional resources.

For each problem particulare state generation procedures and unperspective states elimination rules are presented. Futhermore multi-objective optimization has been discussed.

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej
P 3341/16/82

WYDAWNICTWA NAUKOWE I DYDAKTYCZNE POLITECHNIKI ŚLĄSKIEJ MOŻNA NABYĆ W NASTĘPUJĄCYCH PLACÓWKACH:

44-100 Gliwice — Księgarnia nr 096, ul. Konstytucji 14 b
44-100 Gliwice — Spółdzielnia Studencka, ul. Wrocławska 4 a
40-950 Katowice — Księgarnia nr 015, ul. Żwirki i Wigury 33
40-096 Katowice — Księgarnia nr 005, ul. 3 Maja 12
41-900 Bytom — Księgarnia nr 048, Pl. Kościuszki 10
41-500 Chorzów — Księgarnia nr 063, ul. Wolności 22
41-300 Dąbrowa Górnicza — Księgarnia nr 081, ul. ZBoWiD-u 2
47-400 Racibórz — Księgarnia nr 148, ul. Odrzańska 1
44-200 Rybnik — Księgarnia nr 162, Rynek 1
41-200 Sosnowiec — Księgarnia nr 181, ul. Zwycięstwa 7
41-800 Zabrze — Księgarnia nr 230, ul. Wolności 288
00-901 Warszawa — Ośrodek Rozpowszechniania Wydawnictw Naukowych PAN — Pałac Kultury i Nauki

Wszystkie wydawnictwa naukowe i dydaktyczne zamawiać można poprzez Składnicę Księgarską w Warszawie, ul. Mazowiecka 9.