Time-Dependent Path Scheduling

Helmut A. Sedding

Time-Dependent Path Scheduling

Algorithmic Minimization of Walking Time at the Moving Assembly Line

Springer Vieweg

Helmut A. Sedding
Institute of Theoretical Computer Science
Ulm University
Ulm, Germany

Dissertation, Ulm University, Germany, 2019

ISBN 978-3-658-28414-5 ISBN 978-3-658-28415-2 (eBook)
https://doi.org/10.1007/978-3-658-28415-2

Springer Vieweg
© Springer Fachmedien Wiesbaden GmbH, part of Springer Nature 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Acknowledgements

This monograph is my dissertation thesis in partial fulfillment of the doctoral degree of natural sciences at Ulm University, Germany. It was only possible to get to this point with the support of several other persons. I feel indebted to Prof. Dr. Uwe Schöning for generously offering the supervision of my dissertation and appointing me a doctoral position at the Institute of Theoretical Computer Science at Ulm University. I am highly grateful to Prof. Dr. Nils Boysen for his support and his review of my thesis. As well, I would like to warmly recognize Prof. Dr. Enno Ohlebusch for his effort to referee my thesis. For their committee membership, let me thank Prof. Dr. Birte Glimm, Prof. Dr.-Ing. Franz Hauck, and PD Dr. Friedhelm Schwenker. In addition, I especially thank Prof. Dr. Stanisław Gawiejnowicz for his invitation to Poznań and the valuable discussions about time-dependent scheduling. Let me further take the opportunity to thank the Initiative Wissenschaft und Automobilindustrie in Jena for awarding me their research price of the year 2016. Last but not least, I am highly pleased to thank my family, friends, and colleagues for their enduring support in devising this thesis.

Helmut A. Sedding

Abstract

Moving assembly lines are the stepping stone for mass production of automobiles. Here, every second counts. Therefore, nonproductive walking time of workers is meticulously optimized by production planners. This is difficult at moving assembly lines because walking distances between workpiece and line-side are immanently time-dependent. Therefore, there exist only few computational approaches in optimizing them. In this work, we introduce a core model of the problem settings, analyze combinatorial properties, computational complexity, and provide means for algorithmically optimizing walking time. We optimize the sequence of all operations, and the position of the corresponding line-side material containers at each work station. The performance of our algorithms is evaluated in numerical experiments followed by statistical analyses. This shows that our results provide the base for decision support systems, thus enable to computationally plan moving assembly lines while minimizing time-dependent walking time.

Zusammenfassung

Fließfertigungslinien bilden die Grundlage der Fahrzeug-Massenproduktion. Dabei zählt jede Sekunde. Nichtproduktive Wegezeiten der Werker werden von Produktionsplanern daher sorgfältig optimiert. Dies ist bei bewegten Montagelinien jedoch anspruchsvoll, da die Laufdistanzen zwischen Werkstück und Linienseite immanent zeitabhängig sind. Daher existieren hierfür kaum rechnergestützte Optimierungsverfahren. Diese Arbeit stellt ein Kernmodell für die Problemstellungen vor, analysiert kombinatorische Eigenschaften sowie die Komplexität und bietet Mittel für die algorithmische Optimierung von Wegezeiten. Optimiert wird einerseits die Reihenfolge aller Arbeitsvorgänge, andererseits die Position von Materialbehältern an jeder Arbeitsstation. Eine Evaluation der Leistungsfähigkeit entwickelter Algorithmen findet mit numerischen Experimenten gefolgt von statistischen Analysen statt. Diese zeigt, dass die Arbeitsergebnisse eine Grundlage von Entscheidungsunterstützungssystemen liefern und damit befähigen, Fließfertigungslinien unter Minimierung von zeitabhängigen Wegezeiten rechnergestützt zu planen.

Contents

List of Definitions XV
List of Algorithms XVII
List of Statements XIX
List of Figures XXI
List of Tables XXIII

I Introduction 1

1 Introduction 3
1.1 Introduction 3
1.2 Organization 5

2 Modeling 7
2.1 Modeling 7
2.2 Related literature 8
2.3 Assumptions 12
2.3.1 Variable assumptions 12
2.3.2 Common assumptions 13
2.4 Walking strategies 16

II Operation sequencing 21

3 Operation sequencing 23
3.1 Introduction 23
3.2 Problem definition 24
3.3 Related literature 25
3.3.1 Assembly line balancing 25
3.3.2 Classic scheduling 27
3.3.3 Time-dependent scheduling 30

3.4 Polynomial cases 33
3.5 Lower bound 36
3.6 Dominance rule 39
3.7 Solution algorithms 40
3.7.1 Dynamic programming 40
3.7.2 Mixed integer program 41
3.7.3 Basic heuristics 42
3.7.4 Branch and bound algorithm 43
3.8 Numerical results 44
3.8.1 Instance generation 45
3.8.2 Exact algorithms 46
3.8.3 Heuristics 49
3.9 Conclusion 50

4 Operation sequencing with a single box position 51
4.1 Introduction 51
4.2 Related literature 53
4.3 Computational complexity 54
4.4 Dynamic programming algorithm 59
4.5 Fully polynomial time approximation scheme 62
4.6 Polynomial algorithm for a variable global start time . . . 67

III Box placement 71

5 Box placement for one product variant 73
5.1 Introduction 73
5.2 Related literature 74
5.3 Problem definition 75
5.4 Polynomial cases 77
5.5 Computational complexity 82
5.6 Lower bound 87
5.7 Dominance rule 90
5.7.1 Exact dominance rule 90
5.7.2 Heuristic dominance rule 97
5.8 Solution algorithms 98

5.8.1 Mixed integer program 98
5.8.2 Basic heuristics 99
5.8.3 Branch and bound algorithm 100
5.8.4 Truncated branch and bound heuristic 101
5.9 Numerical results 101
5.9.1 Instance generation 102
5.9.2 Test setup 103
5.9.3 Exact algorithms 104
5.9.4 Heuristics 107
5.10 Conclusion 109

6 Box placement for multiple product variants 111
6.1 Introduction 111
6.2 Problem definition 113
6.3 Computational complexity 114
6.4 Mixed integer programs 115
6.4.1 Disjunctive sequencing 115
6.4.2 Space-indexing 116
6.5 Lower bound 118
6.5.1 Lagrangian relaxation of processing times 119
6.5.2 Subgradient search 120
6.5.3 Solving the Lagrangian relaxation 120
6.5.4 Determining box positions 122
6.5.5 Determining walk times 123
6.5.6 Greedily determining walk times 124
6.6 Solution algorithms 128
6.6.1 Basic heuristics 128
6.6.2 Branch and bound algorithm 129
6.6.3 Truncated branch and bound algorithm 130
6.7 Numerical results 130
6.7.1 Instance generation 131
6.7.2 Test setup 132
6.7.3 Exact algorithms 132
6.7.4 Heuristics 139

6.8 Conclusion 141

IV Conclusion 143

7 Conclusion 145

7.1 Conclusion 145

7.2 Future steps 146

8 Summary of major contributions 149

8.1 Sequencing assembly operations 149

8.2 Line side placement 150

Bibliography 153

Publications 169

List of Definitions

Definition 2.1 Walk time $\varpi_j(t)$ 16
Definition 3.1 Problem $\mathcal{S}$ 24
Definition 4.1 Problem $\hat{\mathcal{S}}$ 51
Definition 4.2 Problem $\hat{\mathcal{C}}$ 52
Definition 4.3 Even Odd Partition (Garey et al., 1988) 54
Definition 5.1 Problem $\mathcal{P}$ 75
Definition 5.2 Three Partition (3P) (Garey and Johnson, 1979) . 82
Definition 6.1 Problem $\mathcal{P}m$ 113

List of Algorithms

Algorithm 1 Dynamic programming algorithm for $\hat{\mathcal{S}}$ 60
Algorithm 2 FPTAS for $\hat{\mathcal{S}}$ 63
Algorithm 3 Combinatorial lower bound for $\mathcal{P}$ 89
Algorithm 4 Dominance rule for $\mathcal{P}$ 96
Algorithm 5 Weighted nearest identity sequence heuristic f. $\mathcal{P}m$ 129

List of Statements

Lemma 3.1 33
Lemma 3.2 35
Property 3.3 39
Theorem 4.1 54
Theorem 4.2 60
Remark 4.3 61
Property 4.4 62
Property 4.5 64
Property 4.6 65
Theorem 4.7 67
Corollary 4.8 67
Theorem 4.9 67
Remark 4.10 68
Remark 4.11 69
Lemma 5.1 77
Lemma 5.2 82
Lemma 5.3 84
Lemma 5.4 85
Theorem 5.5 87
Property 5.6 93
Property 5.7 94
Property 5.8 94
Theorem 6.1 114
Corollary 6.2 115
Remark 6.3 117
Lemma 6.4 122

Remark 6.5 123
Lemma 6.6 124
Property 6.7 125
Corollary 6.8 126
Lemma 6.9 126
Property 6.10 127
Remark 6.11 131

List of Figures

Figure 1 An example assembly station 9
Figure 2 Comparison of walking strategies 17
Figure 3 Example instance for $\mathcal{S}$ 25
Figure 4 Time-dependent scheduling models with piecewise-linear processing times 32
Figure 5 An example for a polynomial case of $\mathcal{S}$ 37
Figure 6 Another example for a polynomial case of $\mathcal{S}$ . . . 37
Figure 7 Lower bound on a $\mathcal{S}$ instance 39
Figure 8 $\mathcal{S}$: MIP, B&B runtime in box and scatter plots . . 47
Figure 9 Example $\hat{\mathcal{S}}$ instance 52
Figure 10 Example instance for $\mathcal{P}$ 76
Figure 11 Example instance for $\mathcal{P}$ with uniform assembly times and uniform box widths, respectively 77
Figure 12 Example of a polynomial case of $\mathcal{P}$ 77
Figure 13 Corresponding $\mathcal{P}$ instance in the pseudopolynomial reduction of an example 3P instance 84
Figure 14 Lower bound on a partial $\mathcal{P}$ schedule 90
Figure 15 Dominance rule on a partial $\mathcal{P}$ schedule 91
Figure 16 $\mathcal{P}$: MIP, B&B runtime in box and scatter plots . . 105
Figure 17 Example instance for $\mathcal{P}m$ 112
Figure 18 $\mathcal{P}m$: MIP, B&B runtime in box and scatter plots for each n 133
Figure 19 $\mathcal{P}m$: MIP, B&B runtime in box and scatter plots for each m 134
Figure 20 $\mathcal{P}m$: MIP, B&B runtime in a product plot for each n and m pair 135

List of Tables

Table 1 Slope values by walking velocity and walking strategy 20
Table 2 Complexity results on related classic objectives for single machine scheduling 29
Table 3 $\mathcal{S}$: MIP, B&B runtime and performance 47
Table 4 $\mathcal{S}$: Mean walking time percentage by assembly time and box width setting 48
Table 5 $\mathcal{S}$: Mean walking time percentage by assembly time and box width setting 48
Table 6 $\mathcal{S}$: Heuristics' runtime and performance 49
Table 7 $\mathcal{P}$: MIP, B&B runtime and performance 104
Table 8 $\mathcal{P}$: B&B runtime by walking velocity and walking strategy 107
Table 9 $\mathcal{P}$: B&B runtime by assembly time and box width setting 107
Table 10 $\mathcal{P}$: Heuristics' runtime and performance 108
Table 11 $\mathcal{P}m$: MIP, B&B runtime and performance 136
Table 12 $\mathcal{P}m$: MIP, B&B runtime and performance for each n and m pair 137
Table 13 $\mathcal{P}m$: B&B runtime by walking velocity and walking strategy 138
Table 14 $\mathcal{P}m$: B&B runtime by assembly time and box width setting 138
Table 15 $\mathcal{P}m$: Heuristics' runtime and performance 139
Table 16 $\mathcal{P}m$: Heuristics' runtime and performance for each n and m pair 140