

New Hybrid Evolutionary Algorithms for Chemical Batch Scheduling under Uncertainty

Zur Erlangung des akademischen Grads eines

Dr.-Ing.

vom Fachbereich Bio- und Chemieingenieurwesen
der Universität Dortmund
genehmigte Dissertation

vorgelegt von

Dipl.-Ing. Jochen Till

aus

Gelnhausen

Tag der mündlichen Prüfung: 24. August 2007

1. Gutachter: Prof. Dr.-Ing. Sebastian Engell, Dortmund
2. Gutachter: Prof. Dr. rer. nat. Petra Mutzel, Dortmund

Dortmund 2007

Schriftenreihe des Lehrstuhls für Anlagensteuerungstechnik
der Universität Dortmund (Prof.-Dr. Sebastian Engell)

Band 5/2007

Jochen Till

**New Hybrid Evolutionary Algorithms for
Chemical Batch Scheduling under Uncertainty**

D 290 (Diss. Universität Dortmund)

Shaker Verlag
Aachen 2007

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: Dortmund, Univ., Diss., 2007

Copyright Shaker Verlag 2007

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8322-6839-8

ISSN 0948-7018

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: www.shaker.de • e-mail: info@shaker.de

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Angestellter am Lehrstuhl für Anlagensteuerungstechnik des Fachbereichs Bio- und Chemieingenieurwesen der Universität Dortmund. Dem Lehrstuhlinhaber, Herrn Prof. Dr.-Ing. Sebastian Engell, danke ich sehr herzlich für die Ermöglichung und Betreuung dieser Arbeit. Ihm sowie seinen ehemaligen und aktuellen Mitarbeitern danke ich für die Schaffung einer kreativen und leistungsfördernden Arbeitsatmosphäre am Lehrstuhl.

Für das mit Interesse übernommene Korreferat gilt mein Dank Frau Prof. Dr. rer. nat. Petra Mutzel, der Inhaberin des Lehrstuhls für Algorithm Engineering am Fachbereich Informatik. Den Herren Prof. Dr. David Agar und Prof. Dr.-Ing. Gerhard Schembecker vom Fachbereich Bio- und Chemieingenieurwesen danke ich für ihre Mitwirkung in der Prüfungskommission.

Für die finanzielle Förderung dieser Arbeit als Projekt C10 des Sonderforschungsbereichs (SFB) 531 „Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence“ an der Universität Dortmund danke ich der Deutschen Forschungsgemeinschaft. Im Rahmen dieses SFB hat sich eine fachbereichsübergreifende, stetige und erfolgreiche Kooperation mit Herrn Prof. Dr. Michael Emmerich entwickelt, die ich hier würdigen möchte. Er ist heute Assistant Professor am Leiden Institute of Advanced Computer Science in den Niederlanden.

Die Zusammenarbeit mit der Arbeitsgruppe „Optimierung und Algorithmische Diskrete Mathematik“ des Fachbereichs Mathematik der Universität Duisburg-Essen ermöglichte mir den Einsatz des Softwarewerkzeugs *DDSIP*. Für ihren Beitrag dazu danke ich dem Leiter der Arbeitsgruppe, Herrn Prof. Dr. Rüdiger Schultz, sowie Frau Dipl.-Math. Eva Cloostermann.

Mein besonderer Dank gilt Frau Dipl.-Inform. Maren Urselmann für ihr stetiges Engagement bei ihrer Mitarbeit als studentische Hilfskraft, als Diplomandin, und später als geschätzte Kollegin. Mein herzlicher Dank gebührt Herrn Dr.-Ing. Guido Sand für seine unermüdliche Bereitschaft zu fruchtbaren fachlichen Diskussionen und für sein stetig kritisches Hinterfragen meiner Arbeiten. Ihm, Frau Sabine Piana, M. Sc. und Herrn Dipl.-Ing. Wolfgang Mautz danke ich für ihre hilfreichen Hinweise beim Korrekturlesen dieser Arbeit.

Darüber hinaus gilt mein Dank für ihren Beitrag auf dem Weg zu dieser Dissertation den Herren Dr.-Ing. Dipl.-Inform. Sebastian Panek und Prof. Dr.-Ing. Olaf Stursberg.

Auch den beiden „guten Geistern“ des Lehrstuhls, Frau Gisela Hensche und Frau Dorothea Weber, gebührt mein Dank für ihre vielfältige Hilfe.

Schließlich möchte ich noch ein sehr herzliches Dankeschön an meine Freundin Verena und an meine Familie aussprechen, die mich stets unterstützt haben.

Weinheim, im August 2007

Jochen Till

Zusammenfassung

Zweistufige stochastische gemischt-ganzzahlige Programme mit diskreten Szenarien (*2S-MILPs*), die bei der Belegungsplanung von Chargenprozessen unter Unsicherheit entstehen, führen zu großen Optimierungsproblemen. Diese großen Probleme können ohne Dekompositionsmethoden oder problemspezifische Ansätze nur schwer gelöst werden.

Diese Arbeit präsentiert neue hybride evolutionäre Algorithmen für *2S-MILPs*, die auf Stufen-Dekomposition basieren. Dabei wird die Suche auf den Erststufenentscheidungen von einem evolutionären Algorithmus (*EA*) durchgeführt, die entkoppelten Zweitstufen-Szenarioprobleme werden durch einen *MILP*-Solver gelöst. Die Algorithmen werden an einem realistischen Fallbeispiel, der Belegungsplanung eines Chargenprozesses zur industriellen Polymerproduktion unter Unsicherheit in der Nachfrage und in der Anlagenkapazität, evaluiert.

Zunächst wird ein generischer hybrider *EA* für *2S-MILPs* realisiert. Dieser wird dann um drei für das Fallbeispiel spezifische Elemente erweitert. Zuerst wird ein schnelles Initialisierungsschema entwickelt. Dann wird eine Methode zur Reduzierung des Suchraums um zwei Größenordnungen vorgeschlagen. Zuletzt wird, basierend auf dem Konzept der ‘minimal moves’, ein problemspezifischer *EA* entwickelt. Es zeigt sich, dass dieser spezifische *EA* robuster ist als der generische, und dass er bessere Fähigkeiten zur vollständigen Erkundung des Suchraums aufweist.

Die Leistungsmerkmale der hybriden evolutionären Algorithmen werden mit den beiden besten verfügbaren Algorithmen, dem *MILP*-Solver *CPLEX*, und dem dekompositionsbasierten *2S-MILP*-Algorithmus von Carøe und Schultz (1999), *DDSIP*, verglichen. *DDSIP* benötigt eine problemspezifische Anpassung, um überhaupt eine andere als die initiale Lösung zu liefern. Der generische *EA* hingegen konvergiert ohne Anpassung. Mit den problemspezifischen Anpassungen liefern *DDSIP* und die *EA*-basierten Ansätze in den meisten Fällen bessere Ergebnisse als *CPLEX*. Der Lösungsaufwand des *EA* skaliert etwa linear mit der Anzahl der Szenarien. *CPLEX* und *DDSIP* hingegen skalieren deutlich stärker als linear. Für *2S-MILPs* mit einer relativ großen Zahl von Szenarien sind hybride evolutionäre Algorithmen daher zu bevorzugen, ebenso wenn relativ gute Lösungen nach kurzer Rechenzeit benötigt werden.

Abstract

Two-stage stochastic mixed-integer linear programs with discrete scenarios (*2S-MILPs*) that arise in chemical batch scheduling under uncertainty usually give rise to large scale optimization problems. The large problems cannot be solved easily without incorporating decomposition methods or problem specific knowledge.

A new hybrid evolutionary algorithm framework is proposed. Based on stage-decomposition, an evolutionary algorithm (*EA*) performs the search on the first-stage variables while the decoupled second-stage scenario problems are tackled by a *MILP* solver. The approach is evaluated for a real-world scheduling problem of an industrial polymer production batch plant with uncertainties in the demand and in the plant capacity.

In the beginning a generic hybrid *EA* for *2S-MILPs* is realized. Then it is extended by three elements that are customized to the example problem. First, a computationally cheap specific initialization scheme is developed. Second, a reduction of the search space by two orders of magnitude is proposed. Third, a problem specific *EA* with mutation operators is developed based on the concept of ‘minimal moves’. The specific *EA* is more robust than the generic one and has better abilities to explore the complete search space.

The performance of the hybrid *EA*-based approaches is compared to the performance of the state-of-the-art *MILP* solver *CPLEX* and to that of the state-of-the-art decomposition based *2S-MILP*-algorithm of Carøe and Schultz (1999), *DDSIP*. The latter *2S-MILP*-algorithm required a problem specific adaptation to provide any other than the initial solution, whereas the generic *EA* converged without modifications. With problem specific adaptations, *DDSIP* and the *EA*-based approaches performed better than *CPLEX* in most cases. The scale-up behavior of the *EA* with respect to the number of scenarios is approximately linear, whereas that of *CPLEX* and *DDSIP* is significantly stronger than linear. For *2S-MILPs* with a relatively high number of scenarios, or when relatively good solutions are needed quickly, the *EA* based approach is preferable.

Contents

1	Introduction	1
1.1	Problem statement and motivation	1
1.2	Purpose	2
1.3	Outline	3
2	Chemical Batch Scheduling under Uncertainty	5
2.1	Deterministic chemical batch scheduling	6
2.1.1	Definition of a chemical batch scheduling problem	6
2.1.2	Formulation of a chemical batch scheduling problem	7
2.2	Uncertainties in batch scheduling	7
2.3	A taxonomy of uncertainty-conscious scheduling approaches	8
2.3.1	Reactive vs. stochastic scheduling	8
2.3.2	Stochastic vs. probabilistic programming	9
2.4	Applications of stochastic programming models	10
2.4.1	Stochastic mixed-integer linear programming with non-integer re- course	11
2.4.2	Stochastic mixed-integer linear programming with mixed-integer re- course	12
2.5	Other applications of stochastic mixed-integer linear programming	12
2.6	Summary, conclusions, and directions	13
3	Scheduling by Stochastic Programming	15
3.1	Motivating example	15
3.1.1	Model formulation of the scheduling problem	15
3.1.2	Demand uncertainties	17
3.2	Deterministic online scheduler	17
3.3	Stochastic online scheduler	19

3.4	Comparison	22
3.5	Conclusions	22
4	Stochastic Programming	25
4.1	Introduction to mathematical programming	25
4.1.1	Classes of mathematical programming formulations	26
4.1.2	Solution techniques	28
4.1.3	Computational complexity	29
4.2	Two-stage stochastic mixed-integer programming	30
4.2.1	General formulation and extensive representation	31
4.2.2	Extensive form with explicit nonanticipativity constraints	32
4.2.3	Implicit representation	33
4.3	Corresponding programs	34
4.3.1	The expected value problem	34
4.3.2	The wait-and-see problem	36
4.3.3	Additional nonanticipativity constraints	37
4.3.4	Integer relaxation	38
4.4	Techniques for solving a <i>2S-MILP</i>	39
4.4.1	Decomposition of a two-stage stochastic program	39
4.4.2	Scenario decomposition based branch-and-bound-algorithm (<i>DDSIP</i>)	40
4.4.3	Stage decomposition based (L-shaped) algorithms	45
4.5	A new stage decomposition based hybrid evolutionary approach	45
4.6	Summary, conclusions, and directions	46
5	Evolutionary Algorithms	47
5.1	Introduction and principle	47
5.2	Modules for unconstrained optimization	48
5.2.1	The generic <i>EA</i> framework	49
5.2.2	Recombination	51
5.2.3	Mutation	52
5.2.4	Mutation parameter adaptation	57
5.2.5	Selection for replacement	58
5.3	Constraint handling techniques	59
5.3.1	Problem statement and introduction	60

5.3.2	Penalty functions	61
5.3.3	Separation of objectives and constraints	62
5.3.4	Methods preserving feasibility of the solutions	64
5.3.5	Repair algorithms	65
5.4	Summary, conclusions, and directions	66
6	A Generic Hybrid EA Framework for 2S-MILPs	67
6.1	Hybrid evolutionary algorithm	67
6.1.1	Algorithmic framework	67
6.1.2	The modified objective function	69
6.1.3	Mixed-integer evolution strategy	70
6.2	Initialization	74
6.2.1	Random generation	74
6.2.2	Corresponding stochastic programs	74
6.3	Summary and directions	74
7	Real-world Scheduling Example: EPS Production	77
7.1	Introduction to the scheduling problem	78
7.2	Deterministic model formulation	79
7.2.1	Model of the production	80
7.2.2	Model of the objective	81
7.2.3	Objective function	83
7.3	Extension to a two-stage stochastic program	83
7.3.1	Scenarios of capacity and demand uncertainties	84
7.3.2	Definition of the stages and formulation of the master-problem	85
7.3.3	Completeness of the recourse	87
7.3.4	Model classification and dimensions	87
7.4	Summary	88
8	Engineered EA Components	89
8.1	EPS-specific feasible initialization operator	90
8.1.1	Introduction to constraint programming	90
8.1.2	Analysis of the given constraint satisfaction problem	92
8.1.3	Solving the constraint satisfaction problem	97
8.1.4	Uniform random feasible initialization	102

8.2	Search space reduction	105
8.3	<i>EA</i> with EPS-problem specific variation operators	107
8.3.1	Problem specific representation of solutions	108
8.3.2	Problem specific mutation operator	110
8.3.3	Realization of the algorithmic framework of <i>P-EA</i>	114
8.4	Summary, conclusions, and directions	115
9	Comparative Numerical Studies	117
9.1	Evaluation of algorithms	117
9.1.1	Desirable properties of an algorithm	117
9.1.2	Performance measures	118
9.2	Experimental setup	120
9.3	Initialization and preliminary investigations	120
9.3.1	Problem-independent analytical methods	121
9.3.2	EPS-specific initialization	122
9.3.3	Density of states	123
9.3.4	Accuracy of the fitness evaluation	123
9.3.5	Conclusions	124
9.4	Evaluation of the hybrid evolutionary algorithms	124
9.4.1	Evaluation of the generic hybrid <i>EA</i> and of EPS-specific adaptations	124
9.4.2	Evaluation of the tailored <i>EA</i> , <i>P-EA</i>	127
9.4.3	Conclusions	128
9.5	Hybrid <i>EAs</i> versus state-of-the-art exact algorithms	129
9.5.1	Generic <i>MILP</i> -algorithm: <i>CPLEX</i>	129
9.5.2	Generic <i>2S-MILP</i> -algorithm: <i>DDSIP</i>	130
9.5.3	Conclusions	131
9.6	Robustness over model variants	132
9.6.1	Robustness to varying problem parameters	132
9.6.2	Scale-up behavior	132
9.7	Conclusions	136
10	Conclusions and Outlook	137
10.1	Summary and Conclusions	137
10.2	Outlook	138

A Demand Data	141
B Probability Distributions	143
B.1 Introduction	143
B.1.1 Continuous distributions	143
B.1.2 Discrete distributions	144
B.2 Distribution functions	144
B.2.1 Continuous uniform distribution	144
B.2.2 Normal distribution	144
B.2.3 Standard normal distribution	145
B.2.4 Geometric distribution	145
B.2.5 Symmetric geometric distribution	146
C Initial Population Data	147
D Parametrization of <i>DDSIP</i>	149
D.1 Default specification file for <i>DDSIP</i>	149
D.2 Analysis of the parameters	152
D.2.1 Input variables	152
D.2.2 Output (responses)	153
D.2.3 Response data and conclusions	155
D.2.4 Conclusions	159
Nomenclature	161
Index	169

List of Figures

2.1	Uncertainty-conscious scheduling: taxonomy of technologies	9
3.1	Scheduling problem: the couplings of the production decisions x_i over the periods due to storage and due to late sale	16
3.2	Moving horizon scheme: only the current decisions $x_{j=i}(t_i)$ are applied to the plant	18
3.3	Uncertainties: scenario tree of the demand for three periods	18
3.4	Deterministic scheduler: sequence of decisions and results for all scenarios (average objective after three periods $\bar{P}_{obj} = -16.05$)	20
3.5	Uncertainties: the multi-stage scenario tree of period $i = 1$ and its two-stage approximation (both with a horizon of $H = 2$ periods)	21
3.6	Moving horizon scheme with two-stage stochastic models: the first-stage decisions $x_i(t_{j=i})$ are applied to the plant (compare to Figure 3.2)	22
3.7	Stochastic scheduler: sequence of decisions and results for all scenarios (average objective after three periods $\bar{P}_{obj} = -17.65$), compare to Figure 3.4	23
3.8	Deterministic vs. stochastic scheduler: comparison of the objective after three periods for the demand scenarios given in Figure 3.3	24
4.1	Constraints of the $2S$ -MILP (4.14)-(4.15): staircase structure	39
4.2	Decomposition: scenario decomposition and stage decomposition of a $2S$ -MILP	40
5.1	Evolutionary algorithm: general scheme as a flow chart	48
5.2	Recombination of two parents: possible offspring of dominant and intermediate operators	52
5.3	Mutation: probability density function of the (continuous) normal distribution, $\mathcal{N}(0, \sigma)$	55
5.4	Mutation: probability function of the (discrete) symmetric geometric distribution, $\mathcal{G}_{sym}(p_{geo})$	56
5.5	Mutation: probability functions of the (discrete) symmetric geometric distribution, $\mathcal{G}_{sym}(p_{geo}(s, n))$, for different mean step sizes s	56

5.6	Self-adaptation: the log-normal distribution, $\exp(\tau\mathcal{N}(0,1))$	58
6.1	Modified objective function: illustrated for a one-dimensional continuous object parameter with first- and second-stage infeasibilities	71
7.1	EPS plant: flow sheet	78
7.2	EPS model: yields $\bar{\rho}_{f_p, r_p}$ of the grain size fractions f_p according to recipe r_p (identical for $p = A$ and $p = B$)	82
7.3	EPS model: realizations of the uncertain demand matrix $B_{i, f_p}(s^B)\forall i, f_p, p$	85
8.1	Constraint programming: the constraint graph of the first-stage constraints	95
8.2	Constraint programming: the incidence matrix of <i>CSP-EPS-MASTER</i>	96
8.3	Constraint programming: decomposition structure of the variables	97
8.4	Constraint programming: domain reduction by the successive application of three filters	97
8.5	Filter 1: the reduced domain of the operating decisions, represented by the so called operating state tree (<i>OST</i>)	98
8.6	Filter 2: the reduced domains of the capacity allocation decisions, represented by the capacity allocation trees (<i>CATs</i>)	99
8.7	Filter 3: a reduced polymerization sub-domain represented by an index tree (<i>IT</i>)	102
8.8	<i>P-EA</i> : the hierarchical structure of the genotype representation $\mathbf{g}$	109
8.9	<i>P-EA</i> : the genotype of the individual (8.31) represented by markings on trees	110
8.10	Minimal moves on the capacity allocation level $\mathbf{mv}_{cap}$: basic mutation principle	113
8.11	Minimal moves on the operating level $\mathbf{mv}_{op1}$, $\mathbf{mv}_{op2}$, $\mathbf{mv}_{op3}$: basic mutation principle	114
9.1	Progress plot: comparison of the algorithms A and B with respect to their termination at time t_1 and t_2	119
9.2	Scale-up behavior: comparison of the practical complexity of the algorithms A and B	119
9.3	Initialization and preprocessing: graphical representation of the bounds of <i>EPS-2S-MILP</i> from Table 9.1	122
9.4	Specific feasible initialization: the cumulated distribution of 1,000 equally distributed feasible solutions of <i>EPS-2S-MILP</i> , the <i>EEV</i> , and the median	123
9.5	(<i>EX1</i>): <i>ES</i> with infeasible initialization, progress plots and final value ($t = 120$ min) of the best objective found vs. CPU-time	126

9.6	(EX2): <i>ES</i> with feasible initialization, progress plots and final value ($t = 120$ min) of the best objective found vs. CPU-time	127
9.7	(EX4) and (EX5): comparison of <i>P-EA</i> with feasible initialization and with zero initialization, progress plots and final value ($t = 120$ min) of the best objective found vs. CPU-time	128
9.8	Summary of experimental results for the <i>EA</i> -based approaches: statistical analysis of the best objective found after a CPU-time of $t_1 = 30$ min and $t_2 = 120$ min	129
9.9	Comparison of algorithms: progress plot of <i>ES</i> , <i>DDSIP</i> , and <i>P-EA</i> (best lower bound from <i>DDSIP</i> : $-20.6 \times 10^3 EUR$)	130
9.10	<i>EPS-2S-MILP</i> : best solutions found by <i>ES</i> , <i>DDSIP</i> , and <i>P-EA</i>	131
9.11	Robustness to data variation: progress plots of the best objective versus CPU-time for <i>CPLEX</i> , <i>DDSIP</i> (reduced), <i>ES</i> , and <i>P-EA</i>	133
9.12	Scale-up behavior: progress plots of the best objective versus CPU-time	135
B.1	Distribution functions: the geometric distribution, $\mathcal{G}(p_{geo})$	145
B.2	Distribution functions: the (discrete) symmetric geometric distribution, $\mathcal{G}_{sym}(p_{geo})$	146
D.1	<i>DDSIP</i> : Visualization of the measures	154
D.2	<i>DDSIP</i> responses for default settings, see Figure D.1 for an explanation	155
D.3	Summary: <i>DDSIP</i> responses to CONIC BUNDLE parameters	156
D.4	Summary: <i>DDSIP</i> responses to branch-and-bound parameters	156
D.5	Summary: rounding heuristics	157
D.6	Summary: Termination criteria	158
D.7	Summary: Termination criteria	159

List of Tables

3.1	Uncertainties: values of the uncertain demands for four periods	17
4.1	<i>2S-MILP</i> -properties: dimensions and properties	30
4.2	<i>DDSIP</i> : user defined termination criteria	44
5.1	Evolutionary Algorithms: symbols introduced in Section 5.2	49
6.1	Mixed-integer <i>ES</i> : user defined strategy parameters	73
7.1	EPS model: indices	78
7.2	EPS model: variables	79
7.3	EPS model: parameters	80
7.4	EPS model: nominal demands B_{i,f_p}	82
7.5	EPS model: cost coefficients	84
7.6	EPS model: realizations of the uncertain capacity vector $N_i^{max}(s^N)\forall i$. . .	84
7.7	<i>EPS-2S-MILP</i> : <i>MILP</i> model dimensions	87
8.1	Analysis of the constraint satisfaction problem: the variables and their domains	94
8.2	Filter 3: the reduced polymerization sub-domain corresponding to $F_{i,p} = 5$, represented by a linear list with indices	101
8.3	Filter 3: the cardinality of the the reduced polymerization sub-domains corresponding to $F_{i,p}$, calculated by Equation (8.24)	101
8.4	Uniform random feasible initialization: the cardinality of the feasible sub-vectors of operating decisions and of the feasible sub-vectors of allocation decisions	104
8.5	Uniform random feasible initialization: the cardinality of the feasible operating states (nodes of the operating state tree shown in Figure 8.5) . . .	105
8.6	Minimal moves on the polymerization level $\mathbf{mv}_{poly}$: example of the list of neighboring solutions for the individual (8.31).	112

8.7	Specific mutation operator: overview of the weighted set of minimal moves	114
8.8	Maximum number of applications of minimal moves	115
8.9	<i>P-EA</i> : user defined strategy parameters	115
9.1	Initialization and preprocessing: values of the bounds of <i>EPS-2S-MILP</i> and the CPU-time consumed to obtain them	121
9.2	Scale-up behavior: overview of the model instances (scale-up factor: 4) . .	134
A.1	EPS model: realizations s^N of the uncertain capacity vector $N_i^{max}(s^N)\forall i$.	142
C.1	Infeasible initial population	147
C.2	Feasible initial population	147
D.2	Designs for the test of <i>DDSIP</i> : CONIC BUNDLE parameters and termi- nation criterion	152
D.3	Designs for the test of <i>DDSIP</i> : branch-and-bound parameters	153
D.4	Designs for the test of <i>DDSIP</i> : heuristics and <i>MILP</i> integrality gaps . . .	153
D.5	Output from default parameters of <i>DDSIP</i>	154