

Schriften zur quantitativen Betriebswirtschaftslehre und
Wirtschaftsinformatik

herausgegeben von Prof. Dr. Stefan Voß

Andreas Fink

**Software-Wiederverwendung bei der Lösung von
Planungsproblemen mittels Meta-Heuristiken**

Shaker Verlag
Aachen 2000

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Fink, Andreas:

Software-Wiederverwendung bei der Lösung von Planungsproblemen mittels
Meta-Heuristiken / Andreas Fink.

Aachen : Shaker, 2000

(Schriften zur quantitativen Betriebswirtschaftslehre und Wirtschaftsinformatik)

Zugl.: Braunschweig, Techn. Univ., Diss., 2000

ISBN 3-8265-5956-8

Copyright Shaker Verlag 2000

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 3-8265-5956-8

ISSN 1616-1920

Shaker Verlag GmbH • Postfach 1290 • 52013 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • eMail: info@shaker.de

Meinen Eltern

Vorwort

In einem Vorwort sind üblicherweise Gedanken zur Meta-Ebene einer Arbeit enthalten. Dieser Pflicht möchte ich teilweise nachkommen, indem ich mich der Worte derer bediene, die sich klarer ausdrücken können; vgl. die folgende Seite. Andererseits habe ich die angenehme Aufgabe, denen zu danken, die mich bei der Erstellung der Arbeit unterstützt haben.

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Abteilung Allgemeine Betriebswirtschaftslehre, Wirtschaftsinformatik und Informationsmanagement der Technischen Universität Braunschweig und wurde am dortigen Fachbereich für Wirtschafts- und Sozialwissenschaften als Dissertation genehmigt. Bei der Anfertigung der Arbeit wurde ich bestärkt durch meinen Doktorvater, Herrn Prof. Dr. Stefan Voß, dem mein besonderer Dank für seine Unterstützung, sein in mich gesetztes Vertrauen sowie die mir gewährte Freiheit bei der Niederschrift gilt. Weiterhin danke ich Herrn Prof. Dr. Thomas Spengler für die Übernahme des Korreferats und seine konstruktiven Hinweise. Herrn Prof. David L. Woodruff, Ph.D., bin ich für die Zusammenarbeit und seine wertvollen Anmerkungen ebenfalls zum Dank verpflichtet.

Zum Gelingen der Arbeit haben zudem die Unterstützung und das anspornende Interesse meiner Freunde und Kollegen Jürgen Böse, Kai Gutenchwager, Dr. Davorin Haller, Torsten Reiners und Dr. Gabriele Schneiderei beigetragen, denen ich insbesondere auch für die kritische Durchsicht des Manuskripts danke, wodurch unter anderem einige meiner manchmal komplexen Satzkonstruktionen – exklusive dieser – und verwegenen Wortschöpfungen entschärft werden konnten.

Dieses Buch ist meinen Eltern gewidmet, die mich im Rahmen meiner Ausbildung in vielfältiger Weise unterstützt haben, wofür ich mich sehr herzlich bedanke.

Andreas Fink

*„Es ist nicht genug, zu wissen, man muß es auch anwenden.
Es ist nicht genug, zu wollen, man muß es auch tun.“
(Mahatma Gandhi)*

*„Glück kommt nicht von Dingen, die wir besitzen.
Glück kann kommen durch die Arbeit und den Stolz auf das, was wir tun.“
(Johann Wolfgang von Goethe)*

*„Hofstadters Gesetz: Es dauert immer länger, als man denkt;
sogar dann, wenn man Hofstadters Gesetz berücksichtigt.“
(Douglas J. Hofstadter)*

Inhaltsverzeichnis

Inhaltsverzeichnis	ix
Abbildungsverzeichnis	xv
Tabellenverzeichnis	xix
Algorithmenverzeichnis	xxi
Symbolverzeichnis	xxiii
I Grundlagen	1
1 Einführung	3
1.1 Problemstellung	3
1.2 Vorgehensweise und Aufbau	5
2 Systeme zur Entscheidungsunterstützung	7
2.1 Planung und Entscheidung	9
2.1.1 Klassische Entscheidungsmodelle	10
2.1.1.1 Deterministische skalare Entscheidungsmodelle	11
2.1.1.2 Deterministische vektorielle Entscheidungsmodelle	14
2.1.1.3 Entscheidungen bei unvollkommener Information	15
2.1.2 Quantitative Planungsmethoden	16
2.1.2.1 Exakte Methoden	17
2.1.2.2 Heuristische Methoden	18
2.1.3 Praxisrelevanz quantitativer Planungsmethoden	18
2.2 Diskussion des Beitrags verschiedener Fachdisziplinen	23
2.2.1 Operations Research	23
2.2.2 Informatik	26

2.2.3	Wirtschaftsinformatik als Integrationsdisziplin	29
2.3	Zielsetzung der Arbeit	32
3	Wiederverwendung von Software	35
3.1	Herausforderungen	35
3.1.1	Qualitätsfaktoren	38
3.1.2	Nicht-technische Aspekte	40
3.2	Domain Engineering	42
3.2.1	Domänenanalyse	44
3.2.2	Domänenendesign	46
3.2.3	Domänenimplementierung	47
3.3	Objektorientierte Software-Technik	48
3.3.1	Klassen und Objekte	49
3.3.2	Vererbung und dynamischer Polymorphismus	50
3.3.3	Objektorientiertes Design	53
3.3.4	Kritische Bewertung	54
3.4	Generische und Generative Programmierung	56
3.4.1	Generizität	56
3.4.2	Generative Ansätze	60
3.5	Frameworks	62
3.5.1	Adaptionsmechanismen	63
3.5.2	Problemfelder	66
3.5.3	Anwendungsbeispiele	68
3.6	Komponententechnik	68
4	Meta-Heuristiken	71
4.1	Grundlagen	71
4.1.1	Begriffe	71
4.1.2	Lokale Suche	74
4.1.3	Lösungsräume und Nachbarschaften	80
4.1.4	Theoretische Aussagen	81
4.1.4.1	Komplexität der Ermittlung lokaler Optima	82
4.1.4.2	Lösungsqualität	83
4.1.4.3	„No Free Lunch“-Theorem	85
4.1.4.4	Zusammenfassende Bewertung	86
4.2	Simulated Annealing und verwandte Ansätze	88
4.2.1	Konkretisierungen des allgemeinen Simulated-Annealing-Schemas	90
4.2.2	Varianten mit deterministischen Akzeptanzkriterien	92
4.3	Tabu Search	93
4.3.1	Klassische Tabu-Search-Methoden	94

4.3.1.1	Striktes Tabu Search	95
4.3.1.2	Statisches Tabu Search	99
4.3.1.3	Reactive Tabu Search	100
4.3.2	Spezielle Konzepte des Tabu Search	102
4.3.2.1	Vocabulary Building	102
4.3.2.2	Path Relinking	103
4.4	Variation der Nachbarschaft und deren Bewertung	105
4.4.1	Kandidatenlisten-Strategien	105
4.4.2	Variation von Nachbarschaftstiefe und Nachbarschaften	107
4.4.3	Pilot-Methode	109
4.4.4	Strategische Oszillation	110
4.5	Evolutionäre Methoden	111
4.5.1	Grundprinzipien	113
4.5.2	Spezielle Ausprägungen	117
4.5.2.1	Genetische Algorithmen	118
4.5.2.2	Evolutionsstrategien und Evolutionäre Programmierung	118
4.5.2.3	Memetische Algorithmen	119
4.5.2.4	Scatter Search	120
4.6	Kombination und Integration verschiedener Ansätze	121

II Entwicklung eines Frameworks 123

5 Domänenanalyse 125

5.1	Problemspezifische Abstraktionen	126
5.2	Spezifikation von Meta-Heuristiken	129
5.2.1	Einfache lokale Suchmethoden	131
5.2.2	Simulated Annealing und verwandte Ansätze	137
5.2.3	Tabu Search	146
5.2.4	Variation der Nachbarschaft und deren Bewertung	154
5.2.5	Evolutionäre Methoden	158

6 Design 165

6.1	Interdependenzen zur Implementierung	166
6.2	Design-Prinzipien	169
6.2.1	Abbildung der Generizität von Meta-Heuristiken	170
6.2.2	Abbildung der spezifischen Variabilität von Meta-Heuristiken	171
6.2.3	Abbildung der Traversierung der Nachbarschaft	173

6.2.4	Abbildung von Gemeinsamkeiten und Variabilität problemspezifischer Abstraktionen	175
6.3	Architektur	178
6.3.1	Grundlegende Konfigurationsmechanismen	178
6.3.1.1	Statische Konfiguration	178
6.3.1.2	Dynamische Konfiguration	183
6.3.2	Problemspezifische Komponenten	185
6.3.2.1	Problemkomponente	188
6.3.2.2	Lösungskomponente	189
6.3.2.3	Lösungsinformationskomponente	193
6.3.2.4	Nachbarschaftskomponente	194
6.3.2.5	Attributkomponente	196
6.3.2.6	Standardkomponenten	198
6.3.3	Meta-Heuristik-Komponenten	207
6.3.3.1	Einfache lokale Suchmethoden	208
6.3.3.2	Simulated Annealing und verwandte Ansätze	210
6.3.3.3	Tabu Search	213
6.3.3.4	Variation der Nachbarschaft und deren Bewertung	217
6.3.3.5	Evolutionäre Methoden	225
6.3.4	Introspektion	232
7	Implementierung	237
7.1	Technische Rahmenbedingungen	237
7.1.1	Systemumgebung	238
7.1.2	Verwendung spezieller programmiersprachlicher Konstrukte	239
7.1.3	Programmierkonventionen	240
7.2	Problemspezifische Standardkomponenten	242
7.3	Meta-Heuristik-Komponenten	244
7.3.1	Einfache lokale Suchmethoden	245
7.3.2	Simulated Annealing und verwandte Ansätze	247
7.3.3	Tabu Search	250
7.3.4	Variation der Nachbarschaft und deren Bewertung . . .	254
7.3.5	Evolutionäre Methoden	256
7.4	Sonstige Komponenten	262
8	Anwendung und Erweiterung des Frameworks	263
8.1	Anforderungen und Vorgehensweisen	263
8.1.1	Einfache lokale Suchmethoden	265
8.1.2	Simulated Annealing und verwandte Ansätze	267

8.1.3	Tabu Search	268
8.1.4	Variation der Nachbarschaft und deren Bewertung . . .	270
8.1.5	Evolutionäre Methoden	271
8.1.6	Kombination und Integration verschiedener Ansätze . .	273
8.2	Inkrementelle Anwendung	274
8.3	Erweiterung	279
 III Evaluation		281
 9 Vorbemerkungen		283
 10 Sequenzierungsprobleme		287
10.1	Sequenzierung von Produktionsmustern	287
10.1.1	Problembeschreibung	288
10.1.2	Anwendung des Frameworks	290
10.1.3	Ergebnisse	293
10.2	Kontinuierliches Flow-Shop-Problem	296
10.2.1	Problembeschreibung	296
10.2.2	Anwendung des Frameworks	298
10.2.3	Ergebnisse	300
 11 Probleme mit Bit-Vektor-Lösungsräumen		307
11.1	Mehrdimensional beschränktes Auswahlproblem	307
11.1.1	Problembeschreibung	308
11.1.2	Anwendung des Frameworks	309
11.1.3	Ergebnisse	310
11.2	Steiner-Problem in Graphen	312
11.2.1	Problembeschreibung	313
11.2.2	Anwendung des Frameworks	313
11.2.3	Ergebnisse	315
 12 Ring-Netzwerk-Probleme		319
12.1	Konstruktion von Ring-Netzwerken	319
12.1.1	Problembeschreibung	320
12.1.2	Anwendung des Frameworks	322
12.1.3	Ergebnisse	323
12.2	Lastausgleichsproblem in Ringen	326
12.2.1	Problembeschreibung	326
12.2.2	Anwendung des Frameworks	327
12.2.3	Ergebnisse	327

IV Diskussion	331
13 Einordnung der Untersuchungsergebnisse	333
13.1 Modell- und Methoden-Management	333
13.2 Modellierungssprachen	334
13.3 Meta-Heuristik-Software-Komponenten	336
14 Bewertung und Ausblick	339
Literaturverzeichnis	343
Autorenverzeichnis	395
Stichwortverzeichnis	407

Abbildungsverzeichnis

1.1	Aufbau der Arbeit.	6
3.1	Domain Engineering.	43
3.2	Beispiel für ein Merkmalsdiagramm.	45
3.3	Beispiel für ein Klassendiagramm mit Vererbungsbeziehungen.	52
3.4	Liste $\langle T \rangle$ als Beispiel für eine parametrisierte Klasse.	57
3.5	Beispielhafte Architektur eines einfachen Frameworks.	65
4.1	Beispiel für einen bewerteten Lösungsraum.	76
4.2	Beispiel für ein „langsameres Entkommen“ einer strikten Tabu-Search-Methode aus einem großen „Tal“ im Lösungsraum.	98
4.3	Schematische Darstellung eines <i>Path Relinking</i>	104
4.4	Schematische Darstellung des <i>Tunneling</i> unzulässiger Bereiche des Lösungsraums mittels <i>Path Relinking</i>	105
5.1	Nachbarschaft.	128
5.2	Merkmalsdiagramm für <i>SteepestDescent</i>	131
5.3	Merkmalsdiagramm für einfache lokale Suchmethoden.	133
5.4	Merkmalsdiagramm für Simulated Annealing.	137
5.5	Merkmalsdiagramm für Tabu Search.	146
5.6	Merkmalsdiagramm für Evolutionäre Methoden.	158
6.1	Klassendiagramm für <i>SteepestDescent</i> (vereinfacht).	170
6.2	Klassendiagramm für <i>IteratedLocalSearch</i> (vereinfacht).	172
6.3	Traversierung einer Nachbarschaft.	173
6.4	Schnittstellen Lösungsraum und Nachbarschaft (vereinfacht).	176
6.5	Vererbungshierarchie Lösungsraum (vereinfacht).	177
6.6	Realisierung der Anforderungen <i>RNumeric</i> durch eine Konfigurationskomponente <i>CNumeric</i>	179
6.7	Konfigurationskomponente <i>CSteepestDescent</i>	180
6.8	Konfigurationskomponente <i>CIteratedLocalSearch</i>	182

6.9	Statische Parametrisierung von Komponenten über numerische Datenelemente mit Klassenbezug am Beispiel von <i>GeometricCooling</i>	183
6.10	Basisklasse <i>OmgalInterface</i>	184
6.11	Primäre Abhängigkeiten zwischen problemspezifischen Komponenten.	187
6.12	Allgemeine Schnittstelle von Problemkomponenten.	189
6.13	Allgemeine Schnittstelle von Lösungskomponenten.	190
6.14	Allgemeine Schnittstelle von Lösungsinformationskomponenten.	193
6.15	Allgemeine Schnittstelle von Nachbarschaftskomponenten.	194
6.16	Sequenzdiagramm zur Beschreibung der Interaktion von Lösungs- und Nachbarschaftsobjekten.	196
6.17	Allgemeine Schnittstelle von Attributkomponenten.	197
6.18	Vererbungshierarchie Lösungsräume.	200
6.19	Vererbungshierarchie Nachbarschaften.	203
6.20	Standardkomponenten Lösungsinformation.	205
6.21	Standardkomponenten Lösungsattribute.	206
6.22	Allgemeine Schnittstelle und Basisklasse <i>Heuristic</i>	207
6.23	Ableitung spezieller Meta-Heuristik-Komponenten von <i>Heuristic</i>	208
6.24	Komponente <i>IteratedLocalSearch</i>	209
6.25	Schnittstelle <i>NeighborSelection</i> und realisierende Komponenten.	209
6.26	Ableitung von <i>IteratedSteepestDescent</i>	210
6.27	Ableitung von <i>RandomWalk</i>	210
6.28	Komponenten <i>GeneralSimulatedAnnealing</i> und <i>SimulatedAnnealingJetal</i>	211
6.29	Schnittstelle <i>AcceptanceCriterion</i> und realisierende Komponenten.	212
6.30	Schnittstelle <i>CoolingSchedule</i> und realisierende Komponenten.	213
6.31	Schnittstelle <i>Reheating</i> und realisierende Komponenten.	214
6.32	Komponente <i>TabuSearch</i>	214
6.33	Basisklasse <i>TabuCriterion</i> und spezialisierte Klassen.	215
6.34	Schnittstelle <i>TabuNeighborSelection</i> und realisierende Komponenten.	217
6.35	Schnittstelle <i>AspirationCriterion</i> und realisierende Komponenten.	217
6.36	Veranschaulichung des Konzepts der parametrisierten Vererbung.	218
6.37	Definition der Elite-Kandidatenlisten-Nachbarschaftstransformationskomponente.	219
6.38	Zustandsdiagramm zur Elite-Kandidatenlisten-Strategie.	220
6.39	Definition der <i>Aspiration-Plus</i> -Kandidatenlisten-Nachbarschaftstransformationskomponente.	221

6.40	Zustandsdiagramm zur <i>Aspiration-Plus</i> -Kandidatenlisten-Strategie.	222
6.41	Komponente <i>IteratedLocalSearchID</i>	224
6.42	Definition der Nachbarschaftstransformationskomponente <i>PilotNeighborhood</i>	224
6.43	Komponente <i>EvolutionaryAlgorithm</i>	225
6.44	Schnittstelle <i>Individual</i> und realisierende Komponente.	226
6.45	Schnittstelle <i>Population</i> und realisierende Komponente.	227
6.46	Schnittstelle <i>Startgenerator</i> und realisierende Komponenten.	227
6.47	Schnittstelle <i>Marriage</i> und realisierende Komponente.	228
6.48	Schnittstelle <i>Evaluation</i> und realisierende Komponente.	228
6.49	Schnittstelle <i>Selection</i> und realisierende Komponenten.	229
6.50	Schnittstelle <i>Recombination</i> und realisierende Komponenten.	230
6.51	Schnittstelle <i>Mutation</i> und realisierende Komponente.	230
6.52	Schnittstelle <i>MutationSelection</i> und realisierende Komponente.	231
6.53	Schnittstelle <i>MutationChip</i> und realisierende Komponente.	231
6.54	Schnittstelle <i>Replacement</i> und realisierende Komponenten.	232
6.55	Schnittstelle <i>ConvergenceCheck</i> und realisierende Komponenten.	233
6.56	Ableitung spezieller <i>Observer</i> -Klassen von einer Basisklasse <i>Observer</i>	234
10.1	Fallunterscheidung hinsichtlich der Auswirkungen eines 2-Austausch-Zugs auf den Zielfunktionswert.	291
10.2	Maschinenorientiertes Gantt-Diagramm zu einem kontinuierlichen Flow-Shop-Problem.	298
10.3	Abhängigkeit der Lösungsqualität von der Iterationsbegrenzung der Pilot-Methode bei 50 Aufträgen.	302
10.4	Abhängigkeit der Lösungsqualität von der Iterationsbegrenzung der Pilot-Methode bei 100 Aufträgen.	303
11.1	Durchschnittliche Lösungsqualität bei der Anwendung der <i>Reverse Elimination</i> Methode auf mehrdimensional beschränkte Auswahlprobleme in Abhängigkeit von den Parametern <i>tabuDuration</i> und <i>rcsLengthParameter</i>	312
12.1	Beispielhafte Veranschaulichung einer Lösung für ein Ring-Netzwerk-Konstruktionsproblem.	320
12.2	Veranschaulichung einer Problemistanz für das Lastausgleichsproblem in Ringen.	327

12.3 Durchschnittliche Lösungsqualität bei der Anwendung der <i>Reverse Elimination</i> Methode auf Lastausgleichsprobleme in Ringen in Abhängigkeit von den Parametern <i>tabuDuration</i> und <i>rclLengthParameter</i>	329
---	-----

Tabellenverzeichnis

4.1	Beispiel für die <i>Reverse Elimination</i> Methode.	97
8.1	Zusammenfassung der wesentlichen problemspezifischen Anforderungen von Meta-Heuristiken.	265
10.1	Ergebnisse der Anwendung von Konstruktionsmethoden sowie von <i>Steepest Descent</i>	294
10.2	Ergebnisse der Anwendung von <i>Steepest Descent</i> , Simulated Annealing sowie statischem Tabu Search.	295
10.3	Ergebnisse der Anwendung von Tabu Search.	295
10.4	Ergebnisse der Anwendung von Konstruktionsmethoden.	301
10.5	Ergebnisse der Anwendung von <i>Steepest Descent</i>	303
10.6	Ergebnisse der Anwendung von Simulated Annealing.	304
10.7	Ergebnisse der Anwendung von statischem Tabu Search.	304
10.8	Ergebnisse der Anwendung von striktem Tabu Search und Reactive Tabu Search.	305
10.9	Durchschnittliche Rechenzeiten (in CPU-Sekunden) bei der Anwendung verschiedener Methoden auf charakteristische Datensätze.	305
11.1	Ergebnisse der iterativen Anwendung von <i>Steepest Descent</i> auf mehrdimensional beschränkte Auswahlprobleme unter sukzessiver Steigerung der Nachbarschaftstiefe.	311
11.2	Ergebnisse der Anwendung von iterativen Verbesserungsmethoden und Tabu Search bei Verwendung der Methode der sukzessiven besten Einfügung.	316
11.3	Ergebnisse der Anwendung von iterativen Verbesserungsmethoden und Tabu Search bei Verwendung der Kürzeste-Wege-Heuristik.	316
11.4	Ergebnisse der Kombination von Kandidatenlisten-Strategien mit Tabu-Search-Methoden bei Verwendung der Methode der sukzessiven besten Einfügung.	317

12.1	Einordnung verschiedener Problemstellungen unter das GRNDP.	321
12.2	Ergebnisse der Anwendung von Tabu Search für Problem- instanzen zum SRNDPR.	324
12.3	Ergebnisse der Anwendung von Tabu Search für STSP/OP- Instanzen mit bekannten Optimallösungen.	325
12.4	Ergebnisse der Anwendung von Tabu Search für schwierige STSP/OP-Instanzen.	325
12.5	Ergebnisse der iterativen Anwendung von <i>Steepest Descent</i> auf Lastausgleichsprobleme in Ringen unter sukzessiver Stei- gerung der Nachbarschaftstiefe.	328
12.6	Ergebnisse der Anwendung von Reactive Tabu Search.	329

Algorithmenverzeichnis

1	SteepestDescent (vorläufig).	132
2	IteratedLocalSearch.	134
3	BestPositiveNeighbor.	134
4	BestNeighbor.	135
5	FirstPositiveNeighbor.	135
6	RandomNeighbor.	135
7	SteepestDescent.	136
8	FirstDescent.	136
9	IteratedSteepestDescent.	136
10	RandomWalk.	136
11	IteratedSteepestDescentWithPerturbationRestarts.	137
12	GeneralSimulatedAnnealing.	138
13	ClassicExponentialAcceptanceCriterion.	140
14	ClassicThresholdAcceptanceCriterion.	140
15	AbsoluteThresholdAcceptanceCriterion.	140
16	GeometricCooling.	141
17	HajekCooling.	141
18	LundiMeesCooling.	141
19	DowlandCooling.	142
20	GreatDelugeCooling.	142
21	RecordToRecordTravelCooling.	142
22	ReheatingToHalfOfInitial.	142
23	ReheatingToBest.	143
24	ReheatingToAverageOfBestAndInitial.	143
25	ClassicSimulatedAnnealing.	143
26	ThresholdAccepting.	143
27	GreatDeluge.	144
28	RecordToRecordTravel.	144
29	SimulatedAnnealingJetal.	145
30	TabuSearch.	147
31	BestAdmissibleNeighbor.	147

32	BestNeighborConsideringPenalties.	148
33	StrictTabuCriterionByTrajectory.	149
34	REMTabuCriterion.	150
35	StaticTabuCriterion.	151
36	ReactiveTabuCriterion.	152
37	StrictTabuSearch.	153
38	REMPen.	154
39	StaticTabuSearch.	154
40	ReactiveTabuSearch.	154
41	EliteCandidateList.	155
42	AspirationPlusCandidateList.	156
43	IteratedLocalSearchID.	157
44	PilotMethod.	157
45	EvolutionaryAlgorithm.	159
46	RandomStart.	160
47	MutationStart.	160
48	Marriage.	161
49	Evaluation.	162
50	Mutation.	163

Symbolverzeichnis

Kontext: Meta-Heuristiken

$\alpha, \alpha_1, \alpha_2$	Parameter für Abkühlungsschemata
b	Parameter zur Veränderung des Wiederholungsintervalls beim Simulated Annealing
β	Anspruchsniveauparameter bei der <i>Aspiration-Plus</i> -Kandidatenlisten-Strategie
C	Konfiguration einer Heuristik
C_H	Meta-Heuristik-spezifische Konfiguration einer Heuristik
C_P	problemspezifische Konfiguration einer Heuristik
d	Nachbarschaftstiefe
δ	Approximationsgrad
δ_1, δ_2	Parameter für Reactive Tabu Search
Δ	Zielfunktionswertveränderung
$f(s)$	Bewertung einer Lösung s (Zielfunktion)
$f(\mu)$	Bewertung eines Zugs μ
Γ	vollständig geordnete Menge
$h(s)$	Lösungsinformation (Hash-Wert) zu einer Lösung s
H	(Meta-) Heuristik
i	Iterationszähler
I_{\max}	maximale Iterationsanzahl
K	Anzahl von Zielen
κ	maximale Tabu-Listen-Länge
l	aktuelle Tabu-Listen-Länge
$n(s)$	Nachbar einer Lösung s
N	Nachbarschaft
N_d	Nachbarschaft einer Tiefe d
N_S	Nachbarschaft eines Lösungsraums S
N_S^μ	Nachbarschaft (unter expliziter transformativer Betrachtung)
ν	Schwellenwert-Parameter für Tabu-Kriterien
ω	externes Abbruchkriterium für einen Algorithmus

Ω	Abbruchkriterium für einen Algorithmus
p	Probleminstanz
P	Problem
Π	Permutation
Ψ	Attributmenge
$\psi(\mu)$	Attributmenge eines Zugs μ
$\psi^+(\mu)$	Menge der Plus-Attribute eines Zugs μ
$\psi^-(\mu)$	Menge der Minus-Attribute eines Zugs μ
$\psi(s)$	Attributmenge einer Lösung s
ϕ	Platzhalter für ungültig oder irrelevant
R_{\max}	maximale Wiederholungsanzahl
s	Lösung
S	Lösungsraum
S_P	Lösungsraum für ein Problem P
S_h	Menge zur (komprimierten) Abbildung des Lösungsraums
σ	Tabu-Strafkostenparameter
t	Rechenzeit eines Algorithmus
T_{\max}	maximale Rechenzeit zur Ausführung eines Algorithmus
τ	Temperatur- oder Toleranzparameter
θ	Glättungsparameter für Reactive Tabu Search
μ	Zug (zu einer Nachbarlösung)
x	Entscheidungsvariable
\mathcal{X}	Handlungsalternativenmenge
ζ_1, ζ_2	Parameter für Reactive Tabu Search
$z(x)$	Bewertung einer Alternative x (Zielfunktion)

Kontext: Problemstellungen

a	untere Schranke für die zu erreichende Ertragssumme
a_{it}	Nettoauszahlung für Investitionsobjekt i in Periode t
α	Skalierungsparameter bei der Ring-Netzwerk-Konstruktion
α_i	Häufigkeit des Produktionsmusters i
b	obere Grenze für die Gesamtkosten einer Ring-Netzwerk-Konstruktion
B	Basisknotenmenge
β	Skalierungsparameter bei der Ring-Netzwerk-Konstruktion
c_i	Kapitalwert eines Investitionsobjekts i
c_{ij}	Bewertung einer Kante (i, j)
C	Produktionsmustermatrix
γ	Skalierungsparameter bei der Ring-Netzwerk-Konstruktion

d_{ik}	Mindestverzögerung zwischen zwei Aufträgen i und k
D	Verzögerungsmatrix
$\#d$	Anzahl der Kommunikationsbedarfe
δ	Skalierungsparameter bei der Ring-Netzwerk-Konstruktion
E	Kantenmenge
G	Graph
h	maximale Knotenanzahl im Ring
m, n	Anzahl (problemspezifisch)
p_i	Kosten bei Nichtaufnahme eines Knotens i in den Ring
r_t	Budget in Periode t
r_{ij}	Ertrag bzw. Kommunikationsbedarf des Knotenpaars (i, j)
R	Ring
t	Periode
t_{ij}	Fertigungszeit von Auftrag i auf Maschine j
T	Periodenanzahl bzw. Teilgraph (Steiner-Baum)
Q	Basisknotenmenge
V	Knotenmenge
x_i	binäre Entscheidungsvariable