

# Heuristische Lösungsverfahren für das ressourcenbeschränkte Projektplanungsproblem

## Bachelorarbeit

zur Erlangung des akademischen Grades  
„Bachelor of Science (B.Sc.)“ im Studiengang  
Wirtschaftswissenschaft

der Wirtschaftswissenschaftlichen Fakultät  
der Leibniz Universität Hannover

vorgelegt von

**Robert Matern**

geb. am 7. März 1987 in Tschimkent  
(Matrikel-Nr. 2798160)

**Erstprüfer: Prof. Dr. Stefan Helber**

Hannover, den 17. September 2014

# Sperrvermerk

Veröffentlichungen über den Inhalt der Arbeit sind nur mit schriftlicher Genehmigung der Volkswagen AG zugelassen.

Die Ergebnisse, Meinungen und Schlüsse dieser Abschlussarbeit sind nicht notwendigerweise die der Volkswagen AG.

Die vorliegende Arbeit ist nur den Mitarbeitern des Volkswagen Konzerns, den Korrektoren sowie den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Eine Veröffentlichung oder Weitergabe der Bachelorarbeit, außer zur Bewertung im Rahmen der Studienprüfung, außerhalb der Volkswagen AG ist nicht erlaubt.

Hannover, den 17. September 2014

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>iv</b>
<b>Symbolverzeichnis</b>	<b>v</b>
<b>Tabellenverzeichnis</b>	<b>viii</b>
<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung in die ressourcenbeschränkte Projektplanung</b>	<b>1</b>
<b>2 Das Grundmodell der ressourcenbeschränkten Projektplanung</b>	<b>4</b>
2.1 Mathematische Modellformulierung . . . . .	4
2.2 Notwendigkeit einer Heuristik zum Lösen des RCPSP . . . . .	8
2.3 Eröffnungsverfahren zur Bestimmung der Ablauffolge einer Pro- jektplanung . . . . .	9
<b>3 Heuristische Lösungsverfahren für das RCPSP</b>	<b>14</b>
3.1 Verfahren der simulierten Abkühlung . . . . .	14
3.2 Hybride Heuristikantastung mit selektiver Rückwärts-Vorwärts- Planung . . . . .	17
3.3 Evolutionäre Lokalsuche aufgrund Variablenachbarschaft . . . .	20
3.4 Hybride elektromagnetische Metaheuristik . . . . .	25
<b>4 Leistungsbewertung der vorgestellten Heuristiken</b>	<b>30</b>
<b>5 Schlussbemerkung</b>	<b>33</b>
5.1 Zusammenfassung . . . . .	33
5.2 Ausblick . . . . .	34
<b>Literaturverzeichnis</b>	<b>36</b>
<b>Anhang</b>	<b>42</b>

# Abkürzungsverzeichnis

BF	Backward-Forward(-Methode)/ Rückwärts-Vorwärts(-Methode)
EM	Elektromagnetische Metaheuristik
GA	Genetischer Algorithmus
GRASP	Greedy Randomized Adaptive Search Procedures
MLFT	Minimum Latest Finish Time/ minimal-spätester Fertigungszeit
OR	Operations Research
PNV	Pfadneuverknüpfung
PSPLIB	Project Scheduling Problem Library
RBRS	Regret-Based Biased Random Sampling
RCPSP	Resource-Constrained Project Scheduling
SA	Simulated Annealing
SGS	Schedule Generation Scheme
SS	Scatter Search

# Symbolverzeichnis

$\alpha$	Dämpfungsfaktor
$\beta$	Ausmaß der Verzerrung
$\delta$	Dimension des Lösungsraums
$\Delta$	Differenzwert
$\rho$	Wert des Bedauerns für einen Arbeitsgang $j$
$\Theta$	Temperatur beim Verfahren des Simulated Annealing
$\psi(j)$	Wahrscheinlichkeit einer Auswahl eines ungeplanten Arbeitsgangs $j$
$\omega_{j_r j_{r+1}}$	Anzahl von Lösungen einer aktuellen Population, bei dem Arbeitsgang $j$ in eine Liste aufgenommen wird, nachdem der Arbeitsgang $i$ gelistet wurde
$A(j)$	Menge von Vorgängern, bevor der Arbeitsgang $j$ in die Liste $L$ aufgenommen wurde
$\mathcal{A}$	Menge an Vorrangsbeziehungen
$\mathcal{B}$	Teilmenge einer Lösungsmenge
$Block_j$	Menge von Arbeitsgängen, die zeitgleich mit dem Arbeitsgang $j$ geplant werden
$C$	Menge an Paaren von Vorrangsbeziehungen von Arbeitsgang $j$
$d_j$	Dauer von Arbeitsgang $j$
$\mathcal{D}_n$	Menge an Arbeitsgängen, die zum Zeitpunkt $t_n$ in den Planungsprozess überführt werden sollen
$\mathcal{E}$	Pool mit Lösungen
$EFT_j$	Frühester Fertigstellungszeitpunkt von Arbeitsgang $j$
$f(y)$	Funktionswert von Punkt $y$
$First$	Frühestmögliche Position in einem $Block_j$
$FT_j$	Fertigstellungszeitpunkt von Arbeitsgang $j$

---

$G_S$	Richtungsgraph der Planung $S$
$i$	Vorgänger von Arbeitsgang $j$
$I$	Menge an Iterationen
$j$	Arbeitsgang
$k_{jr}$	Belastung von Ressourcen $r$ bei Durchführung von Arbeitsgang $j$
$K_r$	Kapazität von Ressourcen $r$
$KA_i$	Vektor der Kraftausprägung vom Punkt $y_i$
$L$	Ablauffolge/ Vorgangsliste
$Last$	Spätestmögliche Position in einem $Block_j$
$LFT_j$	Spätester Fertigstellungszeitpunkt von Arbeitsgang $j$
$\mathcal{L}_t$	Menge von Arbeitsgängen $j$ , die zum Zeitpunkt $t$ starten können, ohne Verletzung der Reihenfolgerestriktionen
$m$	Vorherige Iteration
$n$	Iteration
$N(\cdot)$	Nachbarschaft von Punkt $y$ oder Liste $L$
$\mathcal{N}_n$	Menge an Arbeitsgängen, die sich zum Zeitpunkt $t_n$ im Planungsprozess befinden
$p$	Zufallswert
$P$	Wahrscheinlichkeit
$\mathcal{P}_j$	Menge der Vorgänger von Arbeitsgang $j$
$q_i$	Ladung von Punkt $i$
$q_{ij}$	Ladung in Relation der Entfernung zwischen Punkt $i$ und $j$
$r$	Erneuerbare Ressource
$RefSet$	Referenzmenge
$\mathcal{R}$	Menge der erneuerbaren Ressourcen
$S_j$	Geplante Startzeit von Arbeitsgang $j$
$S(L)$	Aktive Planung bzw. Lösung anhand Liste $L$
$\mathcal{S}_t$	Menge der Arbeitsgänge, die zum Zeitpunkt $t$ bearbeitet werden bzw. bereits eingeplant wurden
$Shift_{ij}$	Bewegung des Arbeitsgangs $i$ , nachdem Arbeitsgang $j$ gemeinsam mit allen Nachfolgern geplant wurde

---

$t$	Zeitpunkt/ Periode
$t_{n,m}$	Zeitpunkt nach Iteration $n, m$
$\mathcal{T}$	Menge an Perioden mit $t, \tau = 1, \dots, T$
$TS(L)$	Tabu-Status für eine Planung $S(L)$
$\mathcal{U}$	Menge an nicht-erledigten Arbeitsgängen
$v(\cdot)$	Prioritätswert von Arbeitsgang $j, i$
$\mathcal{V}_n$	Menge an Arbeitsgängen, die bis zum Zeitpunkt $t_n$ im Planungsprozess bereits eingeplant wurden
$w$	Populationsgröße
$x_{jt}$	Binäre Entscheidungsvariable, ob Arbeitsgang $j$ in Periode $t$ beendet wird
$y$	Punkt im Lösungsraum
$z$	Vorbestimmte Anzahl von Punkten

## Tabellenverzeichnis

1	Durchschnittliche Abweichung (%) von der optimalen Fertigstellungsdauer; $J = 30$ . . . . .	31
2	Durchschnittliche Abweichung (%) vom kritischen Pfad als untere Schranke; $J = 60$ . . . . .	31
3	Durchschnittliche Abweichung (%) vom kritischen Pfad als untere Schranke; $J = 120$ . . . . .	32
4	Eigenschaften der betrachteten Heuristiken . . . . .	33



# Abbildungsverzeichnis

1	Möglicher Aufbau einer Reihenfolgebeziehung eines Projekts . .	10
2	Mögliche Ablauffolge eines Projekts . . . . .	10
3	Zyklische Umsortierung eines Arbeitsgangs . . . . .	11
4	Ablauffolge nach seriellem SGS . . . . .	12
5	Ablauffolge nach parallelem SGS . . . . .	13
6	Grafische Veranschaulichung des EM-Algorithmus . . . . .	42

# 1 Einleitung in die ressourcenbeschränkte Projektplanung

Die Projektarbeit spielt über mehrere Dekaden eine wichtige Rolle bei der Aufgabenabwicklung in Wirtschaft und Verwaltung.<sup>1</sup> Dabei wird unter dem Begriff Projekt verstanden:

**Definition 1.** „*Ein Projekt ist eine zeitlich befristete, relativ innovative und risikobehaftete Aufgabe von erheblicher Komplexität, die aufgrund ihrer Schwierigkeit und Bedeutung meist ein gesondertes Projektmanagement erfordert.*“<sup>2</sup>

Nach dieser Definition geht eine zeitliche Restriktion mit einem Projekt einher. Durch das Zerlegen des Projekts in einzelne Arbeitsgänge wird versucht die Komplexität zu reduzieren und eine geordnete Abfolge der Arbeitsgänge zu erstellen, um das Projektziel zu erreichen.<sup>3</sup> Projektziele können unterschiedlich kategorisiert werden, z. B. in Sach-, Termin- oder Kostenziele.<sup>4</sup>

Ein Arbeitsgang eines Projekts hat nach DIN 69900 einen definierten Anfang sowie ein definiertes Ende und dient für das Projekt als Ablaufelement zur Beschreibung eines bestimmten Geschehens.<sup>5</sup> Trotz der Zerlegung besitzen die einzelnen Arbeitsgänge des Projekts eine Beziehung, mit der die Reihenfolge der Ablauffolge bestimmbar ist.<sup>6</sup> Ein Arbeitsgang ist i. d. R. verbunden mit dem Einsatz von Ressourcen, welche wiederum mit Kosten verbunden sind. Dementsprechend versucht ein effizientes Projektmanagement, neben der Einhaltung des Projektziels, auch den Einsatz der Ressourcen zu minimieren.<sup>7</sup>

---

<sup>1</sup>Vgl. Zimmermann et al. (2006b), S. VI

<sup>2</sup>Vgl. Voigt und Schewe (2014)

<sup>3</sup>Vgl. Zimmermann et al. (2006b), S. 4

<sup>4</sup>Vgl. Felkai und Beiderwieden (2011), S. 52

<sup>5</sup>Vgl. DIN 69900 (2009), S. 15

<sup>6</sup>Vgl. Kellenbrink (2014b), S. 6-7

<sup>7</sup>Vgl. Bartels (2009), S. 11-12

**Definition 2.** „Allgemein bezeichnet der Begriff Projektmanagement alle leitenden und administrativen Aktivitäten, die zur Durchführung eines Projektes notwendig sind. Es beschreibt die Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln zur zielorientierten Durchführung großer Vorhaben.“<sup>8</sup>

Eine Möglichkeit, das Projektziel unter minimaler Ressourcenverwendung zu erreichen, ist die effiziente Planung der Ablauffolge der Arbeitsgänge eines Projekts.<sup>9</sup> Damit ist es möglich, mehrere Projekte bei einer gegebenen Zeitvorgabe unter Einhaltung von Ressourcenrestriktionen fertigzustellen bzw. bei konstanter Ressourcenkapazität ein Projekt in kürzerer Zeit abzuschließen. Dadurch hat die Projektplanung besondere Bedeutung in dem Projektmanagement.<sup>10</sup>

**Definition 3.** „Projektplanung ist die Planung aller Arbeitsgänge eines Projekts durch Zuweisung eines Startzeitpunktes, so dass die Zeitbeziehung zwischen den Vorgängen eingehalten und knappe Ressourcenkapazitäten nicht überschritten werden.“<sup>10</sup>

Eine effiziente Projektplanung reduziert die gesamte Fertigstellungsdauer eines Projekts und gestaltet den Verbrauch der zur Verfügung stehenden Ressourcen über die Laufzeit des Projekts wirkungsvoll. Somit handelt es sich um ein mathematisches Optimierungsproblem, bei dem für ein Projekt die Ressourcenbeschränkung über die Laufzeit einzuhalten ist und die Fertigstellungszeit minimiert werden soll. Für das ressourcenbeschränkte Projektplanungsproblem gibt es Verfahren der exakten und heuristischen Lösung. In der Praxis wird i. d. R. auf Heuristiken zurückgegriffen,<sup>11</sup> worauf auch die vorliegende Arbeit ihren Fokus legt.<sup>12</sup>

Das Projektmanagement in der Automobilindustrie benötigt aufgrund eines hohen Komplexitätsgrades eine effiziente Projektplanung.<sup>13</sup> Dabei gibt es verschiedene Lösungserfahren für das ressourcenbeschränkte Projektplanungsproblem mit unterschiedlicher Lösungsgüte. Ziel der vorliegenden Arbeit ist es, einen Beitrag für die Erforschung des ressourcenbeschränkten Projektplanungsproblems zu leisten. Dies erfolgt durch ausführliches Beschreiben von vier Heu-

---

<sup>8</sup>Vgl. Hering (2014), S. 1-3; in Anlehnung an DIN 69901 und ISO 21500:2012-09

<sup>9</sup>Vgl. Bartels (2009), S. 11-12

<sup>10</sup>Vgl. Zimmermann et al. (2006b), S. VI

<sup>11</sup>Vgl. Herroelen (2005), S. 420

<sup>12</sup>Vgl. Kapitel 2.2

<sup>13</sup>Vgl. Lührig (2006), S. 26

ristiken zum Lösen des ressourcenbeschränkten Projektplanungsproblems in Kapitel 3, damit diese Ausführungen in einer späterer Entwicklung eines effizienten Lösungsansatz überführt werden können. Zu diesem Zweck wird in Kapitel 2 zunächst das ressourcenbeschränkte Projektplanungsproblem näher betrachtet sowie die Grundlagen der Heuristiken.

## 2 Das Grundmodell der ressourcenbeschränkten Projektplanung

### 2.1 Mathematische Modellformulierung

Zur Sicherstellung des Planungserfolgs mittels Terminierung eines Projekts muss neben der Reihenfolgerestriktion auch der Ressourcenbedarf der unterschiedlichen Arbeitsgänge sichergestellt werden, da Projekte meist ein beschränktes Ressourcenkontingent haben.<sup>14</sup> Mit der Einhaltung des Ressourcenbedarfs ist es für das Projekt möglich, alle in Bearbeitung befindlichen Arbeitsgänge auszuführen und damit das Projekt abzuschließen. Ressourcen können über das ganze Projekt limitiert sein, wie bspw. das Projektbudget, oder können nach einer bestimmten Anzahl von Perioden erneuert werden.<sup>15</sup> Erneuerbaren Ressourcen können bspw. die Produktionskapazität einer Maschine oder der Personaleinsatz für das Projekt sein. In dieser Arbeit wird der Fokus auf erneuerbare Ressourcen gelegt.

Zur Lösung des ressourcenbeschränkten Projektplanungsproblems kann das Modell *Resource-Constrained Project Scheduling Problem (RCPSP)* genutzt werden. Das Modell RCPSP legt durch Festlegung der Aktivitätsstartzeitpunkte unter Einhaltung der Startzeitpunkt- bzw. der Vorrangsbedingung der einzelnen Arbeitsgänge sowie der Kapazitätsbeschränkung der erneuerbaren Ressourcen den Projektgrundablauf zur Zielerreichung der Minimierung der Projektdauer fest.<sup>16</sup> Die Zielfunktion des RCPSP zur Minimierung der Projektdauer ist weit verbreitet,<sup>17</sup> aber auch andere Ausprägungen sind möglich.<sup>18</sup>

---

<sup>14</sup>Vgl. Kellenbrink (2014b), S. 11

<sup>15</sup>Vgl. Neumann-Braun et al. (2003), S. 21-22

<sup>16</sup>Vgl. Demeulemeester und Herroelen (2011), S. 23

<sup>17</sup>Vgl. Drexel et al. (1997), S. 98

<sup>18</sup>Vgl. Talbot (1982), S. 1200

Nachfolgend wird das deterministische RCPSP in diskreter Zeit formuliert.<sup>19</sup> Da es sich um eine mathematische Modellformulierung in diskreter Zeit handelt, sind die Zeiteinheiten gleich den Perioden  $t, \tau$ .

### Indizes und (geordnete) Mengen

$t, \tau \in \mathcal{T}$  Perioden mit  $t, \tau = 1, \dots, T$

### Parameter

$EFT_j$  Frühester Fertigstellungszeitpunkt von Arbeitsgang  $j$

$LFT_j$  Spätester Fertigstellungszeitpunkt von Arbeitsgang  $j$

### Entscheidungsvariablen

$$x_{jt} \begin{cases} 1, & \text{wenn Arbeitsgang } j \text{ in Periode } t \text{ beendet wird} \\ 0, & \text{sonst} \end{cases}$$

### Modell RCPSP

$$\min Z = \sum_{t=EFT_J}^{LFT_J} t \cdot x_{Jt} \quad (1)$$

unter Beachtung der Restriktionen

$$\sum_{t=EFT_j}^{LFT_j} x_{jt} = 1 \quad j \in \mathcal{J} \quad (2)$$

$$\sum_{t=EFT_i}^{LFT_i} t \cdot x_{it} \leq \sum_{t=EFT_i}^{LFT_i} (t - d_j) \cdot x_{jt} \quad j \in \mathcal{J}; i \in \mathcal{P}_j \quad (3)$$

$$\sum_{j=2}^{J-1} k_{jr} \cdot \sum_{\tau=t}^{t+d_j-1} x_{j\tau} \leq K_r \quad r \in \mathcal{R}; t \in \mathcal{T} \quad (4)$$

$$x_{jt} \in \{0, 1\} \quad j \in \mathcal{J}; t \in \{EFT_j, \dots, LFT_j\} \quad (5)$$

<sup>19</sup>Vgl. Klein (2000), S. 79-80; Modell und Notation wurde dem Modell von Kellenbrink (2014b), S. 14-15, entnommen.

Ein Projekt hat  $J$  unterschiedliche Arbeitsgänge. Jeder Arbeitsgang  $j$  hat eine definierte Menge von zu erledigenden Vorgängerarbeitsgängen  $i \in \mathcal{P}_j$  und die Arbeitsgänge müssen zur Fertigstellung des Projekts topologisch abgearbeitet werden. D. h. der Vorgänger  $i$  hat stets eine kleinere Ordnungszahl als sein Nachfolger  $j$  ( $i < j$ ) und muss für den weiteren Projektverlauf beendet sein. Die Bearbeitungsdauer eines Arbeitsgangs  $j$  wird mit dem Parameter  $d_j$  festgelegt. Bei dem RCPSP in diskreter Zeit wird die Annahme getroffen, dass die Dauer ganzzahlig ist. Der Startzeitpunkt des Projekts ist  $t = 0$  und erstreckt sich über einen Gesamtzeitraum von  $T$  Perioden. Um die Reihenfolgebedingungen einzuhalten, werden einem Projekt die zwei Dummy-Arbeitsgänge „Beginn“ ( $j = 1$ ) und „Ende“ ( $j = J$ ) hinzugefügt, welche mit einer Dauer von 0 Zeiteinheiten bewertet werden.<sup>20</sup> Die benötigten Kapazitäten der erneuerbaren Ressource  $r$  bei Durchführung von Arbeitsgang  $j$  wird durch  $k_{jr}$  definiert. Die Ressourcen  $r \in \mathcal{R}$  sind in einer Periode innerhalb des Umfangs ihrer Kapazität  $K_r$  nutzbar. Da es sich um erneuerbare Ressourcen handelt, sind diese zu jeder neuen Periode in vollem Umfang erneut nutzbar, wobei nichtverbrauchte Ressourcen nicht auf nachfolgende Perioden übertragbar sind.<sup>21</sup> Der Modellformulierung in diskreter Zeit wird die binäre Entscheidungsvariable  $x_{jt}$  hinzugefügt, damit der Fertigstellungszeitpunkt der einzelnen Arbeitsgänge  $j$  festgelegt werden kann.<sup>22</sup>

Mit der Zielfunktion (1) wird der Fertigstellungszeitpunkt des Projekts minimiert. Die Nebenbedingung (2) sorgt dafür, dass ein Arbeitsgang  $j$  nur jeweils zwischen dem frühesten und dem spätesten Fertigstellungszeitpunkt einmalig durchgeführt wird. Die Reihenfolgerestriktion wird mit der Nebenbedingung (3) eingehalten. Sie stellt sicher, dass jeder Vorgänger  $i \in \mathcal{P}_j$  beendet ist, bevor der Arbeitsgang  $j$  startet. Der Parameter  $K_r$  legt die Kapazitätsgrenze für eine erneuerbare Ressource  $r$  je Periode  $t$  fest und mit der Nebenbedingung (4) wird diese zum einen formal dargestellt und zum anderen wird der Ressourcenverzehr während der gesamten Bearbeitungsdauer der Fertigstellung beachtet. Mit der Nebenbedingung (5) wird die Binärvariable  $x_{jt}$  für den Zeitraum  $t = \{EFT_j, \dots, LFT_j\}$  definiert, da aufgrund der Reihenfolgebeziehung (3) der jeweils betrachtete Arbeitsgang nur in diesem Zeitraum fertiggestellt werden kann. Die gemischt-ganzzahlige Modellformulierung lässt sich durch Standard-Lösungsverfahren exakt lösen.<sup>23</sup>

<sup>20</sup>Vgl. Zimmermann et al. (2006b), S. 66

<sup>21</sup>Vgl. Kellenbrink (2014b), S. 12

<sup>22</sup>Vgl. Pritsker et al. (1969), S. 94

<sup>23</sup>z. B. mittels eines Branch-and-Bound-Verfahrens, Vgl. Kellenbrink (2014b), S. 14

Bei der Modellformulierung in stetiger Zeit wird die reellwertige Entscheidungsvariable  $FT_j$ , die den Fertigstellungszeitpunkt vom Arbeitsgang  $j$  beschreibt,<sup>24</sup> und die Menge  $\mathcal{S}_t$  der bis zum Zeitpunkt  $t$  bearbeiteten Arbeitsgänge definiert.<sup>25</sup> Durch die Annahme, dass die Bearbeitungsdauer  $d_j$  nur ganzzahlige Werte annimmt, wird der Ablauf des Projekts ebenfalls eine ganzzahlige Lösung sein.<sup>26</sup>

### Indizes und (geordnete) Mengen

$j, i \in \mathcal{J}$	topologisch sortierte Arbeitsgänge mit $i = 1, \dots, J$
$j \in \mathcal{S}_t$	Menge der Arbeitsgänge, die zum Zeitpunkt $t$ bearbeitet werden; wobei $\mathcal{S}_t := \{j \mid (FT_j - d_j) \leq t \leq FT_j\}$
$i \in \mathcal{P}_j$	Menge der Vorgänger von Arbeitsgang $j$
$r \in \mathcal{R}$	Menge der erneuerbaren Ressourcen
$t \in [0, T]$	Zeitpunkt

### Parameter

$d_j$	Dauer von Arbeitsgang $j$
$k_{jr}$	Belastung von Ressourcen $r$ bei Durchführung von Arbeitsgang $j$
$K_r$	Kapazität von Ressourcen $r$

### Entscheidungsvariablen

$FT_j$	Fertigstellungszeitpunkt von Arbeitsgang $j$
--------	--

### Modell RCPSP

$$\min Z = FT_J \quad (6)$$

unter Beachtung der Restriktionen

$$FT_i \leq FT_j - d_j \quad j \in \mathcal{J}; i \in \mathcal{P}_j \quad (7)$$

<sup>24</sup>Vgl. Talbot (1982), S. 1200

<sup>25</sup>Vgl. Zimmermann et al. (2006a), S. 124 und S. 207; Modell und Notation wurde Kellenbrink (2014b), S. 13, entnommen.

<sup>26</sup>Vgl. Kellenbrink (2014b), S. 14



$$\sum_{j \in \mathcal{S}_t} k_{jr} \leq K_r \quad r \in \mathcal{R}; t \in [0, T] \quad (8)$$

$$FT_j \geq 0 \quad j \in \mathcal{J} \quad (9)$$

Die Zielfunktion (6) minimiert die Fertigstellungszeit des gesamten Projekts. Die Nebenbedingung (7) sichert, dass kein Arbeitsgang  $j$  beendet werden kann, wenn nicht alle vorherigen Arbeitsgänge  $i \in \mathcal{P}_i$  zuvor beendet wurden. Durch Einhaltung der Nebenbedingung (8) wird sichergestellt, dass die Kapazitätsrestriktionen der Ressource  $r$  für die bisher erledigten Arbeitsgänge  $j \in \mathcal{S}_t$  innerhalb der Periode  $t$  erfolgt und die Nebenbedingung (9) ist die Nichtnegativitätsbedingung für die Entscheidungsvariable  $FT_j$ .

Beide mathematischen Modellformulierungen sind für das Grundverständnis zum Lösen des ressourcenbeschränkten Projektplanungsproblems mittels der nachfolgend in Kapitel 3 beschriebenen Heuristiken notwendig, weshalb sie in diesem Kapitel beschrieben sind.

## 2.2 Notwendigkeit einer Heuristik zum Lösen des RCPSP

Das Modell RCPSP wurde als  $\mathcal{NP}$ -Schwer befunden<sup>27</sup> und somit ist kein deterministischer Algorithmus bekannt, der das Problem in polynomialer Zeit löst.<sup>28</sup> In dieser Arbeit werden heuristische Lösungsverfahren betrachtet. Durch Abweichung der Forderung eine optimale Lösung zu berechnen und durch kleine Änderungen der Lösungsanforderungen werden gewaltige Komplexitätssprünge durch Heuristiken ermöglicht. Damit werden in vertretbarer Zeit gute Lösungen generiert oder es wird u. U. auch die Möglichkeit geschaffen die Problemstellung exakt zu lösen.<sup>29</sup>

Einen Überblick über Lösungsansätze für das RCPSP liefern Demeulemeester (2002), Herroelen (2005) sowie Kolisch und Hartmann (2006).

<sup>27</sup>Vgl. Blazewicz et al. (1983), S. 21

<sup>28</sup>Vgl. Garey und Johnson (1979), S. 109

<sup>29</sup>Vgl. Hromkovič (2014), S. 12

Demeulemeester und Herroelen (2011, S. 31) vertreten die Ansicht, dass für das Projektplanungsproblem die Metaheuristiken die besten computergeschützten Lösungen – zumindest für die frühen Fälle der Project Scheduling Problem Library (PSPLIB) – liefern. Die Vermischung aus Konzepten von unterschiedlichen Metaheuristiken, wie Scatter Search<sup>30</sup> und generische populationsbasierte Evolutionssuche (auch bekannt unter dem Begriff des genetischen Algorithmus),<sup>31</sup> zu einer hybriden Metaheuristik könnte daher erfolgversprechend zum Lösen des RCPSP sein.<sup>32</sup> Aufgabe der Forschung bzgl. des ressourcenbeschränkten Projektplanungsproblems ist es, eine hybride Metaheuristik zu entwickeln, die effizienter arbeitet als die bisher entwickelten Metaheuristiken. Zzt. weichen die heutigen hybriden Metaheuristiken durchschnittlich zwischen 25-30% von der durch die Methode des kritischen Pfads ermittelten Projektfertigstellungsdauer als untere Grenzschränke ab.<sup>32</sup>

Trautmann und Baumann (2009) testeten unterschiedliche Projektplanungssoftwares und kamen zu dem Ergebnis, dass zzt. keines der Softwarepakete mit den aktuell modernsten Heuristiken der neueren Literaturveröffentlichungen wettbewerbsfähig sind.<sup>32</sup> Daher sollte bei der weiteren Entwicklung von Projektplanungssoftware die Aufnahme von modernen hybriden Metaheuristiken vorgenommen werden. Daher ist es zielführend, die hybriden Metaheuristiken für das RCPSP weiterzuentwickeln, damit diese das effizientere Arbeiten der Projektplanungssoftwarepakete für Standardproblemstellungen der Projektplanung gewährleisten.

## 2.3 Eröffnungsverfahren zur Bestimmung der Ablauffolge einer Projektplanung

Durch Auflistung aller Arbeitsgänge nach ihren möglichen Start- bzw. Anfangszeitpunkten  $S_j$  wird eine Ablauffolge bzw. Vorgangsliste  $L = (j_1, j_2, \dots, j_n)$  der für das Projekt benötigten Arbeitsgänge erstellt, wobei diese Liste  $L$  die topologische Reihenfolge der Arbeitsgänge beachten muss.<sup>33</sup> Es handelt sich somit um eine erste zulässige Lösung. Ein Beispiel für eine Ablaufbeziehung eines

---

<sup>30</sup>siehe Kapitel 3.4

<sup>31</sup>siehe Kapitel 3.3

<sup>32</sup>Vgl. Demeulemeester und Herroelen (2011), S. 31-32

<sup>33</sup>Vgl. Brucker und Knust (2006), S. 122-123

Projekts mit zehn Arbeitsgängen liefert Abbildung 1 und eine dazugehörige Ablauffolge des Projekts liefert Abbildung 2.

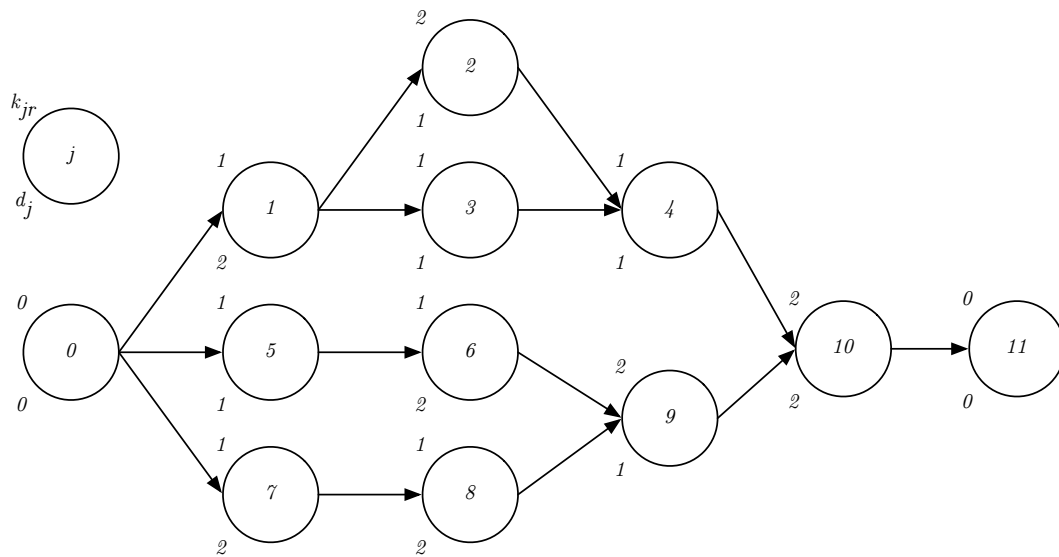


Abbildung 1 Möglicher Aufbau einer Reihenfolgebeziehung eines Projekts

$j$	1	2	3	4	5	6	7	8	9	10	11
$S_j$	0	2	3	4	5	6	8	10	12	13	15

Projektendzeitpunkt ←

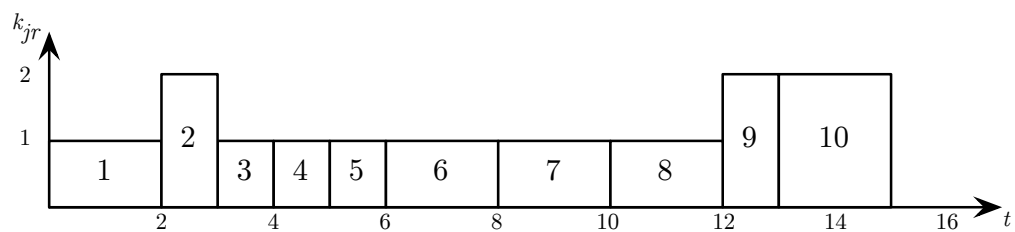


Abbildung 2 Mögliche Ablauffolge eines Projekts

In Anlehnung an: Bouleimen und Lecocq (2003), S. 271

Das aus der engl. Literatur stammende *Schedule Generation Schema (SGS)* liefert gute Ergebnisse für eine nach frühestmöglich sortierten Arbeitsgängen geordnete Ablauffolge eines Projekts.<sup>34</sup>

Das aktivitätsorientierte **serielle SGS** plant bei jeder Iteration einen Arbeitsgang in eine mögliche Arbeitsfolge frühestmöglich ein.<sup>35</sup> Dabei müssen

<sup>34</sup>Vgl. Kolisch und Hartmann (2006), S. 33

<sup>35</sup>Vgl. Brucker und Knust (2006), S. 146-147

die zeitgleich ablaufenden Vorgänge unabhängig voneinander sein, die für den Arbeitsgang notwendigen Ressourcen  $k_{jr}$  müssen vorhanden sein und die notwendigen Vorgänger müssen abgeschlossen sein. Für dieses Schema sind zwei disjunkte Mengen formuliert:<sup>35</sup> Zum einen die Menge an bereits eingeplanten Arbeitsgängen  $\mathcal{V}_n$  zum Zeitpunkt  $t_n$  und zum anderen die Menge an Arbeitsgängen  $\mathcal{D}_n$ , welche zum Zeitpunkt  $t_n$  in die Planung überführt werden sollen. Dabei beschreibt der Parameter  $t_n$  den Zeitpunkt nach  $n$  Iterationen. Anders formuliert bedeutet dies, dass jeder Arbeitsgang  $j$  mit der frühesten Startzeit  $S_j$  in die Ablauffolge  $L$  geplant wird, sofern die Vorgänger  $i \in \mathcal{P}_j$  vom betrachteten Arbeitsgang  $j$  bereits in die Ablauffolge  $L$  eingeplant wurden. Zur Auswahl der möglichen Arbeitsgänge sollte eine Prioritätsregel angewendet werden.<sup>36</sup> Abbildung 3 zeigt den ersten Schritt der Umsortierung des Projekts durch Verschiebung des Arbeitsgangs  $j_7$  zur frühesten Startzeit  $S_j$ . Dabei wird in diesem Fall als Prioritätsregel der Wert der spätesten Fertigstellungszeit  $LFT_j$  verwendet.<sup>36</sup> Es handelt sich bei diesem Schritt um einen left-shift eines Arbeitsgangs  $j$ .<sup>37</sup>

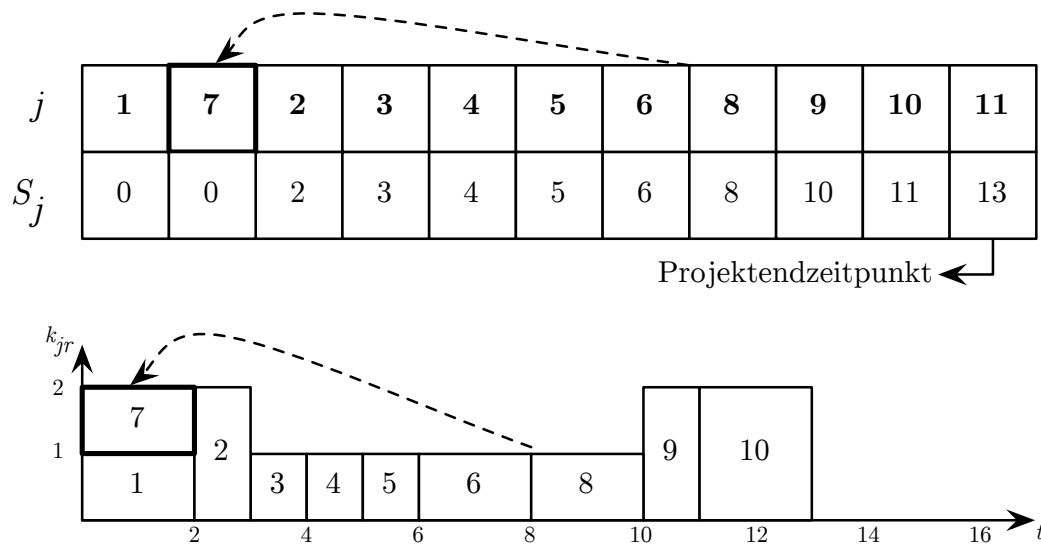


Abbildung 3 Zyklische Umsortierung eines Arbeitsgangs

In Anlehnung an: Bouleimen und Lecocq (2003), S. 271

Nachdem eine Periode verstrichen ist bzw. eine weitere Iteration des SGS erfolgt ist, werden alle erneuerbaren Ressourcen erneuert.<sup>38</sup> Als Ergebnis die-

<sup>36</sup>Vgl. Hartmann (1998), S. 741-742<sup>37</sup>Vgl. Kolisch (1996), S. 325-326<sup>38</sup>Vgl. Bouleimen und Lecocq (2003), S. 271-272

ses iterativen Planungsverfahren ergibt sich ein aktiver Ablaufplan  $S(L)$ .<sup>39</sup> Brucker und Knust (2006) sind der Auffassung, dass für ein RCPSP mit einer regulären Zielfunktionswert eine nach den frühestmöglichen Startzeitpunkten sortierte Ablauffolge eine optimale Lösung darstellt.<sup>40</sup> Dies wird bewiesen durch Formulierung einer Lösung mit Fertigstellungszeit  $FT_j$  eines Projektes sowie der Formulierung der alternativen Lösung mit Fertigstellungszeit  $FT'_j$ , die zusätzlich nach den frühestmöglichen Startzeitpunkten sortiert sind. Dann gilt  $FT'_j \leq FT_j$  für jeden Arbeitsgang  $j$ .<sup>40</sup> Abbildung 4 zeigt eine nach dem seriellen SGS sortierte Ablauffolge eines Projekts.

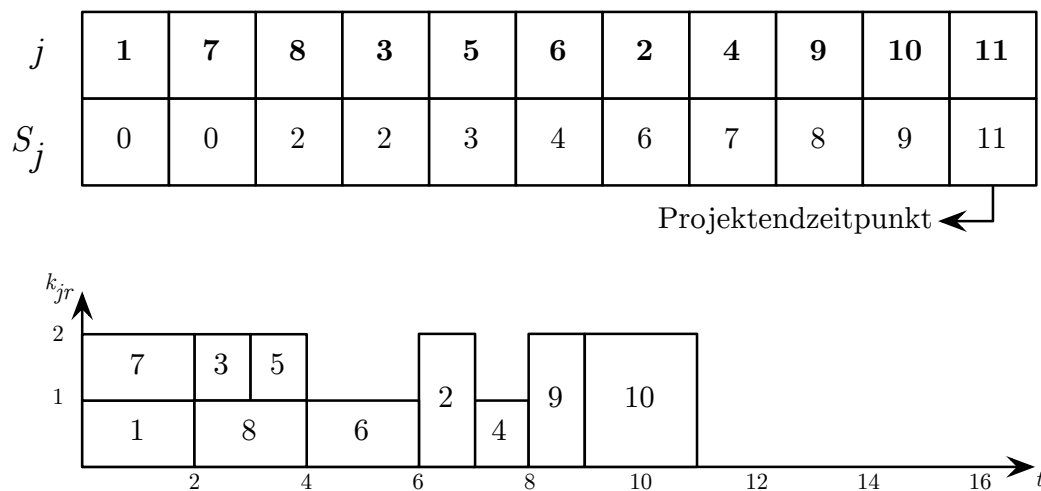


Abbildung 4 Ablauffolge nach seriellen SGS  
**In Anlehnung an:** Bouleimen und Lecocq (2003), S. 271

Bei dem zeit-orientierten **parallelen SGS** wird bei jeder Iteration eine Menge von bisher nicht-geplanten Arbeitsgängen geplant.<sup>41</sup> Eine Iteration  $n$  ist verbunden mit einem Zeitpunkt  $t_n$ , bei dem gilt  $t_m < t_n$  für  $m < n$ . Der Parameter  $m$  steht für die Iteration vor der Iteration  $n$ . Des Weiteren werden drei disjunkte Mengen formuliert.  $\mathcal{D}_n$  steht für die Menge an Arbeitsgängen, die zum Zeitpunkt  $t_n$  zur Entscheidung stehen,  $\mathcal{N}_n$  für die Menge an Arbeitsgängen, die sich zum Zeitpunkt  $t_n$  in Bearbeitung befinden, und der Parameter  $\mathcal{V}_n$  für die Menge an Arbeitsgängen, die bereits bis zum Zeitpunkt  $t_n$  abgeschlossen wurden. Bei jeder Iteration  $n$  wird anhand der Mengen untersucht, welche Arbeitsgänge  $i \in \mathcal{D}_n$  zum Zeitpunkt  $t_n$  mit deren bestimmtem Ressourcenbedarf  $k_{jr}$  verplant werden können. Die Ablauffolge ist generiert bei  $\mathcal{V}_n \cup \mathcal{N}_n$ . Abbil-

<sup>39</sup>Vgl. Kolisch (1996), S. 325-326

<sup>40</sup>Vgl. Brucker und Knust (2006), S. 123-124

<sup>41</sup>Vgl. Tormos und Lova (2003), S. 1073

dung 5 zeigt anhand des eingeführten Beispiels eine durch das parallele SGS generierte Ablauffolge.

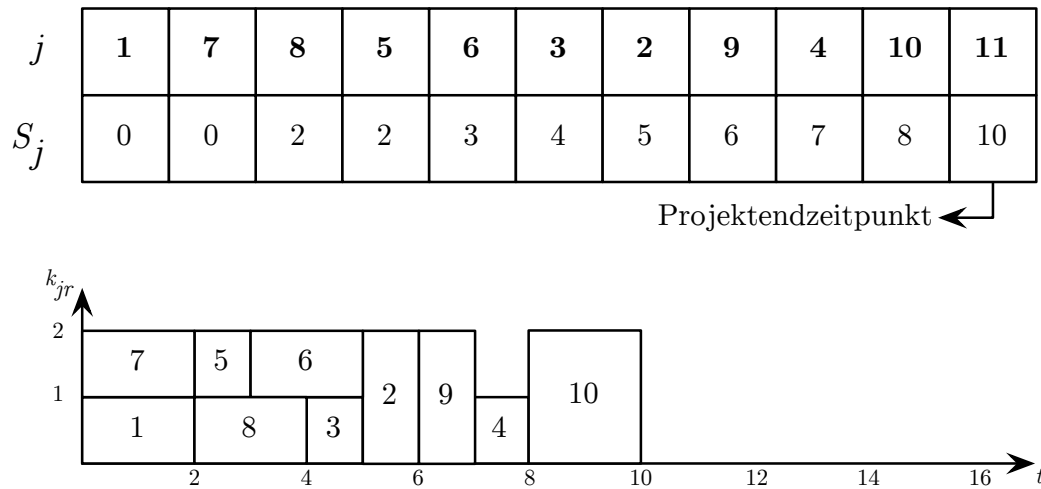


Abbildung 5 Ablauffolge nach parallelem SGS  
**In Anlehnung an:** Bouleimen und Lecocq (2003), S. 271

Das parallele SGS generiert non-delay und das serielle SGS aktive Ablaufplanungen.<sup>42</sup> Bei non-delay Ablaufplanungen geht unter Einhaltung der Vorrangrestriktionen eine frühere Planung eines Arbeitsgangs mit der Verschiebung eines anderen Arbeitsgangs zu einem späteren Startzeitpunkt einher. Bei aktiven Ablaufplänen können Arbeitsgänge nicht durch neuen left-shift früher geplant werden, auch wenn mindestens ein Zwischenplan unzulässig ist. Alle nachfolgenden Heuristiken nutzen zur Bestimmung der Ablauffolge die Verfahren des SGS. Darauf aufbauend erfolgt durch verschiedene Verfahren eine Reoptimierung der Ausgangslösung. Bei der Bewertung der Heuristiken mit Integration des parallelen SGS muss beachtet werden, dass bei einer Untersuchung von Kolisch (1996) die generierten non-delay Ablaufpläne nur 59.7% der optimalen Lösungen enthielten und diese Heuristiken dementsprechend eine verzerrte Lösungsgüte vermitteln könnten.<sup>43</sup>

Zur Entwicklung neuer möglicher Ablaufplanungen anhand der Eröffnungsverfahren gibt es verschiedene Rekombinationsverfahren, bei denen ermittelte Ablauffolgen eines Projekts neu sortiert werden. Im nachfolgenden Kapitel werden verschiedene neu entwickelte Rekombinationsverfahren (Crossover) näher betrachtet und die dazugehörigen Heuristiken aufgeführt.

<sup>42</sup>Vgl. Kolisch (1996), S. 325-326

<sup>43</sup>Vgl. Hartmann (1999), S. 65

## 3 Heuristische Lösungsverfahren für das RCPSP

### 3.1 Verfahren der simulierten Abkühlung

Aufbauend auf einem angepassten Algorithmus zum sogenannten *Simulated Annealing (SA)* wurde eine Heuristik von Bouleimen und Lecocq (2003) entwickelt. Der SA-Algorithmus basiert in seiner ursprünglichen Form auf einem iterativen Verfahren zur Generierung von Nachbarlösungen und der Suche des globalen Minimums. Dabei wird der Lösungsraum durchsucht, indem kontrolliert schlechte Nachbarlösungen vom lokalen Minimum akzeptiert werden. Durch das Durchsuchen des Lösungsraums erfolgt eine Verbesserung des Zielfunktionswerts.<sup>44</sup> Das Verfahren soll für jede Iteration die Menge  $N(y)$  generieren, die eine Menge der Nachbarn  $y'$  vom Punkt  $y$  beschreibt und in jeder Iteration einen Nachbarn  $y'$  von der Nachbarschaft  $N(y)$  vom Punkt  $y$  auswählt. Eine aktuelle Lösung des Punkts  $y$  wird durch den Zielfunktionswert  $f(y)$  charakterisiert und entspricht der Projektfertigungsdauer der generierten Ablauffolge bzw. Lösung  $y$ .

Aufbauend auf den speziellen Eigenschaften haben Bouleimen und Lecocq (2003) den neuen Lösungsansatz der SA entwickelt. Die Heuristik hat als Lösungsansatz die einfache Gestaltung und den schnellen Vergleich der Nachbarschaftslösungen. Bei der Ermittlung der Ablauffolge von  $y_0$  handelt es sich um das in Kapitel 2.3 beschriebene serielle SGS mit der Prioritätsregel der geringsten Bearbeitungsdauer der Arbeitsgänge (SPT-Heuristik). Als untere Schranke wird die Projektfertigstellungsdauer anhand des kritischen Pfads gewählt (CP-Wert).

---

<sup>44</sup>Vgl. Bouleimen und Lecocq (2003), S. 270

**Algorithmus der simulierten Abkühlung****Eingabe:** Projektdaten, CP-Wertberechne  $y_0$  und  $f(y_0)$  mit der SPT-Heuristikspeicher  $y_0 = y_{best}$ ;  $f(y_0) = f_{best}$ ; Ablauffolge von  $y_0 =$  aktuelle Liste, $f(y_0) = f_{aktuelle}$ **Eingabe:** SA-Parameter:  $N_0, h, \Theta_{0max}, \alpha, I, C$ **für**  $C$  Verkettung **tue**     $\Theta = \Theta_{0max}$      $N_n = N_0$     **für**  $I$  Iteration **tue**         $N_n = N_n(1 + h_n)$         **für**  $N_p$  Nachbarn **tue**            erstelle einen Nachbarn  $y'$  von der aktuellen Lösung  $y_{aktuell}$             berechne  $f(y')$             berechne  $\Delta = f(y') - f(y)$             **wenn**  $\Delta < 0$  **dann**                 $y' = y_{aktuell}, f(y') = f_{aktuell}$                 **wenn**  $f(y') < f_{best}$  **dann** speicher  $y' = y_{best}, f(y') = f(y_{best})$ ;                **wenn**  $f_{best} = CP\text{-Wert}$  **dann** beende Prozess;            **sonst wenn**  $P = \exp^{(-\Delta/\Theta)} > y_{zufall}$  **dann** speicher  $y'$  und  $f(y')$  als aktuell;        **Ende**        berechne  $\Theta = \alpha\Theta$     **Ende****Ende****Quelle:** Bouleimen und Lecocq (2003), S. 273

In der Heuristik wird ein neues Abkühlungsschema im Verfahren des simulierten Abkühlens integriert. Es ist lt. den Autoren eine effiziente Suchstrategie für das RCPSP durch Integration von problemabhängigen Parametern. Die Suche nach neuen Lösungen erfolgt anhand multipler Abkühlungsketten und der Neustart des Verfahrens erfolgt bei jedem Zeitpunkt von einer anderen als der ursprünglichen Lösung. Dabei erhöhen sich in jeder Iteration  $n$  die Lösungen bei jeder Prüfung stufenweise um  $N_n = N_n(1 + h_n)$ . Der Parameter  $h$  steht dabei für die Länge der Iteration. Dadurch ist es möglich die Suche weiter auf die Nachbarschaft der guten Lösung auszuweiten.

Für die Nachbarn  $N_p$  wird eine Nachbarlösung  $y'$  der aktuellen Lösung  $y_{aktuell}$  gewählt und anhand des Zielfunktionswerts der aktuellen Lösung und der



Nachbarlösung der Differenzwert  $\Delta$  gebildet. Dieser Wert wird verwendet, um eine bessere Nachbarschaftslösung zu bestimmen und wird wie folgt berechnet:

$$\Delta = f(y') - f(y) \quad (10)$$

Mit der Gleichung (10) kann für das Minimierungsproblem der Wert einer besseren Nachbarschaftslösung bestimmt werden, wenn gilt  $\Delta \leq 0$ , und dementsprechend kann die derzeitige Lösung  $y$  mit der Nachbarlösung  $y'$  ersetzt werden.<sup>45</sup> Damit wird ebenfalls der Zielfunktionswert der Nachbarschaftslösung  $f(y')$  als  $f_{\text{aktuell}}$  gespeichert. Anders ausgedrückt bedeutet dies, dass die ermittelte Nachbarschaftslösung eine bessere Lösung darstellt und dementsprechend ist diese Ablauffolge der Nachbarlösung in der Projektfertigstellungsdauer kürzer.

Sofern die Nachbarlösung eine Ablauffolge mit längerer Projektfertigstellungsdauer hat, wird  $y'$  mit einer Wahrscheinlichkeit von  $P = \exp(-\Delta/\Theta)$  zufällig akzeptiert, wobei dies mit einer zufälligen Lösung  $y_{\text{zufall}} \in [0, 1]$  und dem Verhältnis  $P > y_{\text{zufall}}$  geschieht. Je kleiner der Differenzwert  $\Delta$  ist, desto höher ist die akzeptierte Wahrscheinlichkeit. Der Parameter  $\Theta$  gibt dabei die Temperatur an. Ein höherer Wert der Temperatur gibt eine höhere akzeptierte Wahrscheinlichkeit an. Mit dem Parameter  $\Theta$  wird das spezifische Schema der Abkühlung kontrolliert. Durch die zufällige Auswahl anderer Lösungen und der damit angestoßenen Suche nach Nachbarschaften versucht das Verfahren bessere Lösungen zu generieren. Der Anfangswert der Temperatur ist  $\Theta_0$  und dieser Parameter muss eine ausreichende Größe zur Annahme irgendeiner neuen Nachbarlösung aufweisen.

Nach Festlegung der aktuellen Parameter  $y_{\text{best}}$  und  $f_{\text{best}}$  erfolgt in der nächsten Iteration  $n \in I$  die Reduzierung der Temperatur  $\Theta$  durch den Dämpfungsfaktor  $\alpha$  ( $0 < \alpha < 1$ ) mit der Gleichung  $\Theta_n = \alpha^n \Theta_0$ . Dadurch wird in jeder Iteration bei konstantem Differenzwert  $\Delta$  die Wahrscheinlichkeit der Annahme von  $y_{\text{zufall}}$  reduziert. Damit wird der Effekt des Abkühlens nachempfunden und eine konstante Anzahl von Nachbarlösungen  $N_{\text{sol}}$  erstellt. Der Prozess ist in der Literatur unter dem Begriff der „Abkühlungs- oder Markovketten“ bekannt.

Der Prozess wird beendet, sobald der Zielfunktionswert  $f_{\text{best}}$  den CP-Werts erreicht, da die Ablauffolge den Wert der Projektfertigstellungsdauer des kri-

<sup>45</sup>Vgl. Bouleimen und Lecocq (2003), S. 270

tischen Pfads angenommen hat. Am Ende des Algorithmus wurde die Nachbarschaft der besten Lösung erkundet. Als mögliche Stoppkriterien kann beim SA-Algorithmus lt. den Autoren eine bestimmte Rechenzeit, eine Anzahl an generierten Nachbarn oder auch jede andere Art von objektiven Werten hinterlegt werden. Dabei merken die Autoren an, dass die erlaubte Rechenzeit im Verhältnis zur Größe des Projekts und damit zur Anzahl von Lösungen stehen muss, damit eine gute Lösung generiert wird.

## 3.2 Hybride Heuristikantastung mit selektiver Rückwärts-Vorwärts-Planung

Die hier beschriebene Heuristik nutzt eine *Rückwärts-Vorwärts-Methode* (engl. *Backward-Forward (BF)*) und die Zufallstestprozedur des *Regret Biased Random Sampling (RBRS)*. Der Ansatz der in diesem Kapitel beschriebenen Heuristik sowie die nachfolgenden Aussagen wurden der Arbeit von Tormos und Lova (2003) entnommen, sofern keine anderweitigen Verweise angegeben wurden. Die Heuristik nutzt ebenfalls das SGS zur Ermittlung einer Ablauffolge in einem mehrstufigen Verfahren. Empfohlen wurde der Ansatz eines mehrstufigen SGS von Li und Willis (1992).

Das Verfahren des RBRS kann als eine Zuordnung angesehen werden, welches bei jedem Schritt des SGS für jeden bisher ungeplanten Arbeitsgang  $i \in \mathcal{U}$  eine Wahrscheinlichkeit  $\psi(j)$  wählt ( $\psi: j \in \mathcal{U} \rightarrow [0..1]$ ;  $\sum_{j \in \mathcal{U}} \psi(j) = 1$ ). D. h. bei der Zufallstestprozedur wird jedem nicht-erledigten Arbeitsgang eine Wahrscheinlichkeit  $\psi(j)$  zugeteilt, die besagt, mit welcher Wahrscheinlichkeit der betrachtete Arbeitsgang ausgewählt wird. Die ermittelte Reihenfolge der Vorgänge unterscheidet sich bei jedem Durchgang, aber die beste Lösung wird am Ende der Heuristik generiert. Der parametrisch zugeordnete Wahrscheinlichkeitswert  $\psi(j)$  wird berechnet durch:

$$\psi(j) = \frac{(\rho_j + 1)^\beta}{\sum_{i \in \mathcal{U}} (\rho_i + 1)^\beta} \quad (11)$$

Der Parameter  $\beta$  beschreibt das Ausmaß der Verzerrung. Bei  $\beta = 0$  besteht eine Verzerrung des Wahrscheinlichkeitswerts  $\psi(j)$  und es wird eine zufällige

Auswahl der möglichen Arbeitsgänge durchgeführt.

$$\rho_j = \max_{i \in \mathcal{U}} v(i) - v(j) \quad (12)$$

Die Gleichung (12) bestimmt den Wert des Bedauerns  $\rho_j$  für jeden Arbeitsgang in  $\mathcal{U}$ , indem der Prioritätswert  $v(j)$  für den betrachteten Arbeitsgang  $j$  vom schlechtesten Prioritätswert von allen Arbeitsgängen aus  $\mathcal{U}$  abgezogen wird. Weiter wird der Gleichung (11) beim Parameter  $\rho_j$  die Konstante 1 hinzuge-rechnet. Damit ist immer die Wahrscheinlichkeit eines Arbeitsgangs  $j$  aus  $\mathcal{U}$  größer als 0 und dementsprechend bleibt dieser in jeder Planung der Population wählbar.

### Hybride Heuristikantastung + selektive BF-Methode

**Eingabe:** Lese *Projektdaten*, *#Plangrenze*, *Zeitgrenze*

**Beginn**

```

    Startzeit = Zeit(), berechne  $LFT_j \forall j$ ,  $Plan = 0$ ,  $Sum\_FT_J = 0$ ,
    #RBRS = 0
    solange ( $\#Plangrenze > Plan$ ) und ( $(Zeit() - Startzeit) < Zeitgrenze$ ) tue
         $FT_J = RBRS(SGS, MLFT)$ 
         $Sum\_FT_J = Sum\_FT_J + FT_J$ 
         $\#RBRS = \#RBRS + 1$ 
         $Plan = Plan + 1$ 
        wenn ( $FT_J \leq (Sum\_FT_J / \#RBRS)$ ) dann
             $FT'_J = BF - Durchlauf$ 
             $Plan = Plan + 1$ ;
            wenn ( $FT'_J < FT_J$ ) dann
                 $BF - Durchlauf$ 
                 $Plan = Plan + 1$ ;
```

**Ende**

**Ende**

**Ausgabe:** Finale Planung:  $(S_{best0}, S_{best1}, \dots, S_{bestn+1})$

**Quelle:** Tormos und Lova (2003)

Aufbauend auf der Eigenschaft des RBRS konstruierten Tormos und Lova (2003) den Algorithmus der hybriden Heuristikauswahl inkl. einer selektiven Rückwärts-Vorwärts-Planung.

Der Algorithmus generiert mögliche Lösungen bis dieser entweder eine Begrenzung der CPU-Zeit (*Zeitgrenze*) oder die Menge an Planungserstellung

( $\#Plan$ grenze) erreicht hat. Als Prioritätsregel zum Ziel der Minimierung der Projektdauer wird der minimal-späteste Fertigstellungszeit (engl. minimum latest finish time (MLFT)) verwendet.<sup>46</sup> Die Vorteile dieser Prioritätsregel liegen lt. Tormos und Lova (2003) einerseits an dem niedrigen Rechenaufwand und andererseits an dem über die Anzahl der Iterationen konstant bleibenden Wert.

Zu Beginn des Algorithmus wird die Startzeit festgelegt sowie der Iterationszähler ( $Plan$ ) für die Planung und die kumulierte Summe der Fertigstellungszeit ( $Sum\_FT_j$ ) auf den Wert 0 gesetzt. Des Weiteren wird für jeden Arbeitsgang  $j$  der späteste Fertigstellungszeit ( $LFT_j$ ) berechnet. Dieser wird als jeweiliger Prioritätswert für das RBRS verwendet.

Im Anschluss erfolgt die Schleife zur Ermittlung einer Planung unter Abhängigkeit der Abbruchkriterien. Für eine Iteration wird der Fertigstellungszeitpunkt  $FT_j$  des Projekts mit Hilfe des RBRS durch Anwendung beider Verfahren des SGS mit der  $MLFT$ -Prioritätsregel und dem parametrisch zugeordneten Wahrscheinlichkeitswert  $\psi(i)$  mit einem  $\beta$ -Satz bis 1 berechnet.<sup>46</sup> Das SGS erfolgt als mehrstufiges Verfahren, bei dem eine Hälfte der Planungen durch das serielle Schema und die andere Hälfte durch das parallele Schema geplant wird.<sup>47</sup> Bei dem SGS wird ein Arbeitsgang  $j$  bei jeder Iteration zu einer geordneten Ablauffolge  $L$  mit den geplanten Fertigstellungszeitpunkten  $FT_j$  und den Startzeitpunkten  $S_j$  überführt. Anhand der Ablauffolge  $L$  lässt sich die Projektfertigstellungszeit  $FT_j$  entnehmen.<sup>48</sup> Anschließend wird bei der Iteration der ermittelte Fertigstellungszeitpunkt  $FT_j$  der Summe aller Fertigstellungszeitpunkte ( $Sum\_F_{n+1}$ ) hinzugefügt und der Zähler für die Anzahl an Planungen ( $Plan$ ) und Durchführungen des RBRS ( $\#RBRS$ ) um 1 erweitert.

Im nächsten Teil der Schleife wird geprüft, ob der durch das RBRS berechnete Fertigstellungszeitpunkt  $FT_j$  kleiner gleich der durchschnittlichen Fertigstellungszeit ( $Sum\_FT_j/\#RBRS$ ) ist. Sofern dies zutrifft, wird der Fertigstellungszeitpunkt der Planung  $FT'_j$  durch einen  $BF$ -Durchlauf generiert. Dabei durchläuft die Rückwärtsplanung, die durch den ersten Schritt der Schleife generierte Ablauffolge und erstellt eine Teilplanung für jeden Arbeitsgang  $j$ . Bei der Teilplanung wird der Arbeitsgang  $j$  im Intervall  $[S_j, \min_{(i,j) \in \mathcal{A}} S'_i]$  zum

<sup>46</sup>Vgl. Kolisch und Drexel, 1996, S. 27-29; Schirmer (1997)

<sup>47</sup>Vgl. Tormos und Lova (2001), S.68

<sup>48</sup>Vgl. Kapitel 2.3

spätesten Startzeitpunkt unter Einhaltung der Vorrangs- und Ressourcenrestriktionen umsortiert. Dabei beschreibt die Menge  $\mathcal{A}$  die Menge an Vorrangsbeziehungen und  $S'_i$  die geplante Startzeit des Arbeitsgangs  $i$  in der rückwärtsgerichteten Teilplanung. Aufbauend darauf, erfolgt im Weiteren die Vorwärtsplanung, die anhand der neuen Ablauffolge der Rückwärtsplanung die Arbeitsgänge  $j$  in aufsteigender Sortierungsreihenfolge umsortiert. Die Arbeitsgänge  $j$  werden in Abhängigkeit der Ressourcenkapazität zum frühesten Startzeitpunkt im Intervall  $[\max_{(i,j) \in \mathcal{A}} FT'_i, FT_j]$  umgeplant. Dabei ist  $FT'_i$  die Fertigstellungszeit des Arbeitsgangs  $i$  in der vorwärtsgerichteten Teilplanung. Mit Abschluss dieser Planung ist der Fertigstellungszeitpunkt  $FT'_j$  durch die BF-Methode berechnet.

Falls der Fertigstellungszeitpunkt  $FT'_j$  kleiner als der bisherige Fertigstellungszeitpunkt  $FT_j$  ist, wird ein weiterer BF-Durchlauf durchgeführt und im Anschluss der Zähler der Planungen um 1 erweitert. Nach Ende der Prozedur kann die beste Ablauffolge (Output) mit den besten Startzeitpunkten für alle Arbeitsgänge beschrieben werden.

### 3.3 Evolutionäre Lokalsuche aufgrund Variablennachbarschaft

Bei dem Verfahren der Evolutionären Lokalsuche aufgrund Variablennachbarschaft handelt es sich um einen genetischen Algorithmus, bei dem in jedem Durchlauf eine Population an Individuen bzw. lokalen Optima generiert wird. Der Algorithmus wurde der Arbeit von Kochetov und Stolyar (2003) entnommen.

Genetische Algorithmen (GA) gehören zu den evolutionären Algorithmen. Es sind populations-basierte Metaheuristiken, die nicht nur mit einer einzelnen Lösung arbeiten, sondern mit einer Menge von Lösungen,<sup>49</sup>. Der Ansatz geht auf die Arbeit von Holland (1975) zurück. Da es sich hier ebenfalls um einen heuristischen Lösungsansatz handelt, sind die ermittelten Lösungen eines genetischen Algorithmus meist ebenfalls nur eine gute Lösung des betrachteten Problems in einer akzeptablen Rechenzeit.

---

<sup>49</sup>Vgl. Talbi (2009), S. 25

Ein Individuum repräsentiert eine zulässige Lösung in dem Verfahren der GA<sup>50</sup> und solch ein Individuum wird einem sogenannten Chromosom gleichgesetzt<sup>51</sup>, das üblicherweise als Zeichen- bzw. Zahlenkette dargestellt wird.<sup>52</sup> Damit kann ein Individuum beim Projektplanungsproblem als Ablauffolge angesehen werden, welche aus einer Sequenz aller Arbeitsgänge eines Projekts in deren topologischer Reihenfolge bestehen.<sup>53</sup> Eine Gruppe von Lösungen bzw. Individuen ist eine Population.<sup>54</sup> Für das Projektplanungsproblem sollte das Individuum bzw. das Chromosom den eindeutigen Ablaufplan eines Projektes repräsentieren.<sup>55</sup>

Das Verfahren nutzt ebenfalls die Erstellung einer Ablauffolge zum Lösen des RCPSP. Es wird über das SGS für eine beliebige Liste  $L$  eine aktive Planung erstellt, die eine optimale Planung lt. Kapitel 2.3 darstellt. Die Autoren definierten zur Lösung eines RCPSP durch ihre Heuristik einen  $Block_j$  des Arbeitsgangs  $j$ , der einer Menge von Arbeitsgängen entspricht, die zeitgleich mit dem Arbeitsgang  $j$  geplant werden können. Der  $Block_j := \emptyset$ , wenn er mindestens einen Vorgänger  $i$  vom Arbeitsgang  $j$  enthält.

$$Block_j = \{i \in J | S_i + d_i \in [S_j, S_j + d_j]\} \cup \{i \in J | S_i \in [S_j, S_j + d_j]\} \quad (13)$$

Die Ermittlung der Ablauffolge  $L'$  der Nachbarschaft  $N_1(L)$  der ursprünglichen Ablauffolge  $L$  für jeden Arbeitsgang  $j$  eines nichtleeren Blocks erfolgt in vier Teilschritten. Dafür wird ein sogenanntes Ergebnisnetzwerk genutzt. Lt. den Autoren ist das Ergebnisnetzwerk eines Arbeitsgangs  $j$  das Maximum infolge der Integration eines Teilgraphen in den Richtungsgraph  $G_S$ , welches ein Netzwerk verbunden mit dem Arbeitsgang  $j$  ist. Dabei ist der Richtungsgraph  $G_S$  für eine Planung definiert als  $G_S(V, H)$ , mit  $V = H$  und  $H = \{(i, j) | S_i + p_i = S_j, i \in \mathcal{P}_j\}$ .

Im ersten Schritt werden für eine Liste  $L$  die Positionen *First* und *Last* der Arbeitsgänge ermittelt. Bei dem Wert *First* handelt es sich um die frühestmögliche Position in einem  $Block_j$  des jeweiligen Arbeitsgangs  $j$  und bei dem Wert *Last* um die spätestmögliche Position der zwei Werte  $Last_1$  und  $Last_2$  eines

---

<sup>50</sup>Vgl. Mitchell (1998), S. 6

<sup>51</sup>Vgl. Nissen (1997), S. 13

<sup>52</sup>Vgl. Gerdes et al. (2004), S. 37

<sup>53</sup>Vgl. Hartmann (1998), S. 735

<sup>54</sup>Vgl. Koch (2014), S. 60

<sup>55</sup>Vgl. Kellenbrink (2014a), S. 77

Arbeitsgangs  $j$  der gesamten Arbeitsgänge im  $Block_j$ . Im zweiten Schritt wird für den Arbeitsgang  $First - 1$  über eine Teilplanung die frühestmögliche Position in der Liste  $L$  anhand des seriellen SGS ermittelt. Im nächsten Schritt wird die Teilplanung durch Planung der nächsten Arbeitsgänge  $Last - First + 1$  der Liste  $L$  via parallelem SGS erweitert. Aufgrund dieser Vorgehensweise ist zu einem jeden Zeitpunkt  $t$  der Projektplanung eine entsprechende Menge  $\mathcal{L}_t$  vorhanden, die die Menge an möglichen Arbeitsgängen, die zu diesem Zeitpunkt  $t$  geplant werden können, beschreibt. Die Autoren formulieren zur Bestimmung der Teilmenge der Arbeitsgänge  $\mathcal{L}_t$ , welche in einer Iteration geplant werden sollen, ein multi-dimensionales Knapsack-Problem.

### Formulierung eines multi-dimensionales Knapsack-Problems<sup>56</sup>

$$\max \sum_{j \in \mathcal{L}_t} x_j \frac{1}{\mathcal{R}} \sum_{r=1}^{\mathcal{R}} \frac{k_{jr}}{K_r} \quad (14)$$

unter Beachtung der Restriktionen

$$\sum_{j \in \mathcal{L}_t} x_j k_{jr} \leq K_r - \sum_{j \in S_t} k_{jr} \quad r \in \mathcal{R} \quad (15)$$

$$x_j \in \{0, 1\} \quad (16)$$

Dabei beschreibt  $\mathcal{R}$  die Menge an erneuerbaren Ressourcen. Das multidimensionale Knapsack-Problem hat als Zielfunktion die Maximierung der Ressourcenverwendung bei jeder Teilplanung der Arbeitsgänge aus der Menge  $\mathcal{L}_t$ . Dabei muss die Nebenbedingung der Einhaltung der verbleibenden Kapazitäten des  $r$ -ten Ressourcentyps bei dem Zeitpunkt  $t$  berücksichtigt werden. Zum Lösen des Problems verwenden die Autoren das Verfahren der Greedy Randomized Adaptive Search Procedures (GRASP) von Feo und Resende (1995). Dadurch ist es möglich, zum Zeitpunkt  $t$  jeden Arbeitsgang der Teilmenge in die Teilplanung zu integrieren.

Bei der allgemeinen Formulierung wird in jeder Iteration des GRASP eine Konstruktionsphase und eine Phase der lokalen Suche durchgeführt.<sup>57</sup> In der Konstruktionsphase werden durch Anwenden der Randomisierung in mehre-

<sup>56</sup>Vgl. Valls et al. (2004), S. 310

<sup>57</sup>Vgl. Resende und Ribeiro (2014), S. 6; Feo und Resende (1989), S. 68-70

ren Durchläufen unterschiedliche Lösungen erstellt. Anhand einer Lösung wird eine Kandidatenmenge bestimmt und deren Zuwachskosten bewertet. Solange diese Kandidatenmenge Elemente enthält, wird eine Liste mit Elementen mit den geringsten Zuwachskosten erstellt. Von dieser Liste wird dann ein zufälliger Wert gewählt und in der Lösung berücksichtigt. Anschließend wird die Kandidatenmenge aktualisiert und die Zuwachskosten der Kandidaten aus dieser Menge neu bewertet. Nach Abschluss der Prozedur ist eine Lösung konstruiert. Sofern die Lösung möglich ist, wird die Nachbarschaft erforscht bis ein lokales Minimum durch die Phase der lokalen Suche gefunden wurde. Falls diese Lösung nicht möglich ist, sieht der Algorithmus entweder die Möglichkeit der Verwerfung der erstellten Lösung vor oder durch die Integration einer Reparaturheuristik die Erhaltung dieser Lösung. Nach Ende aller Iterationen kann die beste Lösung wiedergegeben werden. Abschließend sollte durch das Verfahren des GRASP das multi-dimensionale Knapsack-Problem gelöst sein.

Im letzten Schritt wird die neue Ablauffolge  $L'$  konstruiert. Dies erfolgt durch Setzen von  $L' = L$  in der ersten  $First - 1$  Position und dadurch, dass die nächsten  $Last - First + 1$  Arbeitsgänge mit ihrer Startzeit in der Liste  $L'$  in nicht-abnehmender Reihenfolge in die Teilplanung integriert werden. Anschließend wird anhand der Ablauffolge der Liste  $L$  die Ablauffolge der Liste  $L'$  vervollständigt. Für die weitere Nachbarschaft  $N_2(L)$  erfolgt dieses Verfahren analog. Dafür wird die aktuelle aktive Planung in eine rechte aktive Planung umgeändert.

Weiter nutzt dieses Verfahren die Randomisierung der Nachbarschaften. Damit werden die Versuche je Iteration reduziert. D. h., es wird jedem Element einer Nachbarschaft eine Wahrscheinlichkeit  $P \leq 1$  hinterlegt. Anschließend wird eine Tabu-Liste generiert. Mit der Summe aller Startzeitpunkte der Arbeitsgänge  $j$  einer Lösung  $L$  wird der Tabu-Status  $TS(L)$  für eine Planung berechnet und damit die Länge einer Planung  $S(L)$ .

$$TS(L) = \sum_{j=1}^n S_j \quad (17)$$

Das Verfahren nutzt als Crossover Operator die Strategie der *Pfadneuverknüpfung* (PNV) von Glover und Laguna (1999). Mit der Strategie werden die Verbindungen von guten Lösungen erkundet.<sup>58</sup> Dafür wird der Parameter

---

<sup>58</sup>Vgl. Glover et al. (2000), S. 7-13



$Shift_{ij}(L)$  eingeführt. Die Autoren nutzen diesen Parameter, um die Bewegung des Arbeitsgangs  $i$  aufzuzeigen, gleich nachdem Arbeitsgang  $j$  gemeinsam mit allen Nachfolgern in die Liste  $L$  geplant wird. Dabei wird die Vorrangsbeziehung durch die Menge von Paaren  $C = \{(i, j)\}$  beschrieben, die für den Arbeitsgang  $j$  jeweils die Vorgänger  $i$  aufzeigt.

$Shift_{ij}(L) :$

$$L = (...i...t_1...t_m...j...) \rightarrow L' = (...jit_1...t_m...), (i, t_1), ..., (i, t_m) \in C \quad (18)$$

Wenn  $L$  und  $L'$  zwei Lösungen sind, dann lässt sich ein Pfad  $L = L_0, ..., L_k = L', L_i = Shift_{pq}(L_{i-1})$  konstruieren. Der Parameter  $A(j)$  beschreibt die Menge von Vorgängen bevor der Arbeitsgang  $j$  in die Ablauffolge  $L$  aufgenommen wurde. Analog bildet dies die Menge  $A'(j)$  der Ablauffolge  $L'$ . Die Reihenfolgerestriktion für einen Arbeitsgang  $j$  ist dann erfüllt, wenn Folglich gilt:  $A'(j) \subseteq A(j)$ . Die Pfadkonstruktion transformiert die Arbeitsgänge der Menge  $J$  der Liste  $L$  nacheinander in eine neue Liste  $L'$ , in Abhängigkeit ihrer Mächtigkeit und in abnehmender Sortierungsreihenfolge.

Die Ermittlung von neuen Regionen des möglichen Lösungsraums durch Erstellung von neuen Populationen soll das Ende der Evolution sein. Es sollte eine möglichst entfernte neue Population bestimmt werden. Die Autoren haben dafür ein Traveling-Salesman-Problem mit einer Vorrangsbeziehung konstruiert.

$$\min \sum \omega_{j_r j_{r+1}} \quad (19)$$

unter Beachtung der Restriktionen

$$\mathcal{P}_{j_{r+1}} \subseteq \{j_1 \dots j_r\} \quad (20)$$

Mit der Zielfunktion wird die Entfernung zwischen der neuen Lösung und der Lösung der alten Population minimiert. Der Parameter  $\omega_{ij}$  ist dabei eine Anzahl von Lösungen einer aktuellen Population, bei dem der Arbeitsgang  $j$  in die Liste aufgenommen ist, gleich nachdem der Arbeitsgang  $i$  gelistet wird. Auch hier wird der GRASP-Algorithmus zum Lösen des Problems genutzt.

### 3.4 Hybride elektromagnetische Metaheuristik

Die nachfolgenden Erklärungen zu den Heuristiken sind der Arbeit von Debels et al. (2006) entnommen (dementsprechend beruhen sämtliche Aussagen auf dieser Arbeit, sofern keine anderen Verweise angegeben wurden). Die Heuristik bietet lt. den Autoren eine gute Annäherung an die optimale Lösung für relativ große Fälle des ressourcenbeschränkten Projektplanungsproblems. Das Verfahren kombiniert den *Scatter Search (SS)*, bei dem es sich um einen Sonderfall des populations-basierten, evolutionären Algorithmus handelt sowie eine neu entwickelte heuristische Methode der Optimierung von ungekoppelten kontinuierlichen Funktionen, basierend auf einer Analogie der Elektromagnetismustheorie.

#### Pseudo-Code des Scatter Search (SS)

1. erstelle Pool  $P$  willkürlich erstellter Lösungen
2. erstelle  $RefSet = \mathcal{B}_1 \cup \mathcal{B}_2, \mathcal{B}_1 \cup \mathcal{B}_2 \subset \mathcal{E}, \mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$
- solange**  $NrPlanung < Planungslimit$  **tue**
  3. generiere Teilmenge von  $RefSet$
  4. erstelle neuen Pool  $\mathcal{E}$  von Lösungen unter Verwendung von Lösungskombinationsmethode für jede Untermenge
  5. aktualisiere  $RefSet$

**Ende**

**Quelle:** Debels et al. (2006)

Der SS gehört zur Klasse der evolutionären Algorithmen und zur Unterkategorie der GA. Bei dieser Heuristik werden Lösungen kombiniert, die zur Menge der Lösungen eine konvexe oder nicht-konvexe Kombination aufweisen. Durch das Vereinigungsprinzip zur Erweiterung der Menge der Lösungen durch eine verallgemeinerte Konstruktion des Projektverlaufs im euklidischen Raum und durch Nutzen von Strategiedesigns betreffend der Umsortierung der Lösungsansätze der Populationssortierung (Randomisierung) werden neue gute Lösungen generiert.

Der generierte Lösungsvektor aus dem ersten Schritt vom pseudo SS-Algorithmus enthält willkürlich generierte Lösungen. Die Lösungen haben das Format der standardisierten Zufallszahl. Es handelt sich hier um ein Prioritätswertverfahren, bei dem jedem Arbeitsgang einer möglichen Lösung  $y_i$  ein Prioritätswert zugeordnet ist.<sup>59</sup> Im nächsten Schritt werden qualitativ hochwertige

<sup>59</sup>Vgl. Domschke und Drexl (2005), S. 129

Lösungen (Teilmenge  $\mathcal{B}_1$ ) und andere Lösungen (Teilmenge  $\mathcal{B}_2$ ) der Menge  $\mathcal{E}$  in die Referenzmenge *RefSet* überführt. *RefSet* hat die Eigenschaft  $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$ . Anhand der Unterschiedlichkeiten der willkürlichen Lösungen aus  $\mathcal{E}$  werden anhand der besten Bearbeitungsdauer und der minimalen Distanz zwischen den Elementen die Lösungen der Teilmenge  $\mathcal{B}_1$  bestimmt.  $\mathcal{B}_2$  sind Lösungen mit der weitesten Distanz zu der jeweiligen Bearbeitungsdauer der Lösungen aus  $\mathcal{B}_1$ . Im nächsten Schritt werden mehrere Reoptimierungen der Referenzmenge *RefSet* durchgeführt, solange das Stoppkriterium des Planungslimits nicht erreicht ist. Bei der Reoptimierung wird von der Referenzmenge *RefSet* wiederum eine Teilmenge gebildet und anhand dieser ein Pool an Lösungen generiert. Bei der Generierung der Lösungen wird eine Lösungskombinationsmethode für jede Untermenge verwendet. Dies kann in zwei unterschiedlichen Verfahren erfolgen:

1. Paare in  $\mathcal{B}_1$ : Es werden alle bisherigen Paare von  $\mathcal{B}_1$  betrachtet, die eine relativ gute Lösung aufweisen und somit eine relativ kurze Projektfertigstellungsdauer. Von jedem solcher Paare werden Abkömmlinge produziert und in den neuen Pool überführt. Dies erfolgt mit dem Instrument eines zweistufigen Crossovers. D. h. es wird anhand von zwei Eingangslösungen (Eltern) durch Rekombination eine neue Ausgangslösung (Abkömmling) generiert. Durch Auswahl der anfänglichen Arbeitsgangfolge der ersten Eingangslösung und anschließender Vervollständigung anhand der Abfolge der zweiten Eingangslösung wird für die neue Ausgangslösung die Reihenfolge der Arbeitsgänge unter Einhaltung der Vorrangstrektionen generiert.<sup>60</sup>

2. Elemente von  $\mathcal{B}_1 \times \mathcal{B}_2$ : Von jeder Kombination von jeweils einem Element aus  $\mathcal{B}_1$  und  $\mathcal{B}_2$  wird über die elektromagnetische Metaheuristik die Ablauffolge einer Lösung  $y$  verbessert. Die Kraft eines Punkt  $y$  aus  $\mathcal{B}_1$  zieht einen Punkt  $y$  aus  $\mathcal{B}_2$  in seine Richtung an. Für die Paare von  $\mathcal{B}_1 \times \mathcal{B}_1$  kann dieses Verfahren nicht angewendet werden. Dies liegt an der simultanen Bewegungsrichtung der korrespondierenden Planung. Ursächlich liegt es an der Verringerung der Wirksamkeit des EM-Algorithmus, da die Verbesserung basierend auf Unterschieden des Zielfunktionswertes zur Orientierung der Suche resultiert.<sup>61</sup> Abbildung 6 im Anhang versucht die Lösungskombinationsmethoden zu veranschaulichen. Die genauere Vorgehensweise des EM-Algorithmus wird im Folgenden ausführlich dargestellt.

---

<sup>60</sup>Vgl. Hartmann (1998), S. 737

<sup>61</sup>Vgl. Debels et al. (2006), S. 644-645

**Pseudo-Code für die elektromagnetische Metaheuristik (EM)****solange** *Stoppkriterium nicht erfüllt* **tue**

- 1. Lokalsuche
- 2. Kräfte berechnen
- 3. Kräfte verwenden

**Ende****Quelle:** Birbil und Fang (2003)

Die von Birbil und Fang (2003) entwickelte elektromagnetische Metaheuristik löst Probleme der ungekoppelten globalen Optimierung. Jeder Punkt in einem multidimensionalen Lösungsraum repräsentiert eine Lösung und jede Lösung hat eine bestimmte Ladung, die wiederum mit einem Zielfunktionswert verbunden ist. Jeder Lösungspunkt übt eine Anziehungs- oder Anstoßwirkung auf einen anderen Punkt aus. Die Größe ist proportional zu dem Produkt der Ladungen und umgekehrt proportional zu dem Abstand zwischen den Punkten (Coulombsches Gesetz). Das Prinzip des Algorithmus baut auf dieser Eigenschaft auf.

Ein schlechter Lösungspunkt übt eine Bewegung in die Richtung aus, die durch den Anstoß anderer Punkte der Population verhindert wird, und die Bewegung zu den attraktiveren Lösungen wird dadurch vereinfacht. Es kann somit von einer Lokalsuche im euklidischen Raum in einer populations-basierten Sichtweise gesprochen werden. Der Hauptunterschied besteht lt. Debels et al. (2006) darin, dass die Bewegung der Kräfte durch die Gesetze des Elektromagnetismus geregelt werden.

Im ersten Schritt des EM-Algorithmus werden die direkten Nachbarlösungen der individuellen Punkte der Population gesucht. Die ausgeübte Gesamtkraft auf jeden Punkt von allen anderen Punkten wird im zweiten Schritt berechnet. Diese Kraft ist abhängig von der Ladung des betrachteten Punkts unter Berücksichtigung der ausgeübten Kraft der anderen Punkte. Die Ladung von jedem Punkt  $y_i$  ist bestimmt durch den Zielfunktionswert  $f(y_i)$  in Relation zu dem Zielfunktionswert des aktuell besten Punkts  $y^{best}$  der Population. Die Ladung  $q_i$  vom Punkt  $y_i$  ist gegeben durch die Gleichung (21).

$$q_i = \exp \left( -\delta \frac{f(y_i) - f(y^{best})}{\sum_{k=1}^w (f(y_k) - f(y^{best}))} \right) \quad (21)$$

Die Populationsgröße wird durch den Parameter  $w$  gegeben und der Parameter

$\delta$  ist die Dimension des Lösungsraums.  $KA_i$  von  $i = 1, \dots, w$  repräsentiert den Vektor der Kraftausprägungen von einem Punkt  $y_i$ . Formal wird  $KA_i$  definiert durch die Gleichung (22):

$$KA_i = \sum_{j=1, j \neq i}^{j=w} \left\{ \begin{array}{l} (y_j - y_i) \left( \frac{q_i q_j}{\|y_j - y_i\|^2} \right) \text{ wenn } f(y_j) < f(y_i) \\ (y_i - y_j) \left( \frac{q_i q_j}{\|y_j - y_i\|^2} \right) \text{ wenn } f(y_j) \geq f(y_i) \end{array} \right\} \quad (22)$$

Sobald ein Punkt einen relativ guten Zielfunktionswert aufweist, zieht er einen anderen Punkt an. Punkte mit relativ minderwertigen Zielfunktionswerten stoßen die anderen Populationsmitglieder ab. Die auf den Punkt  $i$  ausgeübte Kraft von allen anderen Punkten wird mittels Vektoraddition vollzogen. Durch die Übereinstimmung der Bewegung aufgrund der ausgeübten Kraft wird eine neue Population von Lösungen erzeugt.

Weiteres Vorgehen von Debels et al. (2006) war, den EM-Algorithmus für das RCPSP umzuformen. Dazu wurde ein Konzept verallgemeinert durch Berücksichtigung von einer vorbestimmten Anzahl  $z \in [1; w - 1]$  von Punkten zum Darstellen eines beliebig gegebenen Punktes, wobei  $w$  weiterhin die Populationsgröße beschreibt. Experimente haben gezeigt, dass  $z = 1$  gute Ergebnisse liefert und leicht zu implementieren ist.<sup>62</sup>

Für die Kraftübertragung wird anstelle  $q_i$  und  $q_j$  für das Ausmaß der Ladung der Parameter  $q_{ij}$  eingeführt. Dieser Parameter repräsentiert die Ladung als Relation der Entfernung des Zielfunktionswert zwischen  $i$  und  $j$ .

$$q_{ij} = \frac{f(y_i) - f(y_j)}{f(y^{worst}) - f(y^{best})} \quad (23)$$

Die Berechnung der Ladung  $q_{ij} \in [-1; 1]$  eines Punktes durch Gleichung (23) kann nicht mehr unabhängig erfolgen, aber auf Basis der Punkte kann die ausgeübte Kraft durch  $y^{worst}$  und  $y^{best}$  berechnet werden. Damit können schlechte und bessere Lösung im *RefSet* ermittelt werden, da die besseren Lösungen  $y_j$  als Ergebnis einen höheren Wert  $q_{ij}$  haben. Wenn  $f(y_i) > f(y_j)$ , dann ist  $q_{ij}$  positiv und die bessere Lösung  $y_j$  zieht die schlechtere Lösung  $y_i$  an, da der Punkt  $y_i$  eine längere Bearbeitungsdauer als der Punkt  $y_j$  hat.

$$KA_{ij} = (y_j - y_i) \cdot q_{ij} \quad (24)$$

<sup>62</sup>Vgl. Debels et al. (2006), S. 644-645

Die Gleichung (24) wurde in der Arbeit von Debels et al. (2006) für die Kraftübertragung von Punkt  $y_j$  auf Punkt  $y_i$  verwendet und ist analog zu der PNV von Glover et al. (2000). Da der Parameter  $q_{ij}$  in der Gleichung (24) beinahe immer kleiner als 1 ist, wandert die Lösung  $y_i$  zur Lösung  $y_i + KA_{ij}$  und somit in Richtung  $y_j$ . Aufgrund dieser Eigenschaft wird eine neue Lösung generiert.<sup>63</sup>

Weiter muss zur Integration des EM-Algorithmus in eine Heuristik zum Lösen des RCPSP die Eigenschaft zur Änderung der Prioritätswerte der Arbeitsgänge einer möglichen Lösung  $y_i$  integriert sein.<sup>63</sup> Damit wird garantiert, dass jede Lösung einer einzigartigen Planung entspricht.<sup>64</sup> Dies geschieht dadurch, dass der Kraft ermöglicht wird auf eine einzelne Teilmenge der Dimension zu wirken. Ein Punkt  $y_i$  im multidimensionalen Lösungsraum kann als Vektor mit Prioritätswerten interpretiert werden. Durch diese Prioritätsstruktur kann die spezifische Ablauffolge einer Lösung  $y_i$  bestimmt werden. Die Kraft wirkt auf die Prioritätswerte eines Punktes  $y_i$  im Intervall  $[p_{min}, p_{max}]$  ein. Das Intervall wird durch zufällige Auswahl von  $p_{min} \in [1; n - 1]$  und  $p_{max} \in [2; n]$  mit  $p_{min} \leq p_{max}$  und einer minimalen Distanz zwischen  $p_{min}$  und  $p_{max}$  beschrieben. Aufgrund der Kraftausübung  $KA_{ij}$  von Punkt  $y_j$  aus  $\mathcal{B}_1$  auf Punkt  $y_i$  aus  $\mathcal{B}_2$  werden die Prioritäten eines Punktes  $y_i$  im Intervall  $\in [p_{min}, p_{max}]$  aktualisiert. Die anderen Komponenten der Prioritätsstruktur der Lösungen werden anschließend aktualisiert. Die Prioritätswerte kleiner als  $p_{min}$  werden durch eine  $n$ -große Konstante ( $n$  oder größer) geschmälert und diese Konstante wird allen Prioritätswerten größer  $p_{max}$  hinzugerechnet. Dadurch bleibt die Ablaufstruktur der durch die Kräfte unberührten Arbeitsgänge und die relativen Prioritäten der drei entsprechenden Untergruppen von Arbeitsgängen unberührt. Die relativen Prioritäten für die Arbeitsgänge werden für die Werte von  $< p_{min}$  und  $> p_{max}$  weiterhin durch die optimale Lösung gegeben. Damit bleibt die Prioritätsstruktur erhalten, analog zu den durch den EM bestimmten neuen Prioritäten zwischen  $p_{min}$  und  $p_{max}$ .

Durch Änderungen der Prioritätsstruktur einer möglichen Lösung  $y_i$  aus der Menge  $\mathcal{B}_2$  aufgrund der Wirkung der Kraftausprägung  $KA_{ij}$  kann eine neue Lösung generiert werden.

<sup>63</sup>Vgl. Debels et al. (2006), S. 644-645

<sup>64</sup>Vgl. Tang et al. (2012), S. 5220

## 4 Leistungsbewertung der vorgestellten Heuristiken

Der Vergleich zwischen den Heuristiken wird in Anlehnung an die Arbeit von Kolisch und Hartmann (2006) getätigt. Die Autoren haben in ihrer Arbeit mehrere moderne Heuristiken zum Lösen des RCPSP beschrieben und anhand einer Computersimulation verglichen. In der neuen Untersuchung wurden die selben Testmengen und Stoppkriterien wie in ihren vorherigen Arbeiten verwendet.<sup>65</sup> Es wurden drei unterschiedliche Projekte mit einer Anzahl von Arbeitsgängen bestimmt (30 Arbeitsgänge mit 480 Instanzen, 60 Arbeitsgänge mit 480 Instanzen und 120 Arbeitsgänge mit 600 Instanzen). Die Projekte wurden generiert durch ProGen<sup>66</sup> und sind der PSPLIB entnommen.<sup>67</sup> Für die verschiedenen Testaufbauten wurde eine maximale Anzahl an Planung von 1.000, 5.000 und 50.000 gewählt, da die Autoren die Annahme treffen, dass der Rechenaufwand zur Konstruktion einer Planung für jede Heuristik gleich ist. Zum Testen der Heuristik wurden die originalen Implementationen und besten Konfigurationen der Autoren genutzt. Teilweise wurde aufgrund anderer Stoppkriterien oder mangelnder Daten eine Anpassung der Heuristiken durchgeführt bzw. die Autoren haben die Anforderungen der Studie von Kolisch und Hartmann (2006) auf ihre entwickelten Heuristiken übertragen und die neu ermittelten Ergebnisse bereitgestellt.

Die Ergebnisse der Computersimulation von Kolisch und Hartmann (2006) der in Kapitel 3 beschriebenen Heuristiken sind in den Tabellen 1-3 zusammengefasst. In den Tabellen wird die durchschnittliche Abweichung in Prozent in Abhängigkeit des Stoppkriteriums „Maximale Anzahl an Planungen“ aufgeführt. Dabei muss beachtet werden, dass bei dem Projekt mit 30 Arbeitsgängen eine optimale Lösung bekannt ist. Daher bezieht sich die durchschnittliche Abweichung auf diesen Wert, anders als bei den anderen zwei Projekten. Bei den

<sup>65</sup>Vgl. Hartmann und Kolisch (2000), S. 398-405

<sup>66</sup>Vgl. Kolisch et al. (1995), S. 1695-1696

<sup>67</sup>Für mehr Details zum Testaufbau siehe Kolisch und Sprecher (1997), S. 205-216.

Projekten mit 60 und 120 Arbeitsgängen wird als Wert für die durchschnittliche Abweichung die Projektfertigstellungsdauer anhand des kritischen Pfads als untere Schranke verwendet. In den Tabellen wird ebenfalls aufgeführt, welches SGS bei der Heuristik zum Einsatz kommt. Sortiert wurden die Heuristiken in den Tabellen anhand der Ergebnisse des Stoppkriteriums: Maximale Anzahl an Planung in Höhe von 50.000. Falls eine Heuristik keine Lösung bei der Generierung von 50.000 Planungen ermitteln konnte, wurde das Ergebnis nach der Generierung von 5.000 Planungen als Sortierung verwendet.

Tabelle 1 Durchschnittliche Abweichung (%) von der optimalen Fertigstellungsdauer;  $J = 30$

Kap.	Algorithmus	SGS	Referenz	Max. #Planungen		
				1000	5000	50000
3.3	GA/TS-PNV	Beide	Kochetov und Stolyar	0,10	0,04	0,00
3.4	Scatter Search-EM	Seriell	Debels et al.	0,27	0,11	0,01
3.2	Sampling-LFT/FBI	Beide	Tormos und Lova	0,30	0,17	0,09
3.1	SA-Aktivitätenliste	Seriell	Bouleimen und Lecocq	0,38	0,23	–

**Quelle:** Kolisch und Hartmann (2006), S. 29-32

Die Tabelle 1 zeigt, dass das Verfahren von Kochetov und Stolyar (2003) bei einer Anzahl von 50.000 Planungen bei einem Projekt mit 30 Arbeitsgängen anscheinend eine optimale Lösung und bei den anderen Fällen vergleichsweise gute Ergebnisse liefert. Die anderen Heuristiken liefern ein ähnlich gutes Bild bei der Ermittlung der Lösungen. Erkennbar wird, dass die in dieser Arbeit betrachteten Heuristiken bei kleinen Projekten gute Ergebnisse hervorbringen.

Tabelle 2 Durchschnittliche Abweichung (%) vom kritischen Pfad als untere Schranke;  $J = 60$

Kap.	Algorithmus	SGS	Referenz	Max. #Planungen		
				1000	5000	50000
3.4	Scatter Search-EM	Seriell	Debels et al.	11,73	11,10	10,71
3.3	GA/TS-PNV	Beide	Kochetov und Stolyar	11,71	11,17	10,74
3.2	Sampling-LFT/FBI	Beide	Tormos und Lova	12,14	11,82	11,77
3.1	SA-Aktivitätenliste	Seriell	Bouleimen und Lecocq	12,75	11,90	–

**Quelle:** Kolisch und Hartmann (2006), S. 29-32



Bei einer größeren Anzahl an Arbeitsvorgängen in einem Projekt liefert die Heuristik von Debels et al. (2006) bessere Ergebnisse als die Heuristik von Kochetov und Stolyar (2003), wobei die Unterschiede im Hundertstelbereich der Relation liegen. Dies kann der Tabelle 2 entnommen werden. Einige größere Unterschiede liefern die Heuristik von Tormos und Lova (2003) sowie die Heuristik von Bouleimen und Lecocq (2003) zu den vorher genannten Heuristiken. Deren Abweichungen von der durch den kritischen Pfad bestimmten Projektfertigstellungsdauer als untere Schranke liegen bereits im Zehntelbereich. Aber trotz der Unterschiede kann festgestellt werden, dass alle vier Heuristiken ähnliche Ergebnisse liefern, allerdings ist die Abweichung ähnlich angestiegen.

Tabelle 3 Durchschnittliche Abweichung (%) vom kritischen Pfad als untere Schranke;  $J = 120$

Kap.	Algorithmus	SGS	Referenz	Max. #Planungen		
				1000	5000	50000
3.4	Scatter Search-EM	Seriell	Debels et al.	35,22	33,10	31,57
3.3	GA/TS-PNV	Beide	Kochetov und Stolyar	34,74	33,36	32,06
3.2	Sampling-LFT/FBI	Beide	Tormos und Lova	36,24	35,56	34,77
3.1	SA-Aktivitätenliste	Seriell	Bouleimen und Lecocq	42,81	37,68	–

**Quelle:** Kolisch und Hartmann (2006), S. 29-32

Beim Betrachten der Tabelle 3, bei der die Ergebnisse eines Projekts mit 120 Arbeitsgängen dargestellt werden, ergibt sich ein ähnliches Bild wie bei dem Fall der Generierung von 50.000 Planungen. Die Heuristik von Debels et al. (2006) liefert hier den besten Wert, gefolgt von der Heuristik von Kochetov und Stolyar (2003) sowie der Heuristik von Tormos und Lova (2003). Die gleiche Reihenfolge der drei Heuristiken ergibt sich bei einer Anzahl von 5.000 Planungen. Bei der Anzahl von 1.000 Planungen ergibt sich jedoch ein anderes Bild. Hier scheint die Heuristik von Kochetov und Stolyar (2003) bessere Ergebnisse zu liefern, als die Heuristik von Debels et al. (2006). Damit lässt sich sagen, dass bei einem kleineren Planungslimit die Heuristik von Kochetov und Stolyar (2003) bessere Lösungen liefert – zumindest in der untersuchten Projektstruktur.

Angemerkt werden muss jedoch, dass in der Vergleichsstudie von Kolisch und Hartmann (2006) keine generierte Lösung anhand der Heuristik von Bouleimen und Lecocq (2003) bei keinem der Projekte mit einem Planungslimit in Höhe von 50.000 Planungen hinterlegt wurde.

## 5 Schlussbemerkung

### 5.1 Zusammenfassung

Für das RCPSP liefern Metaheuristiken basierend auf einer Ablauffolge des seriellen SGS die besten Lösungen.<sup>68</sup> Die Heuristiken von Debels et al. (2006) sowie Kochetov und Stolyar (2003) verfolgen eine populations-basierte Metaheuristik. In der Leistungsbewertung liefert das Verfahren von Debels et al. (2006) in den Fällen mit größerer Anzahl an Arbeitsgängen die besten Lösungen, sofern die Heuristik<sup>69</sup> mit dem Verfahren des EM-Algorithmus erweitert wird. Eine Zusammenfassung über die Eigenschaften der Heuristiken liefert Tabelle 4.

Tabelle 4 Eigenschaften der betrachteten Heuristiken

Kap.	Algorithmus	Eigenschaften
3.1	SA-Aktivitätenliste	Lokale Suche Kontrollierte Annahme von Nachbarlösungen
3.2	Sampling-LFT/FBI	Prioritätswertverfahren anhand MLFT Ermittlung der Ablauffolge anhand RBRS Optimierung anhand eines selektiven BF-Verfahrens
3.3	GA/TS-PNV	Optimierung mittels Lösungsmenge (Randomisierung) Crossover durch Pfadneuverknüpfung Suche nach entfernten neuen Lösungsmengen
3.4	Scatter Search-EM	Optimierung mittels Lösungsmenge (Randomisierung) Prioritätswertverfahren Crossover durch Elektromagnetismusanalogie

Andere Studien belegen, dass das Verfahren des Samplings mit Vorwärts-Rückwärts-Methode die besten Lösungen liefert, zumindest im Vergleich zu hybriden Metaheuristiken bei Planung eines Projekts mit geringer Anzahl an

<sup>68</sup>Vgl. Kolisch und Hartmann (2006), S. 33

<sup>69</sup>Scatter Search, als Sonderfall der populations-basierten Metaheuristiken

Arbeitsgängen.<sup>68</sup> Lt. dieser Aussage nimmt die Planungsqualität bei Metaheuristiken mit zunehmender Anzahl von Arbeitsgängen zu. Dies kann anhand des eingeschränkten Vergleichs der Heuristiken in Kapitel 4 nicht bestätigt werden. Valls et al. (2005) zeigten das Potential der Vorwärts-Rückwärts-Methode. Ebenfalls stellen Kolisch und Hartmann (2006) über einen Trend der Anwendung der Vorwärts-Rückwärts-Methode beim Lösen des RCPSP fest. Dies liegt an der leichten Implementierung der Vorwärts-Rückwärts-Methode in bestehende Heuristiken des RCPSP, da es bei jeder Teilplanung durchgeführt werden kann.<sup>70</sup> Dies wurde in der Heuristik von Tormos und Lova (2003) mit einem BF-Ansatz gezeigt.

Angemerkt werden muss, dass die Arbeit von Bouleimen und Lecocq (2003) einige Unklarheiten enthält. Dadurch entstehen Schwierigkeiten im Verständnis des Algorithmus. Unklar bleibt zum einen der mathematische Term  $N_n = N_n(1 + h.n)$ , der mehrfach im Originalartikel in dieser Form beschrieben ist. Daher wird angenommen, dass der Parameter  $h$  ein Index für die Iteration  $n$  hat und dementsprechend müsste der Term  $N_n = N_n(1 + h_n)$  lauten. Zum anderen werden die unterschiedlichen Indizes für die Menge an Nachbarlösungen  $N_{sol}$ ,  $N_p$ , etc. nur eingeschränkt erklärt. Ebenfalls werden andere Abkürzungen, wie z. B. CP-Wert und SPT, nicht definiert und dadurch wird der Artikel nur fachkundigen Lesern empfohlen.

Die Arbeit von Kochetov und Stolyar (2003) ist ebenfalls zu kritisieren, da die Heuristik sehr verkürzt beschrieben ist. Dadurch sind nur wenige Zusammenhänge der einzelnen Schritte der Heuristik erkennbar. Außerdem sind wichtige Verfahren zum Anwenden der Heuristik nicht beschrieben, wie bspw. die Anwendung des GRASP-Algorithmus zum Lösen des multi-dimensionalen Knapsack-Problems oder die Implementierung der PNV in die Heuristik. Es wird somit ein breites Grundwissen im Lösen von Heuristiken angenommen, was die Verständlichkeit der Heuristik einschränkt.

## 5.2 Ausblick

Der in dieser Arbeit aufgeführte Vergleich zeigt, dass eine Heuristik mit seriellem SGS die beste Planungsqualität liefert. Kolisch und Hartmann (2006)

---

<sup>70</sup>Vgl. Valls et al. (2004), S. 18-19

empfehlen für das RCPSP, die Ergebnisse eines seriellen SGS durch Ergänzung des parallelen SGS zu verbessern.<sup>71</sup> Kochetov und Stolyar (2003) liefern ein alternatives Verfahren zum gleichzeitigen Anwenden des seriellen und partiellen SGS, damit nur ein Teil der Planung jeweils durch die Verfahren generiert wird. Daher sollte die nachfolgende Forschung ein möglichst effizientes Verfahren zur gleichzeitigen Anwendung des SGS entwickeln.

Weiter konnte die Studie von Kolisch und Hartmann (2006) weitere effiziente Verfahren aufzeigen, die im Fokus weiterer Forschung stehen sollten. Beispielsweise das Verfahren von Tormos und Lova (2001), dass mit einer Vorwärts-Rückwärts-Methode eine Planung verbessert, die durch das Verfahren der X-Pass-Methode oder Metaheuristik generiert wurde. Das Verfahren liefert durch einfaches Verschieben der Arbeitsgänge gute Ergebnisse und kann zu den meisten anderen Ansätzen hinzugefügt werden.

Die neueren Heuristiken versuchen durch Hinzufügen anderer oder neuerer Methoden sowie der teilweisen Nutzung beider Verfahren des SGS, effizientere Ergebnisse zu generieren.<sup>72</sup> Weitere Forschung sollte sich mit der systematischen Kombination der verschiedenen Ansätze beschäftigen. Als Ergebnis aus dieser Arbeit lässt sich ableiten, dass der Fokus für die Entwicklung neuer hybrider Heuristiken nicht unmittelbar nur auf den Verfahren des GA liegen sollte, da auch andere Verfahren wie bspw. das Sampling gute Ergebnisse für kleine Projekte zu liefern scheint. Aber das Augenmerk sollte auf den Verfahren der populations-basierten Metaheuristiken liegen, besonders mit der Integration einer Vorwärts-Rückwärts-Methode, da diese gute Planungsqualität für das RCPSP liefern.<sup>72</sup>

---

<sup>71</sup>Aussage aufgrund Ergebnisse von Hartmann (2002) mit denen von Hartmann (1998).

<sup>72</sup>Vgl. Kolisch und Hartmann (2006), S. 33-34

## Literaturverzeichnis

- Bartels, J.H. (2009): Projektplanung–Grundlagen und Anwendungsbeispiele. In: Anwendung von Methoden der ressourcenbeschränkten Projektplanung mit multiplen Ausführungsmodi in der betriebswirtschaftlichen Praxis. Springer, S. 7–42.
- Birbil, Ş.İ. und Fang, S.C. (2003): An electromagnetism-like mechanism for global optimization. In: Journal of global optimization. Bd. 25, Nr. 3, S. 263–282.
- Blazewicz, J.; Lenstra, J.K. und Kan, A. (1983): Scheduling subject to resource constraints: classification and complexity. In: Discrete Applied Mathematics. Bd. 5, Nr. 1, S. 11–24.
- Bouleimen, K. und Lecocq, H. (2003): A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. In: European Journal of Operational Research. Bd. 149, Nr. 2, S. 268–281.
- Brucker, P. und Knust, S. (2006): Resource-Constrained Project Scheduling. In: Complex Scheduling. GOR - Publications. Springer Berlin Heidelberg, S. 91–188.
- Debels, D.; De Reyck, B.; Leus, R. und Vanhoucke, M. (2006): A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. In: European Journal of Operational Research. Bd. 169, Nr. 2, S. 638–653.
- Demeulemeester, E. und Herroelen, W. (2011): Robust project scheduling. Bd. 3. Now Publishers Inc.
- Demeulemeester, E.L. (2002): Project scheduling: a research handbook. Bd. 102. Springer.

- DIN 69900 (2009): Projektmanagement - Netzplantechnik; Beschreibung und Begriffe. In: Berlin: Beuth.
- Domschke, W. und Drexl, A. (2005): Einführung in operations research. Bd. 5. Springer.
- Drexl, A.; Kolisch, R. und Sprecher, A. (1997): Neuere Entwicklungen in der Projektplanung. In: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, S. 95–120.
- Felkai, R. und Beiderwieden, A. (2011): Analysieren und Formulieren von Projektzielen. In: Projektmanagement für technische Projekte. Springer, S. 45–64.
- Feo, T.A. und Resende, M.G. (1989): A probabilistic heuristic for a computationally difficult set covering problem. In: Operations research letters. Bd. 8, Nr. 2, S. 67–71.
- Feo, T.A. und Resende, M.G. (1995): Greedy randomized adaptive search procedures. In: Journal of global optimization. Bd. 6, Nr. 2, S. 109–133.
- Garey, M.R. und Johnson, D.S. (1979): Computers and intractability.
- Gerdes, I.; Klawonn, F. und Kruse, R. (2004): Evolutionare Algorithmen. In: Vieweg, Wiesbaden.
- Glover, F. und Laguna, M. (1999): Tabu search. Springer.
- Glover, F.; Laguna, M. und Martí, R. (2000): Fundamentals of scatter search and path relinking. In: Control and cybernetics. Bd. 39, Nr. 3, S. 653–684.
- Hartmann, S. (1998): A competitive genetic algorithm for resource-constrained project scheduling. In: Naval Research Logistics (NRL). Bd. 45, Nr. 7, S. 733–750.
- Hartmann, S. (1999): Project scheduling under limited resources. Springer.
- Hartmann, S. (2002): A self-adapting genetic algorithm for project scheduling under resource constraints. In: Naval Research Logistics (NRL). Bd. 49, Nr. 5, S. 433–448.

- Hartmann, S. und Kolisch, R. (2000): Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. In: European Journal of Operational Research. Bd. 127, Nr. 2, S. 394–407.
- Hering, E. (2014): Projektmanagement für Ingenieure. Springer-Verlag.
- Herroelen, W. (2005): Project scheduling-Theory and practice. In: Production and Operations Management. Bd. 14, Nr. 4, S. 413–432.
- Holland, J.H. (1975): Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.. U Michigan Press.
- Hromkovič, J. (2014): Einleitung. In: Theoretische Informatik. Springer Fachmedien Wiesbaden, S. 1–12.
- Kellenbrink, C. (2014a): Ein Genetischer Algorithmus zur Lösung des RCPSP-PS. In: Ressourcenbeschränkte Projektplanung für flexible Projekte. Produktion und Logistik. Springer Fachmedien Wiesbaden, S. 75–114.
- Kellenbrink, C. (2014b): Einführung in die ressourcenbeschränkte Projektplanung. In: Ressourcenbeschränkte Projektplanung für flexible Projekte. Springer, S. 5–18.
- Klein, R. (2000): Resource-Constrained Scheduling Problems. Springer.
- Koch, S. (2014): Genetische Algorithmen - Allgemeine Grundlagen. In: Genetische Algorithmen für das Order Batching-Problem in manuellen Kommissioniersystemen. Produktion und Logistik. Springer Fachmedien Wiesbaden, S. 59–93.
- Kochetov, Y. und Stolyar, A. (2003): Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem. In: Proceedings of the 3rd international workshop of computer science and information technologies. Bd. 132.
- Kolisch, R. (1996): Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. In: European Journal of Operational Research. Bd. 90, Nr. 2, S. 320–333.

- Kolisch, R. und Drexl, A. (1996): Adaptive search for solving hard project scheduling problems. In: Naval Research Logistics (NRL). Bd. 43, Nr. 1, S. 23–40.
- Kolisch, R. und Hartmann, S. (2006): Experimental investigation of heuristics for resource-constrained project scheduling: An update. In: European Journal of Operational Research. Bd. 174, Nr. 1, S. 23–37.
- Kolisch, R. und Sprecher, A. (1997): PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. In: European Journal of Operational Research. Bd. 96, Nr. 1, S. 205–216.
- Kolisch, R.; Sprecher, A. und Drexl, A. (1995): Characterization and generation of a general class of resource-constrained project scheduling problems. In: Management science. Bd. 41, Nr. 10, S. 1693–1703.
- Li, K. und Willis, R. (1992): An iterative scheduling technique for resource-constrained project scheduling. In: European Journal of Operational Research. Bd. 56, Nr. 3, S. 370–379.
- Lühlig, T. (2006): Risikomanagement in der Produktentwicklung der deutschen Automobilindustrie: von der Konzeptentwicklung bis zum Produktionsanlauf. Shaker.
- Mitchell, M. (1998): An introduction to genetic algorithms. MIT press.
- Neumann-Braun, K.; Schwindt, C. und Zimmermann, J. (2003): Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions. Springer.
- Nissen, V. (1997): Überblick und thematische Einordnung. In: Einführung in Evolutionäre Algorithmen. Computational Intelligence. Vieweg+Teubner Verlag, S. 1–32.
- Pritsker, A.A.B.; Waiters, L.J. und Wolfe, P.M. (1969): Multiproject scheduling with limited resources: A zero-one programming approach. In: Management science. Bd. 16, Nr. 1, S. 93–108.



- Resende, M.G. und Ribeiro, C.C. (2014): GRASP: greedy randomized adaptive search procedures. In: *Search Methodologies*. Springer, S. 287–312.
- Schirmer, A. (1997): Parameterized heuristics for project scheduling-biased random sampling methods. In: *Technical Report*. Bd. 456.
- Talbi, E.G. (2009): *Metaheuristics: from design to implementation*. Bd. 74. John Wiley & Sons.
- Talbot, F.B. (1982): Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. In: *Management Science*. Bd. 28, Nr. 10, S. 1197–1210.
- Tang, J.; Zhang, G.; Zhang, B. und Cen, H. (2012): Resource-constrained Project Scheduling Using Electromagnetism-based Scatter Search. In: *Journal of Computational Information Systems*. Bd. 8, Nr. 12, S. 5219–5227.
- Tormos, P. und Lova, A. (2001): A competitive heuristic solution technique for resource-constrained project scheduling. In: *Annals of Operations Research*. Bd. 102, Nr. 1-4, S. 65–81.
- Tormos, P. und Lova, A. (2003): An efficient multi-pass heuristic for project scheduling with constrained resources. In: *International Journal of Production Research*. Bd. 41, Nr. 5, S. 1071–1086.
- Trautmann, N. und Baumann, P. (2009): Resource-allocation capabilities of commercial project management software: An experimental analysis. In: *Computers & Industrial Engineering, 2009. CIE 2009. International Conference on*. IEEE, S. 1143–1148.
- Valls, V.; Ballestín, F. und Quintanilla, S. (2004): A population-based approach to the resource-constrained project scheduling problem. In: *Annals of Operations Research*. Bd. 131, Nr. 1-4, S. 305–324.
- Valls, V.; Ballestín, Francisco, F. und Quintanilla, S. (2005): Justification and RCPSP: A technique that pays. In: *European Journal of Operational Research*. Bd. 165, Nr. 2, S. 375–386.
- Voigt, K.I. und Schewe, G. (2014): *Definition Projekt Version 7 - Gabler Wirtschaftslexikon*.

Zimmermann, J.; Stark, C. und Rieck, J. (2006a): Projektplanung unter Zeitrestriktionen. In: Projektplanung. Springer-Lehrbuch. Springer Berlin Heidelberg, S. 115–196.

Zimmermann, J.; Stark, C.; Rieck, J. et al. (2006b): Projektmanagement. In: Projektplanung. Springer Berlin Heidelberg, S. 1–113.

# Anhang

Ergänzung zum EM-Algorithmus (3.4):

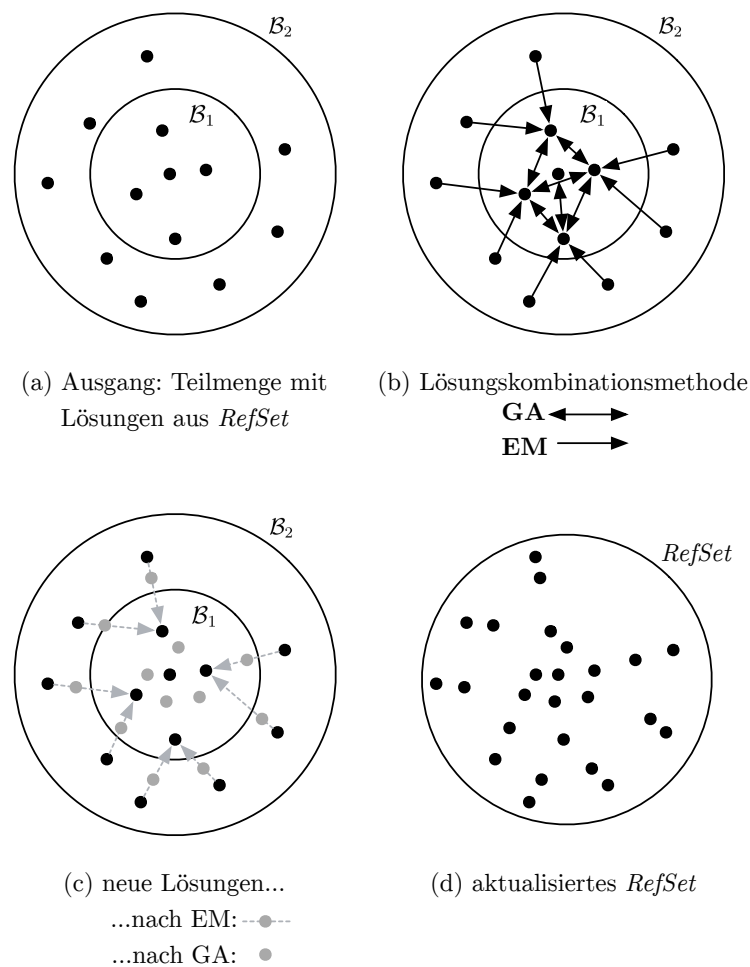


Abbildung 6 Grafische Veranschaulichung des EM-Algorithmus

Abbildung (a) zeigt die generierte Teilmenge  $B_1$  und  $B_2$  mit ihren jeweiligen Lösungspunkten. Dabei repräsentiert der Mittelpunkt der zwei Kreise die optimale Lösung. Je weiter ein Punkt  $y$  vom Mittelpunkt entfernt ist, desto größer ist der Zielfunktionswert  $f(y)$  und dementsprechend länger ist die

Projektfertigstellungsdauer durch Anwendung der Ablauffolge des Punkts  $y$ . Abbildung (b) veranschaulicht die mögliche Lösungskombinationsmethode des GA-Algorithmus für  $\mathcal{B}_1 \times \mathcal{B}_1$  und des EM-Algorithmus für  $\mathcal{B}_1 \times \mathcal{B}_2$ . Die Abbildung (c) zeigt die generierten neuen Lösungspunkte nachdem die Lösungskombinationsmethoden vollzogen wurden. Abbildung (d) zeigt die aktualisierte Referenzmenge *RefSet* nachdem die neuen Punkte bzw. Lösungen generiert wurden.

## Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Hannover, 17. September 2014

