

White paper

Algorithmen zur Ressourcenbelegungsplanung in der Produktion

Karlsruhe, im Frühjahr 2011

Leistungsfähige Algorithmen zur Ressourcenbelegungsplanung in der Produktion

Dr. Michael Baumann
Fraunhofer-Institut IOSB, Karlsruhe

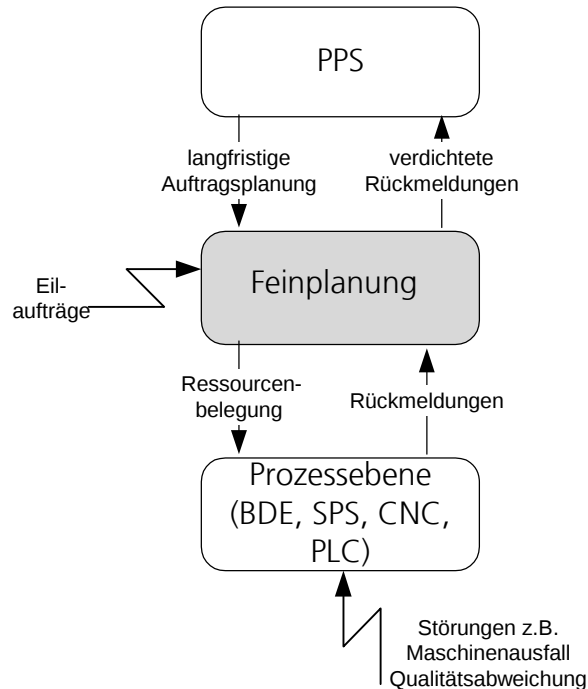
1 Motivation

Zum Erhalt der Konkurrenzfähigkeit sind kleinere und mittelständische Unternehmen gezwungen, kundenorientiert zu produzieren. Kleine Liefereinheiten, hohe Produktvielfalt, kurze Lieferzeiten und hohe Qualitätsanforderungen sind Kennzeichen dieser Entwicklung. In der Produktion führt dies zu steigendem Auftragsbestand bei kleineren Losgrößen und hohem Termindruck.

Um diesen Anforderungen gerecht zu werden, ist der Einsatz von Softwarewerkzeugen zur Unterstützung der Fertigungssteuerung notwendig. Die zentrale Aufgabe liegt hier in der kurzfristigen Feinplanung der Ressourcenbelegung, d.h. der zeitliche Zuordnung von Maschinen, Personal und Betriebsmitteln zu den zu produzierenden Aufträgen. Bei der klassischen Maschinenbelegungsplanung liegt das primär verfolgte Ziel in der Einhaltung der Sollendtermine. Als wesentliche Randbedingung muss die begrenzte Kapazität der Maschinen (Kapazitätsrestriktion) und die zulässige Bearbeitungsreihenfolge der Arbeitsvorgänge (Reihenfolgebeziehung) berücksichtigt werden.

Da sich die Verhältnisse in der Fertigung bedingt durch Störungen, Eilaufträge etc. ständig ändern, muss die Feinplanung fortlaufend überprüft und ggf. dem veränderten Systemzustand angepasst werden. Die Voraussetzung hierfür ist die Rückmeldung der Betriebs- und Maschinendaten, um den aktuellen Systemzustand abzubilden. In den meisten Betrieben erfolgt die Rückmeldung durch den Einsatz entsprechender Softwareprodukte zur Betriebs- (BDE) und Maschinendatenerfassung (MDE). Durch diese Informationsrückkopplung liegt die Feinplanung in einem geschlossenen Regelkreis und arbeitet somit nicht als passives Steuerelement, sondern als aktiver Regler.

Abbildung 1:
 Regelkreise der
 Fertigungsplanung



Das Ressourcenbelegungsproblem ist ein äußerst komplexes kombinatorisches Optimierungsproblem, für dessen exakte Lösung es derzeit kein echtzeitfähiges Verfahren gibt. Erschwerend kommt hinzu, dass die Optimierung der Ressourcenbelegung nach einer einzigen Zielfunktion nicht ausreicht, vielmehr sind zur Minimierung der Kosten mehrere Zielfunktionen gleichzeitig zu berücksichtigen, z.B.

- Termineinhaltung,
- Durchlaufzeiten,
- Ressourcenauslastung,
- Lagerbestände,
- Rüstaufwände.

Es liegt also ein ‚multikriterielles Optimierungsproblem‘ vor. Die besondere Schwierigkeit liegt hierbei darin, dass die vorgegebenen Zielkriterien meist konträr sind, und es keine Lösung gibt, die hinsichtlich aller Kriterien optimal ist.

Die Lösung des Ressourcenbelegungsproblems wird auch durch die Berücksichtigung zusätzlicher Restriktionen erschwert. Z.B. wenn neben den erwähnten Kapazitätsrestriktionen und Reihenfolgebeziehungen etwa folgende Randbedingungen berücksichtigt werden müssen:

- Bearbeitungszeiten, die z.B. von den eingesetzten Ressourcen oder anderen Bedingungen abhängen,
- zeitliche Beziehungen zwischen Arbeitsvorgängen, um z.B. einzuhaltende Abkühlzeiten (Minimalabstand) oder das Aufrechterhalten einer gewissen Werkstücktemperatur nach vorheriger Erwärmung sicherzustellen (Maximalabstand),
- Unterbrechbarkeit der Arbeitsvorgänge, wenn ein Arbeitsvorgang etwa durch eine Pause unterbrochen werden kann oder wenn ein Arbeitsvorgang unterbrochen werden kann, um einen höherpriorien zu starten,
- überlappende Fertigung, d.h. ein Nachfolge-Arbeitsvorgang kann bereits mit einer Teilmenge gestartet werden,
- Präzedenzbeziehungen zwischen Arbeitsvorgängen unterschiedlicher Fertigungsaufträge, um diese beispielsweise zeitlich zu synchronisieren,
- Mehrbedarfsplanung, wenn nicht nur Maschinen, sondern alle relevanten Ressourcenbedarfe wie etwa Personal, Werkzeuge, Material geplant werden müssen,
- Splitten von Arbeitsvorgängen zur Erhöhung der Parallelität bei der Bearbeitung oder Bündeln von Arbeitsvorgängen, um den notwendigen Rüstaufwand zu minimieren,
- parallele Ressourcen, die sich in der Bearbeitung komplett ersetzen können,
- Rüstzeiten, die entweder statisch vorgegeben sind oder von den Bearbeitungsreihenfolgen auf den Maschinen abhängen,
- Parallelbearbeitung von mehreren Arbeitsvorgängen auf einer Ressource,
- Verschleißverhalten von Ressourcen,
- Wartungsmodelle,
- Verfügbarkeit der Ressourcen über Schichtmodelle, Pausenmodelle etc.,

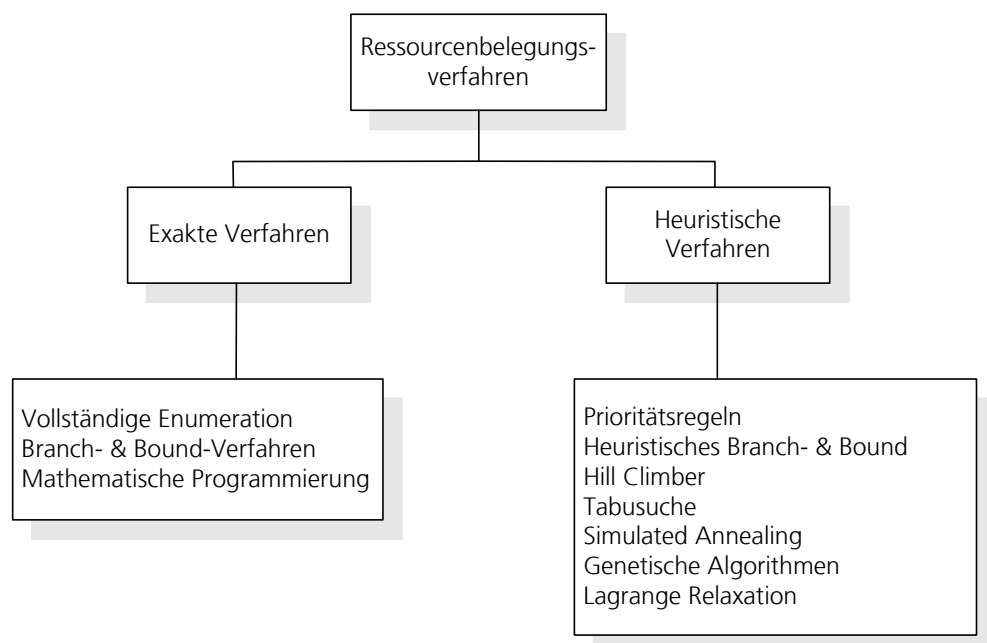
- Zwischenlagerbeschränkungen,
- Transportzeiten und/oder Transportmittel zwischen den Arbeitsvorgängen.

2 Lösungsverfahren zur Planung der Ressourcenbelegung

Wir entwickeln und vergleichen permanent bekannte und neue Optimierungsverfahren

Das Fraunhofer Institut IOSB erforscht und entwickelt innovative Lösungen zur Ressourcenbelegungsplanung. Das Angebot des IOSB erstreckt sich von der Analyse der Problemstellung über eine Pflichtenheftphase, Konzeption und Beratung bis hin zur Implementierung und Wartung sowie Schulung. Unsere Erfahrungen auf dem Gebiet der Ressourcenbelegungsplanung wurden im Laufe von 15 bis 20 Jahren im Rahmen von Forschungsarbeiten und deren Umsetzung in die industrielle Anwendung gewonnen. Unsere Kompetenz schließt die Auswahl und Konfiguration von Belegungsverfahren, auch mit Hilfe von Simulationsuntersuchungen, ein. Im folgenden werden eine Reihe von Ansätzen zur Lösung kombinatorischer Optimierungsprobleme erläutert, wie das Ressourcenbelegungsproblem eines darstellt. Diese Ansätze wurden am IOSB theoretisch und praktisch auf Ihre Brauchbarkeit hin untersucht.

Abbildung 2:
Verfahren zur
Ressourcen-
belegungsplanung



Als mögliche exakte Lösungsverfahren lassen sich die vollständige Enumeration, Branch- & Bound-Verfahren und Mathematische Programmierung nennen. Es gibt einige Beispiele in der Forschung für spezifische Ressourcenbelegungsprobleme, in denen diese Verfahren eingesetzt werden. Da das Ressourcenbelegungsproblem jedoch i.d.R. ein schwer lösbares Optimierungsproblem (NP-vollständig) darstellt, werden in der Praxis meist heuristische Lösungsverfahren eingesetzt, die das Problem in annehmbaren Laufzeiten, möglicherweise nicht optimal, aber doch sehr gut lösen. Hierunter fallen im wesentlichen Prioritätsregeln sowie heuristische Branch- & Bound-Verfahren. Genetische Algorithmen sowie lokale Suchverfahren wie Hill Climber, Tabusuche oder Simulated Annealing werden im Anschluss vorgestellt. Zuletzt wird noch auf die Lagrange Relaxation zur Lösung der Ressourcenbelegungsplanung eingegangen.

2.1 Exakte Lösungsverfahren

2.1.1 Vollständige Enumeration

Ein Verfahren zur vollständigen Enumeration erzeugt prinzipiell alle möglichen Lösungen. Daraus werden die nicht zulässigen Lösungen ausgesondert, da irgend eine oder mehrere Restriktionen verletzt sind. Für die verbleibenden zulässigen Lösungen werden die Zielfunktionswerte berechnet und miteinander verglichen. So kann in jedem Fall der optimale Plan bestimmt werden.

Verfahren die alle möglichen Lösungen erzeugen werden explizite Enumerationsverfahren oder greedy-Methoden genannt. Vorteilhafter sind natürlich solche Verfahren, bei denen nur zulässigen Lösungen erzeugt werden. Denn das Aussondern von nicht zulässigen Lösungen entfällt dann.

Nur bei kleinen Problemen anwendbar

Suchaufwand sehr groß

Verfahren dieser Klasse nutzen in der Regel Entscheidungsbäume. Hierbei wird das Optimierungsproblem der Ressourcenbelegung als Folge von Entscheidungen verstanden. Ein Entscheidungsbaum besteht aus Knoten und Kanten. Ein Kante repräsentiert eine Entscheidung einen Arbeitsvorgang den Ressourcen zuzuordnen. Eine Kantenfolge von der Wurzel, bei der noch keine Zuteilungen erfolgt sind, bis zu einem dedizierten Knoten stellt einen Teilbelegungsplan dar, bei dem alle Arbeitsvorgänge der Kantenfolge bereits verplant sind. Die Menge der Blätter enthält alle möglichen vollständigen Belegungspläne, bei denen alle Arbeitsvorgänge zugeteilt sind.

Angenommen die Reihenfolge der Arbeitsvorgänge von n einstufigen Aufträgen soll für eine einzige Maschine festgelegt werden, so sind bereits hierfür $(n-1)!$ Reihenfolgen möglich. Bei 11 Arbeitsvorgängen sind es beispielsweise ca. 3.6 Mio. Möglichkeiten. Bei 20 Arbeitsvorgängen ist eine vollständige

Enumeration bereits nicht mehr durchführbar (121.645.100.408.832.000 Möglichkeiten).

Wesentlich für die Effizienz derartiger Verfahren ist es daher, redundante Knoten zu erkennen und zu eliminieren. Hierfür gibt es Kriterien, die äquivalente Kantenfolgen bzw. Knoten erkennen lassen.

Um den Suchaufwand für diesen Entscheidungsbaum weiter zu reduzieren, kann beispielsweise ein Branch- & Bound-Verfahren eingesetzt werden, bei denen nur die Zweige im Entscheidungsbaum weiter betrachtet werden, die zu einer optimalen Lösung führen können. Diese Verfahren zählen deshalb zur Klasse der impliziten Enumerationsverfahren.

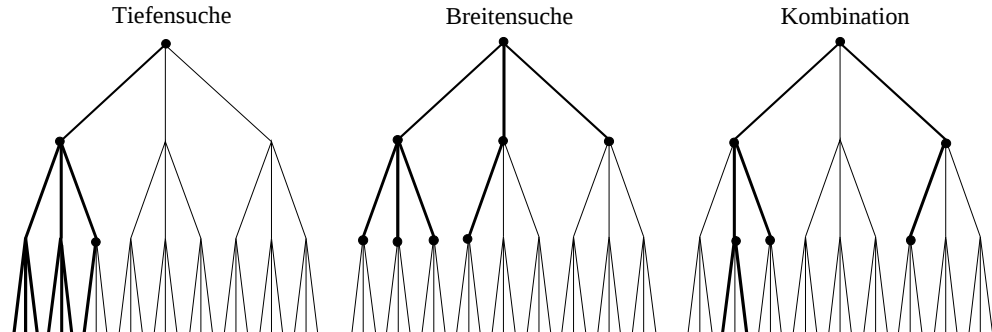
2.1.2 Branch- & Bound-Verfahren

Das wohl bekannteste Prinzip, einen Lösungsraum, hier den Entscheidungsbaum, zu durchsuchen, ist das Branch-and-Bound-Verfahren (B&B-Verfahren). Bei den B&B-Verfahren wird der Lösungsraum immer feiner unterteilt (Branching). Die Teilräume werden anschließend getrennt untersucht. Sofern ausgeschlossen werden kann, dass ein Teilraum keine optimale Lösung enthält, wird er verworfen (Bounding). Somit kann eine optimale Lösung gefunden werden, ohne alle Lösungen explizit aufzuzählen.

„Intelligenter“ als Enumeration
Nur kleine Probleme lösbar

Beim **Branching** wird ein Teilbelegungsplan durch weitere Zuordnungen Arbeitsvorgang zu Ressourcen fortgesetzt. Beim **Bounding** wird ein Teilbelegungsplan daraufhin untersucht, ob er durch Fortsetzung zu einem optimalen vollständigen Belegungsplan werden kann. Ein unterhalb eines Knotens E liegender Teilbaum (Zweig von E) wird nicht weiter untersucht, wenn er eine Restriktion verletzt oder es eine Lösung mit besserem Zielfunktionswert gibt. Der Zielfunktionswert eines Knotens stellt eine untere Schranke dar, die bei Fortsetzung dieses Knotens bestenfalls realisiert werden kann.

Eine wesentliche Komponente eines B&B-Verfahrens ist die **Suchstrategie**, mit der der als nächstes zu betrachtende Knoten ausgewählt wird. Grundsätzlich bieten sich als Suchstrategie die Breiten- oder die Tiefensuche oder eine Kombination beider Strategien an. Bei der Tiefensuche wird der Knoten ausgewählt, der als letzter betrachtet wurde (Lifo-Regel), bei der Breitensuche dagegen der Knoten, der als erster betrachtet wurde (Fifo-Regel). Bei der Kombination aus Breiten- und Tiefensuche kann beispielsweise der Knoten ausgewählt werden, der am aussichtsreichsten erscheint (Best-First-Regel).

Abbildung 3:
 Suchstrategien


Bei der Tiefensuche wird immer nur ein Zweig des Entscheidungsbaumes verfolgt, ein bereits untersuchter Zweig wird abgebaut. Bei der Breitensuche oder der Kombination werden die einzelnen Zweige dagegen parallel aufgebaut. Somit entsteht ein hoher Speicherplatzbedarf und Verwaltungsaufwand. Außerdem ist es vorteilhaft, möglichst frühzeitig einen vollständigen Belegungsplan zu erzeugen, um Zweige aufgrund des Zielfunktionswerts verwerfen zu können.

2.1.3 Mathematische Programmierung

Unter diesem Begriff werden mathematische Formulierungen von beschränkten Optimierungsproblemen verstanden. Je nach der zu minimierenden Zielfunktion werden diese Probleme linear, quadratisch usw. genannt. In Abhängigkeit vom Wertebereich der Entscheidungsvariablen werden diese Probleme als ganzzahlig, gemischt-ganzzahlig usw. bezeichnet. Es handelt sich um Probleme der folgenden Art:

$$\begin{aligned}
 &\text{Minimiere } f(x_1, \dots, x_n) \\
 &\text{unter den Nebenbedingungen: } g_1(x_1, \dots, x_n) \leq b_1 \\
 &\quad \quad \quad \vdots \\
 &\quad \quad \quad g_m(x_1, \dots, x_n) \leq b_m
 \end{aligned}$$

Modellierung durch mathematisches Modell sehr mächtig und allgemein

Nur kleine, einfache Probleme lösbar

Das Ressourcenbelegungsproblem lässt sich in dieser Form modellieren. Hierauf können dann Standardalgorithmen der mathematischen Programmierung angewandt werden. Vorausgesetzt es gäbe effiziente Solver, könnten beliebig komplexe mathematische Modelle aufgestellt und gelöst werden. Das aufwändige und problemspezifische Anpassen von Algorithmen wäre damit vorbei. Reale Probleme führen jedoch zu einer Vielzahl von Variablen. Bereits wenige Ganzzahligkeitsbedingungen für Variablen können zu einer Explosion der Rechenzeit führen, was praktisch die Unlösbarkeit der Modelle nach sich zieht. Aus diesem Grund sind auf diese Weise i.d.R. nur „kleine“ Probleme lösbar. Problemspezifische Algorithmen arbeiten durch das Ausnutzen von problemimmanenten Wissens effizienter.

2.2 Heuristische Verfahren

2.2.1 Prioritätsregeln

Die Entwicklung heuristischer Prioritätsregeln ist wahrscheinlich so alt wie die industrielle Produktion selbst und bis heute nicht abgeschlossen. Es sind zwei Grundtypen von Prioritätsregeln bekannt:

- Kürzeste Operationszeit Regel mit guten Mittelwerten bei Durchlaufzeiten und Terminabweichungen, jedoch starker Streuung der Einzelwerte.
- Terminorientierte Regeln, wie z.B. die Kürzeste-Pufferzeit-Regel mit schlechteren Mittelwerten, jedoch deutlich geringerer Streuung.

**Umfangreich
getestet**

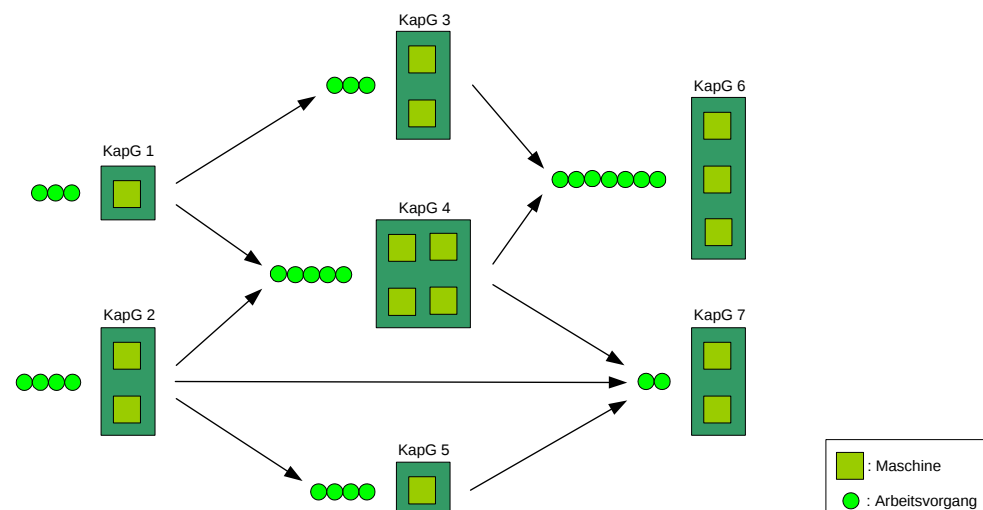
**Relativ ein-
fach zu
verstehen**

**Einsetzbar
im Kurz-
fristbereich**

Bis heute gibt es zahlreiche Versuche, die Vorteile beider Regeltypen zu kombinieren. Dies hat zu einer Vielfalt unterschiedlicher Regeln geführt, die immer wieder in Simulationsstudien untersucht und verglichen wurden. Hierbei wurde das Ziel verfolgt, eine zuverlässige und allgemeingültige Aussage über die Leistungsfähigkeit der verschiedenen Regeln bei wechselnden Randbedingungen zu bekommen.

Ausgehend von einem Warteschlangenmodell der Fertigung ordnen Prioritätsregeln jedem Arbeitsvorgang in der Warteschlange vor einer Kapazitätsgruppe eine Priorität zu. In einer Kapazitätsgruppe werden sich ersetzende Maschinen zusammengefasst. Der nächste Arbeitsvorgang, der bearbeitet werden soll, wird anhand der Priorität ausgewählt.

Abbildung 4:
Warteschlangen-
modell



Wenn der Zeitpunkt der Entscheidung in die Berechnung der Priorität eingeht, handelt es sich um eine dynamische sonst um eine statische Prioritätsregel. Wird nur eine Warteschlange betrachtet handelt es sich um eine lokale, bei Betrachtung eines ganzen Ausschnitts des Warteschlangenmodells um eine globale Prioritätsregel. Fügt man noch ein vorausschauendes Element hinzu, das die Auswirkung der Zuteilung eines Arbeitsvorgangs auf die anderen Arbeitsvorgänge bewertet, erhält man eine prädiktive Regel. Erfolgt die Vorausschau über mehrere Stufen des Warteschlangenmodells, steigt der Aufwand drastisch an und nähert sich dem von Enumerationsverfahren.

2.2.2 Heuristische Branch- & Bound-Verfahren

Einsetzbar
im Kurz-
fristbereich
abhängig von
Suchstrate-
gie

Während exakte Branch- & Bound-Verfahren Zweige nur abschneiden, wenn sie zu keinem optimalen Plan führen können, schneiden heuristische Branch- & Bound-Verfahren Zweige bereits ab, wenn sie wahrscheinlich zu keinem guten Plan führen können. Dies reduziert den Suchaufwand im Entscheidungsbaum beträchtlich, kann aber zum Verlust des Optimums führen.

2.2.3 Genetische Algorithmen

Biologische Organismen haben sich im Laufe von Jahrmillionen von einfachsten Urformen zu unvorstellbarer Komplexität entwickelt. Durch Kreuzung und Mutation entstehen neue Organismen, die sich nur gering von ihren Vorgängern unterscheiden. Die natürliche Selektion stellt auf lange Sicht sicher, dass bessere Ausprägungen einer speziellen Eigenschaft erhalten bleiben.

Genetische Algorithmen sind Suchverfahren, die auf dem Prinzip der Evolution aufbauen: In einer allmählich fortschreitenden Entwicklung setzen sich bei den Vertretern einer Spezies die Merkmale derjenigen Individuen durch, die am stärksten sind, d.h. mit ihrem Umfeld am besten interagieren. Charles Darwin hat als wesentliche Mechanismen dieser Entwicklung die natürliche Selektion der Individuen und die Art ihrer Reproduktion ausgemacht.

Sehr gute
Ergebnisse
erzielbar

Aufwändiges
Optimie-
rungsverfah-
ren für „Ex-
perten“

Überträgt man die Mechanismen der Evolution auf ein Suchverfahren, so repräsentieren die Individuen Lösungen des zu optimierenden Problems, d.h. in unserem Fall des Ressourcenbelegungsproblems. Der prinzipielle Ablauf eines Genetischen Algorithmus' ist etwa folgendermaßen: Zur Initialisierung werden zufällig Individuen erzeugt, und zwar so viele wie die Größe der Population vorgibt, die alle Individuen umfasst. Für jedes Individuum, wird anschließend die Fitness, d.h. der Zielfunktionswert ermittelt. Danach werden durch den Verfahrensschritt Selektion Individuen für die anschließende Reproduktion ausgewählt. Hierbei werden die gemäß Zielfunktionswerten besseren Individuen häufiger ausgewählt, damit sie ihre „Erbinformationen“ entsprechend häufiger weitergeben können. Dadurch setzen sich diese Gene gegenüber anderen in stärkerem Maße durch und erzeugen wiederum „gute“ Individuen. Im An-

schluss werden mittels Kreuzung von jeweils zwei selektierten Individuen Nachkommen erzeugt. Die Nachkommen werden dann durch den Verfahrensschritt Mutation zufälligen Veränderungen unterworfen. Die Verfahrensschritte Selektion, Kreuzung und Mutation werden solange wiederholt, bis ein gegebenes Abbruchkriterium erreicht ist, etwa eine Anzahl Generationen oder eine vorgegebene Laufzeitschranke. Die beste Lösung, die für das zu optimierende Problem ermittelt werden konnte, ist das Individuum mit dem besten Zielfunktionswert in der letzten Generation.

Bei der konkreten Umsetzung dieses Verfahrens auf die Ressourcenbelegungsplanung liegen die Probleme im wesentlichen in der Repräsentation eines Belegungsplans in einer Form, so dass einfach und schnell die Fitness berechnet und selektiert, gekreuzt und mutiert werden kann. Denn je nach Problem muss die Iteration über 1000 und mehr Generationen durchgeführt werden, um zu guten Ergebnissen zu gelangen. Insbesondere unter Berücksichtigung der fortschreitenden Leistungsfähigkeit heutiger Rechner, stellt dies aber bzgl. der Einsetzbarkeit bei realen Problemgrößen kein Ausschlusskriterium für Genetische Algorithmen dar.

Lokale Suchverfahren dagegen variieren eine gefundene Lösung nur leicht, um eine neue Lösung zu erzeugen. Ist die neue Lösung besser als die vorige, wird die neue Lösung ihrerseits Ausgangspunkt für den nächsten Optimierungsschritt. Dieser Prozess wird so lange wiederholt, bis keine bessere Lösung mehr gefunden werden kann. Ein großer Nachteil dieser Optimierungsverfahren ist jedoch, dass sie die Suche vorzeitig, an einem lokalen Optimum, beenden können.

Diese unerwünschte Eigenschaft ist bei globalen Suchverfahren deutlich schwächer ausgeprägt. Sie suchen das Optimum nicht nur in der Nachbarschaft einer guten Lösung. Globale Suchverfahren eignen sich daher besonders für komplexe oder wenig erforschte Suchräume. Da Genetische Algorithmen eine Menge von Suchpunkten in jeder Generation untersuchen, zählt man sie zu den globalen Suchverfahren.

Im Anschluss werden die lokalen Suchverfahren Hill Climber, Tabusuche und Simulated Annealing kurz vorgestellt.

2.2.4 Hill Climber

**Ergebnisse
eher mäßig**
**Schnelles
Suchverfahren**
**Einfach zu
realisieren**

Eines der bekanntesten lokalen Suchverfahren ist der Hill Climber. Das Verfahren startet mit einer zufälligen Lösung, die so lange verändert wird, bis eine Lösung mit einem besseren Zielfunktionswert gefunden wird. Hill Climber suchen nur in der direkten Nachbarschaft einer Lösung eine Verbesserung. Sie müssten daher zufälligerweise in der Nachbarschaft des globalen Optimums beginnen, um dieses überhaupt finden zu können.

Hill Climber finden nur in Ausnahmefällen eine optimale Lösung. Die gefundenen Lösungen in lokalen Optima sind in der Regel weit vom globalen Optimum entfernt.

2.2.5 Tabusuche

Der Tabusuche liegt die Idee zugrunde, bei der Suche nach einer optimalen Lösung nicht mehrfach zu bereits bekannten Lösungen zu gelangen. Die zuletzt ermittelten Lösungen werden dazu in einer Liste fester Größe, der Tabuliste, gespeichert. Lösungen, die in dieser Liste enthalten sind, gelten für eine Zeit als »tabu«, ein Weg zurück zu diesen Lösungen ist nicht erlaubt.

Der nächste Suchschritt wird anders als beim Hill Climber nicht nur in die Richtung unternommen, die besser ist als die aktuelle Lösung. Die Tabusuche merkt sich die beste bisher gefundene Lösung, erlaubt aber auch Suchschritte zu Lösungen, die schlechter sind als die aktuelle. Dazu wählt die Tabusuche aus der Nachbarschaft der aktuellen Lösung eine festgelegte Anzahl Kandidaten für den nächsten Schritt aus. Der Kandidat, der ein vorgegebenes Akzeptanzkriterium erfüllt, etwa den höchsten Zielfunktionswert aller Kandidaten, ersetzt die aktuelle Lösung. Dabei ist ausdrücklich auch eine Verschlechterung des aktuellen Lösungsniveaus gestattet.

**Ergebnisse
besser als
bei Hill
Climber,
aber meist
suboptimal**

Um zu vermeiden, dass die Suche dadurch in einen Zyklus mündet, wird jede aktuelle Lösung in die Tabuliste eingetragen. Bereits bekannte Lösungen sind nicht für die gesamte Dauer der Suche gesperrt, denn auch ein Schritt zurück kann zu einem besseren Nachbarn führen als zuvor. Ein Kriterium, nach dem man Lösungen wieder aus der Tabuliste entfernt, kann die Anzahl der Suchschritte sein, die seit der Evaluierung einer Lösung zurückgelegt wurden. Oder man legt einen Toleranzwert fest, den eine gesperrte Lösung übertreffen muss, um wieder zugelassen zu werden. Als Toleranzwert bietet sich der Zielfunktionswert der aktuellen Lösung an.

Die Suchschritte werden so lange wiederholt, bis ein Abbruchkriterium wie maximale Laufzeit erreicht ist.

Von Bedeutung ist die Größe der Tabuliste, denn die Anzahl der nicht zugelassenen Rückschritte hat großen Einfluss auf das Suchverhalten des Verfahrens. Wählt man die Größe der Tabuliste zu klein, beschränkt sich die Suche auf ein kleines Gebiet um die initiale Lösung. Eine zu große Tabuliste hat den Nachteil, dass der Suchraum sehr schnell durchquert wird, ohne Gebiete genau zu durchsuchen. Damit steigt die Gefahr, sehr gute Lösungen und letztlich das Optimum auszulassen.

Die Idee, bereits ermittelte Lösungen in eine Liste einzutragen, kann auch in Genetischen Algorithmen vorteilhaft genutzt werden. Zum einen lassen sich so

doppelte Individuen in der Population vermeiden, zum anderen kann auch hier die Rückkehr zu schlechteren Lösungen verhindert werden.

2.2.6 Simulated Annealing

Dieses Verfahren imitiert die Abkühlung erhitzter Materialien. Zunächst wird auch in diesem Verfahren eine zufällige Lösung ermittelt. Zu Beginn der Suche werden neben besseren Kandidaten aus der Nachbarschaft auch schlechtere Lösungen akzeptiert. Allerdings werden schlechtere Lösungen nur mit einer Wahrscheinlichkeit akzeptiert, die mit jedem Suchschritt abnimmt. Diese Wahrscheinlichkeit simuliert die physikalische Abkühlung von Materialien. Zur Berechnung dieser Wahrscheinlichkeit wird ein Kontrollparameter eingesetzt, den eine Abkühlungsfunktion in jedem Suchschritt neu bestimmt.

Etwas aufwändiger als Tabusuche

Suchvorgang über Parameter von „Experten“ steuerbar

Durch die Akzeptanz auch schlechterer Lösungen zu Beginn der Suche soll die Gefahr verringert werden, in einem lokalen Optimum stecken zu bleiben, ähnlich wie bei der Tabusuche. Erst zu einem späteren Zeitpunkt, wenn sich die Suche möglicherweise auf das Gebiet des globalen Optimums konzentriert, nimmt Simulated Annealing nur noch bessere Lösungen an. Dennoch ist nicht garantiert, dass Simulated Annealing eine sehr gute Lösung oder sogar das globale Optimum findet. In Abhängigkeit vom Startpunkt der Suche und von der Beschaffenheit des Suchraums kann auch dieses Verfahren in einem lokalen Optimum stehen bleiben.

Bei sehr schneller Abkühlung verhält sich Simulated Annealing ähnlich einem Hill Climber: schlechte Lösungen werden kaum akzeptiert, die Suche konzentriert sich auf ein lokales Optimum. Je langsamer die Abkühlung voran schreitet, desto größer ist die Wahrscheinlichkeit, das globale Optimum zu erreichen.

2.2.7 Lagrange Relaxationsverfahren

Aufwändiges mathematisches Verfahren

Abweichung der Lösung vom Optimum bestimmbar

Das Lagrange Relaxationsverfahren nutzt die Tatsache aus, dass sich das Ressourcenbelegungsproblem als ganzzahliges mathematisches Optimierungsproblem formulieren lässt, d.h. mit einer Zielfunktion und Nebenbedingungen.

Das Verfahren geht nun davon aus, dass durch Lockerung eines Teils der Nebenbedingungen ein einfacher lösbares Problem entsteht. So werden beispielsweise die Kapazitätsbeschränkungen mit Lagrange-Multiplikatoren versehen in der Zielfunktion aufgenommen. Die Multiplikatoren werden mit einem Iterationsverfahren immer wieder neu gesetzt. Bei Abbruch des Verfahrens wird der gefundene Plan im allgemeinen nicht ausführbar sein, denn die Kapazitätsbeschränkung ist möglicherweise verletzt.

Die berechneten Startzeiten der einzelnen Arbeitsvorgänge liefern aber eine implizite Ordnung. Mittels einfacher Heuristik kann das Ergebnis unter Verwendung der ermittelten Startzeiten in einen ausführbaren Plan überführt werden.

Das Verfahren liefert auch eine untere Schranke für die optimale Lösung, damit kann die Abweichung der gefundenen Lösung vom Optimum bestimmt werden. Bei den anderen Verfahren ist die Abweichung vom Optimum nicht bekannt.

Das Lagrange Relaxationsverfahren ist insbesondere bei Ressourcenbelegungsproblemen sinnvoll einsetzbar, wenn die Anzahl der zu berücksichtigenden Nebenbedingungen oder Restriktionen relativ gering ist. Dann lassen sich gute Ergebnisse erzielen.

3 Zusammenfassung

**Ihr Nutzen:
Passendes
Opt.-
verfahren
für Ihre
Anwendungs-
fälle**

**Punktgenaue
Spezifikation**

**Schnelle
Realisierung**

Das Fraunhofer Institut IOSB arbeitet bereits seit ca. 15 bis 20 Jahren auf dem Gebiet der Belegungsplanung. Durch wissenschaftliche Arbeiten wie Studien-, Diplomarbeiten und Dissertationen wurden viele Verfahren theoretisch entwickelt und im Rahmen von Simulationsstudien untersucht. Die Ergebnisse dieser Arbeiten wurden auf wissenschaftlichen Tagungen präsentiert und verteidigt. Der Transfer der erarbeiteten Ergebnisse in die industrielle Praxis ist uns im Rahmen vieler Projekte gelungen. Durch die Erfahrungen aus einer Vielzahl von Simulationsuntersuchungen ist es dem IOSB möglich, problemangepasste heuristische Lösungsverfahren zu entwickeln bzw. Standardverfahren durch geeignete Parametrierung an die spezifische Problemstellung anzupassen.

Wesentlich ist hierbei das Verständnis, wie ein Problem abstrakt mit den Objekten Auftrag, Ressource und Restriktionen modelliert werden kann, so dass das Problem ausreichend detailliert beschrieben ist. Keine wesentlichen Eigenschaften dürfen vernachlässigt werden, gleichzeitig sollte aber nur soviel ins Modell eingebracht werden, wie für die Problemlösung notwendig ist. D.h. das Modell sollte so abstrakt wie möglich und so detailliert wie nötig sein. Die Belegungsverfahren arbeiten dann auf dem Modell, die Ergebnisse werden auf das reale Problem übertragen. Hierbei ist es auch möglich, die Methode der Simulation einzusetzen, um unterschiedliche Verfahren oder Parametereinstellungen oder auch manuelle Planungseingriffe im Rahmen von Simulationsszenarien oder Planspielen miteinander zu vergleichen.

Exakte Lösungsverfahren wie die vollständige Enumeration, Branch- & Bound-Verfahren und Mathematische Programmierung scheitern an den Problemgrößen in realistischen Produktionsumgebungen. Der Rechenaufwand wächst exponentiell mit der Problemgröße. Je mehr Randbedingungen oder Restriktionen zu berücksichtigen sind, desto aufwändiger ist die Suche nach zulässigen Lösungen.

Da reale Ressourcenbelegungsprobleme i.d.R. schwer lösbare Optimierungsprobleme darstellen, müssen in der Praxis meist heuristische Lösungsverfahren eingesetzt werden. Diese lösen das Problem in annehmbaren Laufzeiten, möglicherweise jedoch nicht optimal. Je mehr Rechenzeit zur Verfügung gestellt werden kann, desto besser wird normalerweise das Planungsergebnis. So können insbesondere bei Prioritätsregeln mit einer vorausschauenden Komponente auf das komplette Warteschlangennetzwerk gute Ergebnisse erzielt werden. Ebenso verhält es sich mit heuristischen Branch- & Bound-Verfahren, die so parametrisiert werden können, dass solange nach einer besseren Lösung gesucht wird, bis ein externes Ereignis, z.B. Benutzerabbruch, die Suche abbricht.

Die stochastischen Suchverfahren wie Hill Climber, Tabusuche oder Simulated Annealing können bei einem lokalen Optimum hängen bleiben, auch wenn noch Rechenzeit zur Verfügung steht. Bei genetischen Algorithmen wird der Suchraum breiter analysiert. Die Gefahr an einem lokalen Optimum hängen zu bleiben, ist deshalb reduziert.

Mit den Heuristiken und stochastischen Verfahren kann keine Aussage über die Güte der erzielten Lösung gemacht werden, da das Optimum i.d.R. nicht bekannt ist. Beim Lagrange-Relaxationsverfahren kann dagegen eine Abweichung vom theoretischen Optimum bestimmt werden. Der Einsatz beschränkt sich jedoch auf Probleme mit relativ wenigen Randbedingungen.

	Heur. B&B	Prioritätsregeln	Hill Climber	Tabusuche	Sim. Annealing	Genetische Alg.	Lagrange Rel.
Rechenaufwand	Abhängig von Parametern, niedrig bis hoch	Niedrig, bei Prioritätsregeln mit Prädiktion bis hoher Aufwand	Niedrig	Mittel		Hoch	Mittel
Laufzeit	Abhängig von Parametern kurz bis lang	Kurz, bei Prädiktion auch bis lange Laufzeit	Kurz	Mittel		Lang	
Lösungsgüte	Je nach Aufspannen des Suchraums (Parameter) bis Mittel	Abhängig von benutzter Prioritätsregel und den Zielen	Niedrig	Mittel		Hoch	
Einsatzmöglichkeiten Produktion	Abhängig von Parametern, Kurzfristbereich bis Schichtplanung	Kurzfristbereich, Störungsbehandlung, Online-Planung	Kurzfristbereich	Nicht für Kurzfristbereich		Schichtplanung eher offline, längerfristige Planung	Nur für Optimierungsprobleme mit wenigen Restriktionen
Realisierungsaufwand	Mittel	Niedrig		Mittel		Hoch	
Nachvollziehbarkeit der gefundenen Lösung	Sehr aufwändig nachvollziehbar	Belegungsplan bei einfachen Prio.regeln nachvollziehbar	Nicht nachvollziehbar, stochastische Einflüsse				Nicht nachvollziehbar
Komplexität des Verfahrens	Niedrig			Mittel		Hoch	
Parameteranzahl	Breite, Tiefe der Suche, Suchstrategie	Abhängig von verwendeter Prio. regel, eher niedrig	Wenig	Größe der Tabuliste, Laufzeitschranke	Abkühlfunktion, Laufzeitschranke	Flexible Vorgabe von Zielgewichten möglich, Laufzeitschranke	Hoch
Konfigurationsmöglichkeiten	Vorrangig durch Administrator	Auswahl von Regeln durch Benutzer	Vorrangig durch Administrator			Vorrangig durch Administrator, Zielgewichte durch Benutzer	Vorrangig durch Administrator

Abbildung 5: Merkmale und Einsetzbarkeit der heuristischen Verfahren