3 Bisherige Lösungsansätze

3.1 Lösung in der schulischen Praxis Berlins

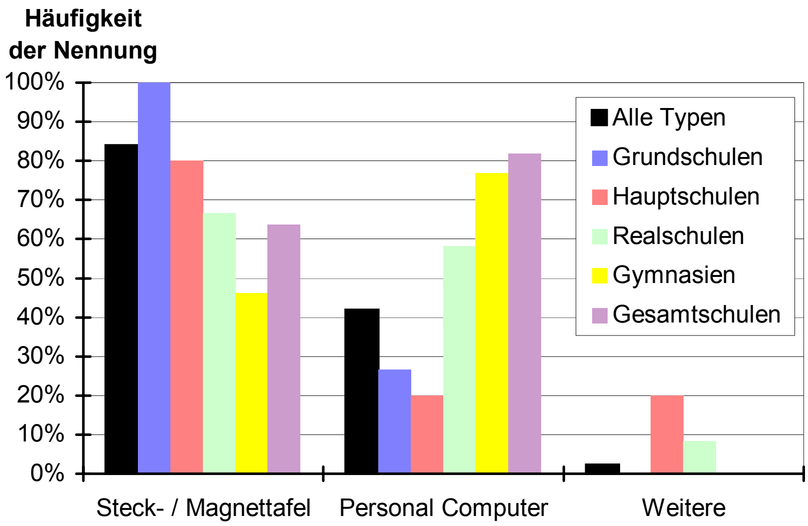
Unter allen Ansätzen, die in der Vergangenheit für die Stundenplansetzung an allgemeinbildenden Schulen entwickelt wurden, gibt es einen, der bisher in jedem Fall zu einem realisierbaren Stundenplan geführt hat: den „praktischen Ansatz“. Sieht man sich diesen jedoch etwas genauer an, kann man feststellen, dass es eine einheitliche Vorgehensweise in der Praxis der Setzung nicht gibt. Zu unterschiedlich sind, wie im vorigen Kapitel dargestellt, die Bedingungen, unter denen die Planerstellung stattfindet. Dennoch ist es in gewissem Umfang möglich, Handlungsmuster zu identifizieren, nach denen die Setzung üblicherweise abläuft. Um Aussagen hierüber treffen zu können, wurden in die in Unterkapitel 2.2 vorgestellte schriftliche Befragung Berliner Stundenplaner vom Winter 1997/98 diesbezügliche Fragen integriert. Die daraus gewonnenen Ergebnisse stellen die Basis der im Folgenden dargestellten Ausführungen dar. Sie werden ergänzt durch zahlreiche Einzelgespräche, die ich im Anschluss an die Befragung mit verschiedenen Stundenplanern geführt habe.

Die nachfolgende Präsentation der Befragungsergebnisse steht unter dem bereits erwähnten Vorbehalt, dass der aus der Befragung erfolgte Rücklauf zuverlässig repräsentative Aussagen nur über die Grundgesamtheit aller allgemeinbildenden Schulen in Berlin zulässt. Demgegenüber sind die auf einzelne Schultypen bezogenen Ergebnisse lediglich als Indizien zu deuten, deren Allgemeingültigkeit aufgrund der jeweils geringen Anzahl auswertbarer Fragebögen nicht gesichert ist.

3.1.1 Planungsmethoden

Für die Durchführung der Setzung kommen in der schulischen Praxis zwei Werkzeuge zur Anwendung: der mit entsprechender Software ausgestattete PC einerseits und die Steck- oder Magnettafel andererseits (vgl. Abbildung 3.1). Dabei stellt die Steck- bzw. Magnettafel das klassische Instrument der handgesteuerten Setzung dar, während der PC je nach benutztem Programm und Neigung des Stundenplansetzers sowohl zur Unterstützung der Handsetzung als auch zur zumindest teilweisen Automatisierung der Setzung verwendet wird. Nur wenige Schulen verwenden neben diesen beiden Instrumenten weitere Hilfsmittel wie speziell für den Setzungszweck entworfene handschriftliche Tabellenblätter oder ähnliche Einsatzformen von Papier und Bleistift.

Abbildung 3.1: Verwendete Hilfsmittel für die Setzung
(Mehrfachnennungen möglich)

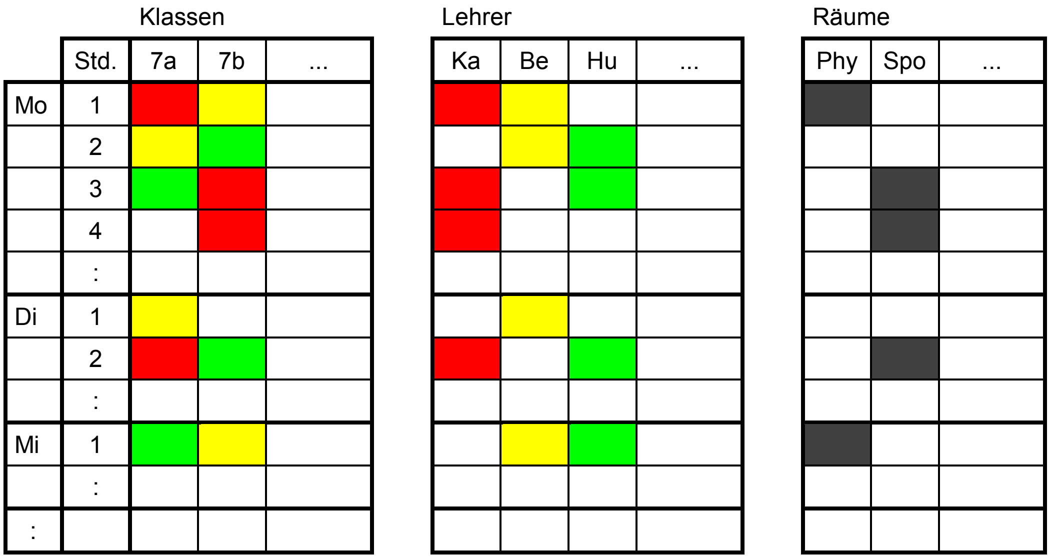


3.1.1.1 Handsetzung mit Hilfe von Steck- und Magnettafeln

Steck- und Magnettafeln sind seit vielen Jahrzehnten in Gebrauch. Auf einer großen Wandfläche ermöglichen sie einen globalen Überblick über sämtliche Belegungen von Klassen, Lehrern und Räumen im Verlaufe der Schulwoche (vgl. Abbildung 3.2). Sie enthalten zumeist, nebeneinander in Matrixform angeordnet, einen Klassen-, einen Lehrer- und einen Raumplan, in denen die einzelnen Klassen, Lehrer und Räume durch Spalten, die verfügbaren Zeiteinheiten durch Zeilen repräsentiert sind. Ist eine Klasse, ein Lehrer oder ein Raum während einer bestimmten Zeiteinheit belegt, so wird dies durch ein Steck- oder Magnetplättchen an der entsprechenden Position in der Matrix kenntlich gemacht.[19]

[19] Eine ausführliche Beschreibung der verschiedenen Darstellungsformen für Stundenpläne und ihrer Umsetzung durch die Stecktafel findet sich bei Krins [1981], S.81-87 u. 91-93.

Abbildung 3.2: Schema einer Magnettafel



Erläuterung:
Eingefärbte Kästchen markieren eine Belegung durch Unterricht. Die im Klassen- und im Lehrerplan verwendeten Farben identifizieren dabei jeweils den Lehrer, der den Unterricht erteilt. Im Raumplan werden üblicherweise keine Farbmarkierungen verwendet. Raumbelegungen sind daher einheitlich dunkelgrau markiert.

Um die Abhängigkeit zwischen Klassen- und Lehrerbelegungen sichtbar zu machen, sind die Plättchen mit verschiedenen Farben versehen, die entweder, wie in Abbildung 3.2, die einzelnen Lehrer oder aber die einzelnen Klassen identifizieren. Den Räumen hingegen werden keine eigenen Farben zugeordnet. Soweit der Stundenplaner die Raumzuordnung nicht im Gedächtnis hat, muss er daher die Plättchen des Klassenplans mit entsprechenden schriftlichen Vermerken versehen. Auch ein Vermerk des jeweils unterrichteten Faches auf den einzelnen Plättchen ist denkbar und durchaus üblich.

Die Handsetzung beruht auf einer sukzessiven Verplanung der einzelnen in der Unterrichtsverteilung gegebenen Unterrichtseinheiten (vgl. Abbildung 3.3). Die Reihenfolge, nach der dabei vorgegangen wird, ist grundsätzlich individuell, doch wenden die meisten Stundenplaner eine Prioritätsregel an, die sich an der zu erwartenden Schwierigkeit orientiert, mit der eine für die jeweilige Unterrichtseinheit geeignete Raum-Zeit-Zuordnung gefunden werden kann. Stehen für eine zu setzende Unterrichtseinheit mehrere zulässige Zuordnungsmöglichkeiten zur Verfügung, wird diejenige gewählt, die den Zielen der Setzung am meisten gerecht wird. Erweist es sich hingegen als unmöglich, eine zulässige Zuordnung zu finden, so werden bereits erzeugte Sitzungen durch räumliche und/oder zeitliche Verschiebung oder Tausch variiert, bis die Einheit gesetzt werden kann. Ist es schließlich gelungen, einen zulässigen Stundenplan zu generieren, so kann versucht werden, ihn durch weitere Tauschoperationen zwischen den erzeugten Sitzungen im Sinne der gesetzten Ziele zu verbessern.

Abbildung 3.3: Typische Vorgehensweise der Handsetzung

Solange noch ungesetzte Unterrichtseinheiten vorhanden sind, **wiederhole**:

> Wähle diejenige ungesetzte Unterrichtseinheit *u*, für die die zu erwartende Schwierigkeit, eine zulässige Raum-Zeit-Zuordnung zu finden, am größten ist
>
> **Für jede** für *u* zu erzeugende Sitzung *s* **wiederhole**:
>
> > **Wenn** zulässige Raum-Zeit-Zuordnung für *s* möglich, **dann**
> >
> > > Wähle unter allen zulässigen Raum-Zeit-Zuordnungen diejenige mit dem höchsten Zielbeitrag und lege *s* entsprechend fest
> >
> > **sonst**
> >
> > > Variiere bereits erzeugte Sitzungen durch Verschiebung oder Tausch von Räumen und/oder Zeiten, bis eine zulässige Raum-Zeit-Zuordnung für *s* möglich ist und lege *s* entsprechend fest
> >
> > **Ende Wenn**
>
> **Ende Schleife**

Ende Schleife

Solange noch Verbesserungen in der Zielerreichung möglich sind, **wiederhole**:

> Variiere die erzeugten Sitzungen durch zulässige Verschiebung oder Tausch von Räumen und/oder Zeiten, bis eine Verbesserung der Zielerreichung realisiert ist

Ende Schleife

Die konkrete Ausgestaltung der Prioritätsregel für die Auswahl der Unterrichtseinheiten ist von Planer zu Planer unterschiedlich. Aus Gesprächen mit Berliner Stundenplanern ergab sich, dass häufig verschiedene Prioritätsklassen gebildet werden, die der Setzer dann nacheinander abarbeitet. Eine beispielhafte, jedoch nicht allgemeinverbindliche Abfolge ist die Setzung von:

1. Unterrichtseinheiten des Kurssystems der gymnasialen Oberstufe, soweit vorhanden und nicht bereits von anderer Instanz räumlich und zeitlich fixiert,

2. Unterrichtseinheiten, die stark genutzte Räume beanspruchen, z.B. Sport- und Schwimmunterricht,

3. Unterrichtseinheiten des Wahlpflichtbereichs der Mittelstufe, die jeweils alle Klassen eines Jahrgangs sowie mehrere Lehrer und Räume binden,

4. weiteren schwierigen Unterrichtseinheiten, z.B. Kopplungen und Teilungen, Randstunden des Religionsunterrichts oder Unterricht von Lehrern, die sich nur an wenigen Tagen in der Woche in der Schule aufhalten,

5. allen übrigen Unterrichtseinheiten.

Wesentliches Charakteristikum der Handsetzung ist die intensive Nutzung der menschlichen Intelligenz. Sie stellt den Hauptvorteil der Handsetzung gegenüber automatisierten Setzverfahren dar. So verschafft sich der Stundenplaner mit Hilfe der Magnettafel einen Überblick über komplexe Problemzusammenhänge, den kein automatisierter Algorithmus im selben Ausmaß realisieren kann. Auch kann er wesentlich kompliziertere Tauschvorgänge handhaben. Darüber hinaus erlaubt es die Handsetzung, das Lösungsverfahren entsprechend den Eingebungen des Setzers spontan zu variieren und auf diese Weise unkonventionelle, durch kein algorithmisches Schema darstellbare Wege zur Problemlösung zu beschreiten. Dabei bleibt die Mehrfachzielsetzung stets in der direkten Kontrolle des Setzers und muss nicht durch quantitative Messgrößen approximiert werden.

3.1.1.2 Setzung mit Hilfe von PC-Programmen

Computerprogramme zur Unterstützung der Setzung existieren bereits seit etwa 40 Jahren, doch haben sie erst mit der Ausbreitung des Personal Computers seit Mitte der 1980er Jahre größere Bedeutung erlangt. Sie bieten zwar wegen der begrenzten Bildschirmgröße nicht dieselbe Übersichtlichkeit wie eine Steck- oder Magnettafel, ermöglichen jedoch eine verbesserte Kontrolle über den Setzungsprozess, da jede neu erzeugte Sitzung automatisch auf ihre Konsistenz mit dem bereits vorhandenen Teilplan überprüft wird. Etwaige Konflikte, etwa wenn ein Lehrer oder ein Raum in einer Periode doppelt verplant ist, werden dem Planer sofort angezeigt, so dass es, im Gegensatz zur Planung mit einer Magnettafel, unmöglich ist, sie zu übersehen. Darüber hinaus bieten PC-Programme umfangreiche Druckfunktionen, die angesichts der Vielzahl zu erstellender individueller Lehrer-, Klassen- und Raumpläne von den Planern sehr geschätzt werden (vgl. Tabelle 3.2).

Die Verbreitung der PC-Anwendung ist erwartungsgemäß dort am größten, wo das Planungsproblem am aufwendigsten erscheint, an Gymnasien und Gesamtschulen (vgl. Abbildung 3.1). Dort verwenden bereits 77% bzw. 78% der Stundenplaner einen PC, während der entsprechende Anteil bei den Grund- und Hauptschulen lediglich 27% bzw. 20% beträgt. Auch die parallele Anwendung von PC und Magnettafel ist durchaus gebräuchlich, da so die Vorteile beider Systeme genutzt werden können.

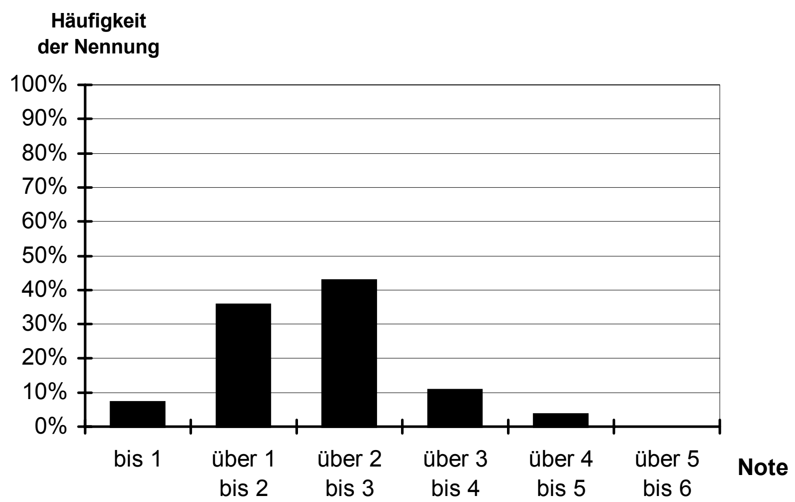
Einen Überblick über die von den befragten Stundenplanern verwendeten PC-Programme für die Setzung gibt Tabelle 3.1. Sie zeigt, dass der Berliner Markt zu mehr als 50% auf nur drei Hersteller konzentriert ist. Alle unter den Nummern 1 bis 7 genannten Stundenplanprogramme verfügen über einen Algorithmus für die automatisierte Setzung. Nicht alle PC-Nutzer ziehen jedoch problemspezifische Stundenplanprogramme zu Rate. Vier von 30 PC-Nutzern verlassen sich auf allgemeine kommerzielle Bürosoftware oder auf ein selbst entwickeltes Programm.

Tabelle 3.1: Für die Setzung verwendete PC-Programme
(Befragung Berliner Stundenplaner vom Winter 1997/98)

Nr.	Programm (Hersteller in Klammern)	Anzahl der Nennungen
1	Curriculum (Stüber Software)	8
2	gp-Untis/gp-Curs (Gruber und Petters)	5
3	Schoolmaster (Aucoteam)	5
4	WinSchule/WinStundenplan (Tillmann)	3
5	Turbo-Planer (Haneke Elektronik)	2
6	SV Paedago (Software für Pädagogik)	1
7	IBS-Planer (Interessenvertretung Berliner Schulleiter e.V.)	1
8	Excel (Microsoft)	1
9	Works (Microsoft)	1
10	Eigenentwickelte Software	2
	nicht identifizierbar	1
	Summe	**30**

Jene Stundenplaner, die einen PC einsetzen, wurden um eine Bewertung des von ihnen verwendeten Programms gebeten. Das Resultat dieser Bewertung, die anhand einer Skala von 1 (= ”sehr gut”) bis 6 (= ”ungenügend”) vorzunehmen war, enthält Abbildung 3.4.

Abbildung 3.4: Benotung für die Setzung verwendeter PC-Programme
(Schulnotensystem; alle PC-nutzenden Schulen, ohne Gewichtung nach Schultypen)



Die meisten Planer erteilten ihrem jeweiligen PC-Programm die Note 3 oder besser. Nur wenige vergaben eine 4, und nur einmal wurde die Leistung des Programms als mangelhaft beurteilt. Demnach sind die Stundenplaner mit den von ihnen benutzten Programmen überwiegend zufrieden. Befragt nach den Vor- und Nachteilen des jeweiligen Programms, hoben die Planer neben den bereits genannten Vorzügen der Kollisionskontrolle und Druckfunktion vor allem die gute Handhabbarkeit sowie, soweit vorhanden, die Verbindung mit einer allgemeinen Schulverwaltungsdatenbank oder einem Vertretungsplanungsmodul positiv hervor (vgl. Tabelle 3.2).

Tabelle 3.2: Kritik der für die Setzung verwendeten PC-Programme
(Anzahl der Nennungen in Klammern)

Positive Wertungen	Negative Wertungen
• Hohe Benutzerfreundlichkeit / gute Handhabung (10)	• Schlechte Setzautomatik / viel manuelle Nacharbeit (9)
• Umfangreiche Druckfunktionen (6)	• Geringe Benutzerfreundlichkeit / schlechte Handhabung (4)
• Kombination mit einem Vertretungsmodul (5)	• Unausgereifte Software / viele Programmabstürze (3)
• Kombination mit einer Datenbank bzw. einem Schulverwaltungsprogramm (3)	• Schlechte Anpassung an individuelle Situation (2)
• Automatische Kollisionskontrolle (2)	• Schnittstellenprobleme mit anderen Programmen (1)
• Möglichkeit der stufenweisen Setzung mit Hilfe der Setzautomatik (2)	• Kein Protokoll manuell durchgeführter Änderungen (1)
• Gute Ergebnisse der Setzautomatik (2)	

Negativ wurde hingegen beurteilt, dass zahlreiche individuelle Besonderheiten der einzelnen Schulen wie komplexe Teilungs- und Kopplungsunterrichtstrukturen durch die Setzautomatik nicht berücksichtigt würden und die von ihr erzeugten Pläne unvollständig und unbrauchbar seien. Darüber hinaus schätzten mehrere Befragte die Benutzerfreundlichkeit ihres Programmes negativ ein und gaben Hinweise zu deren Verbesserung.

Die überwiegend negative Beurteilung der jeweiligen Setzautomatik findet zumindest für die drei meistgenutzten Programme Curriculum, Untis und Schoolmaster ihre Bestätigung in einer vergleichenden Untersuchung von Baumgarten und Müller [1995]. Sie haben anhand praktischer Tests festgestellt, dass die Setzautomatiken dieser Programme nicht in der Lage sind, einen zulässigen Plan zu garantieren, der ohne eine umfangreiche manuelle Nachbesserung realisierbar ist. Als typische, diesen Nachbesserungsbedarf verursachende Mängel automatisch erzeugter Lösungen identifizieren die Autoren die Verletzung der Vollständigkeitsbedingung und/oder des Springstundenverbots für Klassen.

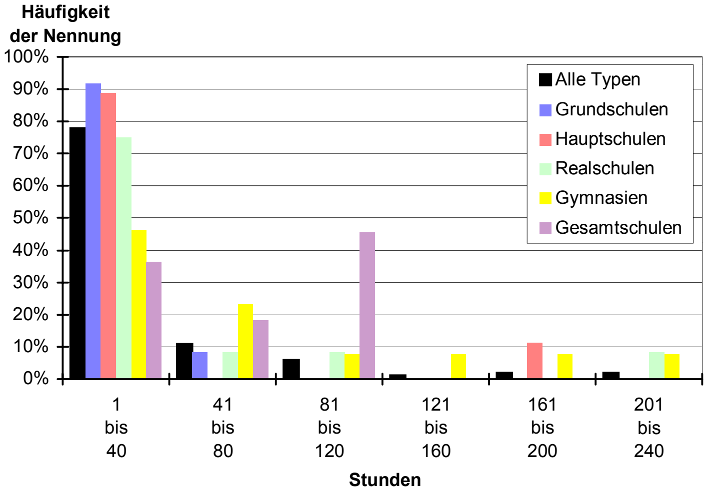
Soweit die in dem jeweiligen Programm implementierte Setzautomatik nicht genutzt wird, entspricht das Vorgehen bei der Setzung am PC im wesentlichen dem der Handsetzung an der Steck- oder Magnettafel. Das Programm trägt lediglich durch die Möglichkeit der automatischen Kollisionskontrolle neu erzeugter Sitzungen und durch seine Datenverwaltungs- und Druckfunktionen zur Arbeitserleichterung bei. Der Einsatz der Setzautomatik hingegen bewirkt für den Stundenplaner eine Verlagerung seines Arbeitsschwerpunktes. Dabei steht die Aufgabe im Mittelpunkt, die durch das Programm erzeugte, i.d.R. unzulässige Lösung durch umfangreiche Verschiebungs- und Tauschoperationen in einen zulässigen und im Sinne der verfolgten Ziele möglichst guten Stundenplan zu verwandeln.

3.1.2 Arbeitsaufwand

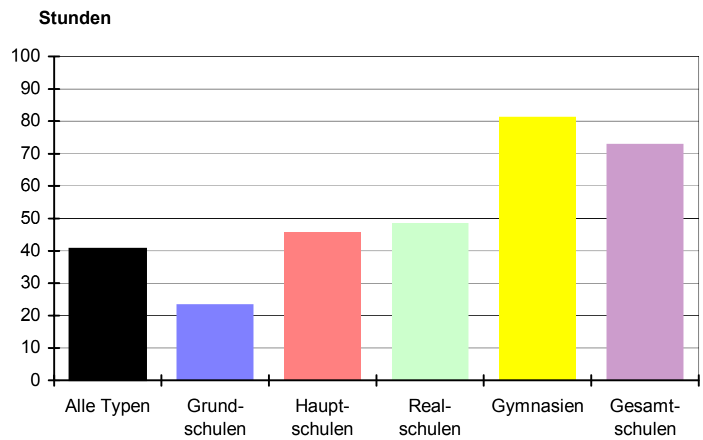
Die Setzung eines neuen Plans verursacht dem mit ihrer Ausführung beauftragten Lehrer i.d.R. einen erheblichen Arbeitsaufwand. Über die Hälfte der Gymnasial- und Gesamtschulplaner setzen dafür mehr als 40 Arbeitsstunden ein (vgl. Abbildung 3.5). Im Durchschnitt beträgt der Aufwand an Gymnasien 81 und an Gesamtschulen 73 Stunden, gegenüber nur 23 Stunden an Grundschulen, 46 an Hauptschulen und 49 an Realschulen. Der Mittelwert für alle Schultypen liegt bei 41 Stunden.

Abbildung 3.5: Arbeitsaufwand für die Setzung eines neuen Stundenplans

Häufigkeiten

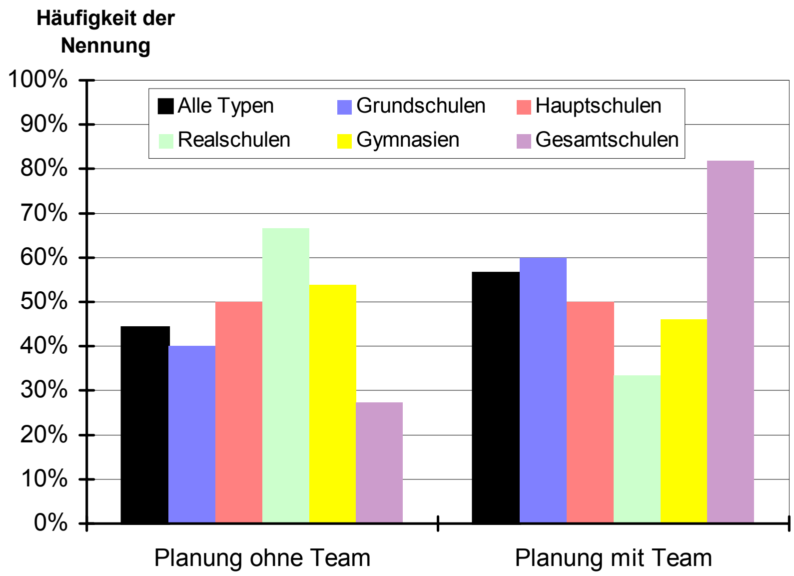


Mittelwerte



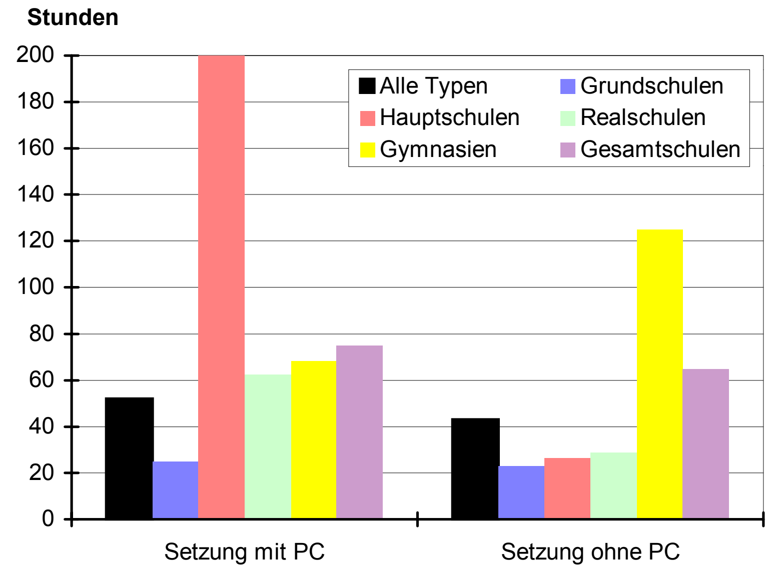
Die hohe Zahl von mehr als 200 Stunden, die dabei für einige Schulen beobachtet wurde, resultiert aus der Tatsache, dass der Stundenplan häufig mit einem Team erstellt wird und entsprechende Personalkapazität bindet. Dabei ist Teamarbeit unter den verschiedenen Schultypen unterschiedlich weit verbreitet (vgl. Abbildung 3.6). Über 80% der Gesamtschulen und 60% der Grundschulen wenden sie an, aber nur 33% der Realschulen. Ein Grund für diese Unterschiede ist nicht erkennbar. Bezogen auf alle Schulen, überwiegt die Teamarbeit knapp die Einzelarbeit.

Abbildung 3.6: Teamarbeit in der Planerstellung



Bemerkenswert ist, dass eine signifikante Zeitersparnis durch den Einsatz von PC-Programmen gegenüber der rein manuellen Setzung anhand der Beobachtungen nicht zu belegen ist. Vergleicht man PC-Nutzer und Nicht-PC-Nutzer unter den Stundenplanern hinsichtlich der für die Setzung aufgewendeten Zeit, so ergibt sich, dass offensichtlich nur Gymnasien eine Beschleunigung durch die Nutzung eines PCs erreichen, alle anderen Schulen hingegen der Setzaufgabe am PC eher mehr Zeit widmen (vgl. Abbildung 3.7).

Abbildung 3.7: Durchschnittlicher Zeitaufwand in Stunden für die Setzung mit und ohne PC



Allerdings ist die Datenbasis für diesen Vergleich teilweise äußerst dünn. Die hohe Stundenzahl für die Setzung mit PC an Hauptschulen beispielsweise beruht nur auf einem einzigen Wert, der möglicherweise als Ausreißer zu interpretieren ist. Dennoch kann insgesamt festgestellt werden, dass PCs als Hilfsmittel gegenüber der herkömmlichen Magnettafel bislang offenbar nur eine äußerst begrenzte Arbeitsersparnis bewirken. Die Feststellung wird untermauert durch einen Vergleich des in Abbildung 3.5 dokumentierten Zeitbedarfs mit der von Krins [1981] (S.209) unternommenen vagen Aufschlüsselung, nach der der Zeitaufwand bei manueller Planerstellung, d.h. ohne Zuhilfenahme eines Computers, an Realschulen und Gymnasien zwischen 45 und 135 Mannstunden beträgt. Offenbar hat die in den letzten Jahren verstärkte Zuhilfenahme von Computern den für die Setzung benötigten Zeitaufwand gegenüber der Situation zu Beginn der 1980er Jahre nicht wesentlich verringert.

3.1.3 Beurteilung

Aufgrund der geringen verfügbaren Information über die Ausgestaltung der in den kommerziellen Stundenplanprogrammen implementierten Setzalgorithmen soll in diesem Unterkapitel lediglich für die Handsetzung eine Verfahrensbeurteilung erfolgen, die für spätere Vergleiche herangezogen werden kann. Ein Informationsverlust ist mit dieser Einschränkung nur insoweit verbunden, wie kommerzielle Ansätze von den durch die Wissenschaft dokumentierten, in Unterkapitel 3.2 eingehend behandelten Verfahren signifikant abweichen, was jedoch unwahrscheinlich ist. Die Beurteilung der Handsetzung ist in Tabelle 3.3 zusammengefasst.

In Bezug auf die bisherige Praxis der manuellen Setzung kann davon ausgegangen werden, dass jede Anforderung, sei sie eine Restriktion oder eine Zielsetzung, so berücksichtigt wird, wie es das Setzungsproblem erfordert. Sofern sich keine schwerwiegenden Veränderungen der schulischen Rahmenbedingungen ergeben, ist daher auch für die Zukunft zu erwarten, dass eine angemessene Beachtung dieser Restriktionen und Zielsetzungen durch den Stundenplaner möglich ist. Die Wahrscheinlichkeit, dass eine zulässige Lösung für das Problem gefunden wird, ist dabei wegen der hohen Genauigkeit der Problemerfassung durch den Stundenplaner und seiner Fähigkeit zur Verfolgung komplexer Suchpfade als sehr hoch anzusehen. Da der Stundenplaner den Handsetzungsprozess selbst lenkt, ist davon auszugehen, dass die zu beachtenden Zielsetzungen in seinem Sinne ausgewogen gehandhabt werden und nicht einem Ziel ein ungerechtfertigtes Übergewicht zukommt.

Tabelle 3.3: Beurteilung des Lösungsansatzes *Handsetzung*
(zur Erläuterung der Kriterien vgl. Tabelle 2.11, S.42)

Kriterium	bisherige Praxis	Potenzial-einschätzung
Berücksichtigung der Restriktionstypen		
• Vollständigkeit (R-01):	ja	möglich
• Kollisionsfreiheit (R-02):	ja	möglich
• Sperrungen (R-03):	ja	möglich
• Springstundenfreiheit für Klassen (R-04):	ja	möglich
• Doppelunterrichtsverbot: (R-05):	ja	möglich
• Randstunden (R-06):	ja	möglich
• Fixierungen (R-07):	ja	möglich
• Wegezeiten (R-09):	ja	möglich
• Freie Tage für Lehrer (R-12):	ja	möglich
Wahrscheinlichkeit einer zulässigen Lösung bzgl. aller o.g. Restriktionstypen:		sehr hoch
Berücksichtigung der Zielsetzungen		
• Gleichlange Klassentage (Z-01):	ja	möglich
• Früher Unterrichtsbeginn (Z-02):	ja	möglich
• Springstundenminimierung für Lehrer (Z-06):	ja	möglich
• Vertretungsbereitschaft (Z-10):	ja	möglich
• Fächerbeziehungen (Z-12):	ja	möglich
Handhabung der Mehrfachzielsetzung:	ausgewogen	ausgewogen
Laufzeit:	sehr lang	sehr lang

Als große Belastung für die Handsetzung erweist sich jedoch die lange Laufzeit von, im Durchschnitt über alle Schultypen, über 40 Stunden (vgl. Abbildung 3.5). Sie ist besonders schwerwiegend, da die Handsetzung – im Gegensatz zu einem computerisierten Verfahren – die ständige Aufmerksamkeit des Stundenplaners erfordert. Der hohe Arbeitsaufwand, der mit der Handsetzung verbunden ist, gibt daher Anlass, nach einer Arbeitserleichterung durch die Automatisierung des Setzungsprozesses zu suchen.

3.2 Lösungsansätze der Wissenschaft

Sowohl für das Problem der Stundenplansetzung an allgemeinbildenden Schulen als auch für die verwandten Probleme der Setzung von Veranstaltungsplänen an Universitäten und der Setzung von Examensplänen haben Wissenschaftler in den vergangenen 40 Jahren zahlreiche Lösungsansätze entwickelt. Ziel war es stets, mit Hilfe der Computertechnologie eine Alternative zur herkömmlichen Handsetzung zu schaffen. Gegenstand dieses Unterkapitels ist es, einen Überblick über die dabei beschrittenen Wege zu geben und, auf Basis der in Unterkapitel 2.4 dargelegten Kriterien, zu beurteilen, welches Potenzial den verschiedenen Ansätzen hinsichtlich einer effektiven Lösung des Setzungsproblems an allgemeinbildenden Schulen innerhalb einer akzeptablen Zeitspanne zukommt. Der Darstellung einzelner Lösungsansätze wird dabei eine Kategorisierung mittels einer Modell- und einer Verfahrenstypologie vorangesetzt. Sie ist bewusst ausführlich gestaltet, um die Identifikation verbindender wie trennender Charakteristika der einzelnen Ansätze zu erleichtern.

3.2.1 Kategorisierung

3.2.1.1 Modelltypologie

Jeder Ansatz zur automatisierten Lösung von Setzungsproblemen stützt sich auf ein Modell.[20] Durch dieses wird das reale Problem in eine künstliche Formulierung abgebildet, die mit einem speziell für diese Formulierung entwickelten Verfahren bearbeitet werden kann. Gelöst wird somit nie das Problem selbst, sondern lediglich das Modell, in das es abgebildet wurde. Es wird dabei im Allgemeinen angenommen, dass eine für das Modell erzielte Lösung auch als Lösung des zugrunde liegenden Problems betrachtet und entsprechend in die Realwelt übertragen werden kann. Modelle, die für die automatisierte Setzung verwendet werden, enthalten drei Kernbestandteile:

1. eine Menge von *Entscheidungsvariablen* mit jeweils gegebenem Wertebereich. Durch die Definition der Entscheidungsvariablen wird abgegrenzt, welche der das Problem betreffenden Regelungen durch das Modell intern zu disponieren sind (endogene Faktoren) und welche zu Beginn des Lösungsprozesses bereits getroffen sein müssen (exogene Faktoren). Der Wertebereich der Entscheidungsvariablen kann diskret oder kontinuierlich sein.

2. eine Menge von *Modellrestriktionen*, die bestimmte Wertekombinationen der Entscheidungsvariablen verbieten und auf diese Weise den Entscheidungsraum des Planers bzw. des Setzverfahrens eingrenzen. Sie können für die Abbildung sowohl von Restriktionen als auch von Zielsetzungen des zugrunde liegenden Problems verwendet werden.

[20] Die Darlegungen dieses Abschnitts beziehen sich speziell auf Modelle, die für die Lösung von Setzungsproblemen entwickelt wurden. Eine allgemeine und umfassende Einführung in die Modellierung betriebswirtschaftlicher Entscheidungsprobleme findet sich bei Bamberg und Coenenberg [2000].

3. eine Menge von *Modellzielsetzungen*, über die eine qualitative Bewertung der nicht durch die Modellrestriktionen verbotenen Wertekombinationen der Entscheidungsvariablen vorgenommen wird. Auch sie können sowohl Zielsetzungen als auch Restriktionen des zugrunde liegenden Problems abbilden. Dabei erfolgt die Abbildung von Problemrestriktionen dadurch, dass Wertekombinationen, die unter diese Restriktionen fallen, eine schlechte Bewertung erhalten. Sie werden jedoch nicht verboten, wodurch eine Entschärfung (Relaxation) der Restriktionen bewirkt wird.

Eine Lösung des Modells liegt vor, wenn jeder Entscheidungsvariablen genau ein Wert aus ihrem Wertebereich zugewiesen ist. Die Lösung ist *zulässig*, wenn durch die in ihr enthaltene Wertekonstellation der Variablen alle Modellrestriktionen eingehalten sind. Sie ist *optimal*, wenn sie zulässig ist und es keine andere zulässige Lösung des Modells gibt, die hinsichtlich mindestens einer Modellzielsetzung einen besseren Wert aufweist. Als *nicht dominiert* gilt eine zulässige Lösung, wenn keine andere zulässige Lösung des Modells existiert, die hinsichtlich aller Modellzielsetzungen einen besseren Wert aufweist. Somit ist jede optimale Lösung zugleich eine nicht dominierte Lösung. Der umgekehrte Schluss gilt hingegen nicht.

Ob eine zulässige oder nicht dominierte Modelllösung tatsächlich als Lösung des Problems in die Realwelt übertragen werden kann, hängt entscheidend von der Abbildungsgenauigkeit des Modells ab. Enthält seine Formulierung nur wenige, einfache Restriktionen, so kann schnell eine zulässige Lösung gefunden werden. Andererseits führt die Auslassung oder großzügige Relaxation wichtiger Problemrestriktionen leicht dazu, dass sich eine Lösung des Modells für das reale Problem als inadäquat erweist. Bei der Beurteilung jedes Modellansatzes ist daher unbedingt zu beachten, dass eine zulässige Lösung des Modells nicht automatisch eine zulässige Lösung des zugrunde liegenden Problems impliziert.

Die für die Setzung von für Schulstunden-, Veranstaltungs- und Examensplänen entwickelten Modelle lassen sich nach ihrer Grundstruktur in vier verschiedene Typen einteilen: logische, graphische, mathematische und naturwissenschaftliche Modelle. Diese sollen im Folgenden kurz beschrieben werden, bevor in Abschnitt 3.2.1.2 eine Verfahrenstypologie der Lösungsansätze vorgestellt wird.

3.2.1.1.1 Logische Modelle

Logische Modelle beinhalten eine Abbildung des Setzungsproblems in Form logisch interpretierbarer Aussagen. Diese können in natürlicher Sprache, mit Hilfe der Notation der Aussagen- und Prädikatenlogik oder in der Syntax einer Programmiersprache oder eines Pseudocodes getroffen werden. Ein in natürlicher Sprache formuliertes Modell der Stundenplansetzung an Schulen lässt sich leicht und mit hoher Abbildungsgenauigkeit aus den in Tabelle 2.11 enthaltenen Restriktionen und Zielsetzungen konstruieren, ist jedoch für die Spezifikation eines Algorithmus‘ wegen der zwangsläufig umständlichen Ausdrucksweise unpraktikabel.

Die Verwendung der Aussagen- und Prädikatenlogik sowie der Syntax spezieller Logischer Programmiersprachen wie PROLOG oder CHIP ist hingegen in der Logischen Programmierung bedeutsam (vgl. van Hentenryck [1991]). Dort werden diskrete Entscheidungsprobleme als Probleme der Restriktionserfüllung (Constraint Satisfaction Problem - CSP) modelliert. Ein CSP besteht dabei aus zwei Komponenten: einer Menge diskreter Entscheidungsvariablen mit jeweils endlichem Wertebereich und einer Menge an Restriktionen, die bestimmte Wertekombinationen der Variablen ausschließen. Letztere werden als logische Implikationen formuliert. Beispielsweise kann durch die Aussageform

$$L(u_1) \cap L(u_2) \neq \varnothing \;\rightarrow\; P(u_1) \cap P(u_2) = \varnothing$$

die Restriktion ausgedrückt werden, dass die Mengen $P(u_1)$ und $P(u_2)$ der Periodenzuordnungen zweier Unterrichtseinheiten u_1 und u_2 sich nicht überschneiden dürfen, wenn sich die Mengen $L(u_1)$ und $L(u_2)$ der jeweils beteiligten Lehrer überschneiden. Betrachtet man jede Periodenzuordnung einer Unterrichtseinheit als eine Modellvariable, dann schließt die obige Implikation aus, dass irgendeine Variable der Unterrichtseinheit u_1 denselben Wert erhält wie irgendeine Variable von u_2. Sie sichert so die Kollisionsfreiheit für die betroffenen Lehrer. Die Aufgabe des Logischen Programms besteht darin, eine Kombination von Werten aller betrachteten Entscheidungsvariablen zu identifizieren, die sämtliche Restriktionen erfüllt. Die Optimierung einer Zielsetzung ist dabei nicht standardmäßig vorgesehen, kann aber mit den Mitteln der Logischen Programmiersprachen durchaus implementiert werden.

Letztlich kann jedes Modell der drei nachfolgend beschriebenen Typen auf ein äquivalentes logisches Modell zurückgeführt werden, was tatsächlich immer dann geschieht, wenn der mit dem Modell verknüpfte Lösungsansatz mit Hilfe einer Programmiersprache als Software implementiert wird. Dennoch haben graphische, mathematische und naturwissenschaftliche Modelle ihre Berechtigung, da sie für zahlreiche Ansätze eine wesentlich anschaulichere Beschreibung ermöglichen, als dies auf der Ebene einer Programmiersprache möglich ist.

3.2.1.1.2 Graphische Modelle

Beginnend mit Kirchgässner [1965] haben zahlreiche Autoren für Setzungsprobleme Lösungsansätze entwickelt, die auf einer Modellierung mit Hilfe von ungerichteten Graphen oder von Netzwerken basieren.[21] Die häufigste, aber keineswegs die einzige dabei gewählte Modellvariante stellen Knotenfärbungsmodelle in ungerichteten Graphen dar, deren Grundmuster aufgrund ihrer hohen Bedeutung im Folgenden beispielhaft beschrieben werden soll. Für die allgemeine Darstellung anderer Ansätze sei an dieser Stelle auf die Beiträge von de Werra [1985a, 1985b und 1996] und Defrenne [1978] verwiesen.

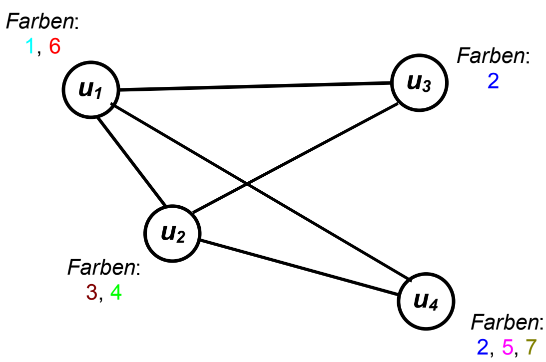
In einem Knotenfärbungsmodell wird das Setzungsproblem durch einen ungerichteten Graphen repräsentiert, der für jede zu setzende Unterrichts- bzw. Prüfungseinheit einen

[21] Eine Einführung in die Graphentheorie mit Definition und Erklärung der im Folgenden verwendeten graphentheoretischen Begriffe geben Dörfler und Mühlbacher [1973].

Knoten enthält. Dürfen zwei Unterrichts- bzw. Prüfungseinheiten wegen einer Ressourcenüberschneidung zeitlich nicht miteinander kollidieren, werden die zu ihnen gehörenden Knoten durch eine Kante verknüpft. Die verfügbaren Perioden des Planungshorizontes werden durch „Farben“ repräsentiert. Aufgabe ist es im einfachsten Fall, jedem Knoten genau eine Farbe so zuzuordnen, dass niemals zwei adjazente Knoten dieselbe Farbe aufweisen (vgl. Carter [1986]). Auf diese Weise wird sichergestellt, dass die Restriktionen der zeitlichen Kollisionsfreiheit eingehalten werden.

Ein vereinfachtes Beispiel für die Anwendung der Knotenfärbung auf das Setzungsproblem an Schulen gibt Abbildung 3.8. In Erweiterung des oben beschriebenen Knotenfärbungsproblems gilt es hier, jedem Knoten genau diejenige Zahl an Farben zuzuweisen, die in der Spezifikation der zugehörigen Unterrichtseinheit gefordert ist. Dabei sind Sitzungen mit einer Dauer von mehr als einer Periode durch Zuordnung einer entsprechenden Zahl aufeinander folgender Farben zu berücksichtigen. Die Anzahl der einem Knoten zuzuweisenden Farben wird in einigen Ansätzen als Knotengewichtung interpretiert (vgl. Boufflet und Nègre [1996], Cangalovic und Schreuder [1991]).

Abbildung 3.8: Modellierung der Stundenplansetzung an Schulen als Knotenfärbungsproblem eines Graphen
(vereinfachtes Beispiel)



	Unterrichtseinheit			
	u_1	u_2	u_3	u_4
Klassen	9c	8a, 8b	9c	8a
Lehrer	Müller	Müller, Kunert	Kunert	Müller
Fächer	Geschichte	Chemie	Chemie	Geschichte
Räume[1]	Klassenr. 9c	Ch. 1, Ch. 2	Ch. 1	Klassenr. 8a
Anz. Sitzungen x Dauer	2 x 1	1 x 2	1 x 1	3 x 1

[1] bereits vorab festgelegt

Die hier dargestellte Lösung eines Setzungsproblems mit vier Unterrichtseinheiten benötigt sieben verschiedene Farben (Perioden). Adjazente Knoten weisen dabei entsprechend der Problemdefinition disjunkte Farbzuordnungen auf.

Die Entscheidungsvariablen des Knotenfärbungsmodells korrespondieren mit den zu färbenden Knoten, wobei für jeden Knoten genau so viele Variablen definiert werden müssen, wie ihm Farben zuzuordnen sind. Sind einem Knoten mehrere aufeinander folgende Farben zuzuweisen, genügt es, nur eine Variable für die Anfangsfarbe zu definieren, sofern durch den Färbungsalgorithmus sichergestellt ist, dass die Mehrstündigkeit der betreffenden Sitzung beachtet wird.

Andere Restriktionstypen als die Vollständigkeit und die zeitliche Kollisionsfreiheit in einem Knotenfärbungsmodell zu berücksichtigen, ist bis auf wenige Ausnahmen ausgesprochen schwierig. De Werra [1985a] zeigt, wie mit Hilfe zusätzlicher Knoten, die jeweils eine Periode des Planungshorizontes repräsentieren, Sperrungen von Ressourcen in das Modell mit aufgenommen werden können. Darüber hinaus können Fixierungen durch Vorabzuweisung der entsprechenden Farben zu den betroffenen Knoten berücksichtigt werden.

Soweit über die Erreichung einer zulässigen Lösung hinaus ein Ziel formuliert wird, besteht es häufig darin, den Graphen so zu färben, dass seine chromatische Zahl, d.h. die unter Beachtung der Zulässigkeit geringstmögliche Farbanzahl, realisiert wird. Dieses Ziel wird in der Examenssetzung verwandt, um den für die Durchführung des Prüfungsprogramms benötigten Zeitraum zu minimieren (vgl. Mehta [1981], Wood [1969], Welsh und Powell [1967]). In anderen Anwendungsfällen hingegen steht von vornherein fest, dass keine konfliktfreie Färbung des Graphen existiert, die mit der in der Realität maximal zulässigen Farbanzahl auskommt. Dies gilt häufig für die Veranstaltungssetzung an Universitäten, die sich mit einer hohen Zahl potenzieller Veranstaltungskonflikte für einzelne Studierende auseinandersetzen muss. Tritt ein solcher Fall ein, wird die Gleichfärbung adjazenter Knoten durch das Modell gestattet, gleichzeitig jedoch das Ziel gesetzt, die Summe der hierdurch entstehenden zeitlichen Kollisionen für Studierende zu minimieren. Dazu wird für jede Kante ein Gewicht bestimmt, das der Anzahl bei Gleichfärbung entstehender Konflikte zwischen den benachbarten Knoten entspricht. Das Lösungsverfahren versucht dann, Gleichfärbungen adjazenter Knoten mit hoher Gewichtung der verbindenden Kante zu vermeiden (vgl. Kiaer und Yellen [1992], Dowsland [1990]).

3.2.1.1.3 Mathematische Modelle

Neben der logischen oder graphischen Modellbildung besteht die Möglichkeit, Setzungsprobleme mit Hilfe von Ungleichungssystemen mathematisch abzubilden und mit entsprechender Software zu lösen. Hierzu hat die Forschung in den Bereichen der Linearen und (Gemischt-)Ganzzahligen, aber auch der Nicht-linearen Programmierung über die vergangenen Jahrzehnte einen umfangreichen Katalog möglicher Modellierungsvarianten entwickelt (vgl. Zimmermann [1992], Meyer und Hansen [1996], Domschke und Drexl [1998]). Exemplarisch soll an dieser Stelle der Ansatz von Zehnder [1965] dargestellt werden, der als erster eine Formulierung des Setzungsproblems allgemeinbildender Schulen als 0/1-Optimierungsmodell entwickelt hat.

Seien mit F, K, L, S und Z die jeweiligen Anzahlen aller Fächer, Klassen, Lehrer, Stunden und Zimmer (Räume) bezeichnet. Seien ferner Binärvariablen $x_{fklsz} \in \{0, 1\}$ mit folgenden Wertinterpretationen definiert:

$$x_{fklsz} := \begin{cases} 1 & \text{falls Fach } f \text{ in Klasse } k \text{ durch Lehrer } l \text{ zur Stunde } s \text{ in Zimmer } z \text{ unterrichtet wird} \\ 0 & \text{sonst} \end{cases} ; \qquad \forall\ f = 1, .., F,\ k = 1, .., K,\ l = 1, .., L,\ s = 1, .., S,\ z = 1, .., Z$$

Bezeichne ferner p_{fk} für alle Fach-Klasse-Kombinationen die Stundenzahl, mit der Klasse k gemäß Unterrichtsverteilung in Fach f zu unterrichten ist, sowie w_{fkl} für alle Fach-Klasse-Lehrer-Kombinationen die Stundenzahl, mit der Klasse k gemäß Unterrichtsverteilung in Fach f durch Lehrer l zu unterrichten ist. Dann lässt sich mit Zehnder das Setzungsproblem anhand folgender Modellrestriktionen beschreiben:

$$\sum_{f=1}^{F}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{fklsz} \leq 1 \ ; \qquad \forall\ z = 1, .., Z;\ s = 1, .., S \qquad \text{(Zehnder 1)}$$

$$\sum_{f=1}^{F}\sum_{k=1}^{K}\sum_{z=1}^{Z} x_{fklsz} \leq 1 \ ; \qquad \forall\ l = 1, .., L;\ s = 1, .., S \qquad \text{(Zehnder 2)}$$

$$\sum_{f=1}^{F}\sum_{l=1}^{L}\sum_{z=1}^{Z} x_{fklsz} \leq 1 \ ; \qquad \forall\ k = 1, .., K;\ s = 1, .., S \qquad \text{(Zehnder 3)}$$

$$\sum_{l=1}^{L}\sum_{s=1}^{S}\sum_{z=1}^{Z} x_{fklsz} = p_{fk} \ ; \qquad \forall\ f = 1, .., F;\ k = 1, .., K \qquad \text{(Zehnder 4)}$$

$$\sum_{s=1}^{S}\sum_{z=1}^{Z} x_{fklsz} = w_{fkl} \ ; \qquad \forall\ f = 1, .., F;\ k = 1, .., K;\ l = 1, .., L \qquad \text{(Zehnder 5)}$$

Die Gleichungen (Zehnder 5) bilden die Vollständigkeitsrestriktionen des Setzungsproblems ab. Sie verlangen, dass für jede Kombination von Fach, Klasse und Lehrer genau so viele Variablen auf 1 gesetzt werden, wie Unterrichtsstunden durch die Unterrichtsverteilung vorbestimmt sind. Mit den Gleichungen (Zehnder 4) wird für jede Klasse die Erfüllung des durch den Lehrplan für jedes Fach vorgegebenen Stundenumfangs durch alle Lehrer gesichert. Die Ungleichungen (Zehnder 1) bis (Zehnder 3) bilden die Restriktionen der zeitlichen Kollisionsfreiheit der Belegung von Räumen, Klassen und Lehrern ab.

Zusätzlich zu den o.g. Restriktionen unterstellt Zehnder das Vorliegen einer eindimensionalen Zielfunktion, die er als lineare Funktion der Variablen charakterisiert, jedoch nicht näher spezifiziert. Die Aufgabe des Setzverfahrens besteht darin, jeder Variablen x_{fklsz} einen Wert aus $\{0, 1\}$ so zuzuweisen, dass einerseits sämtliche Gleichungen und Ungleichungen (= Restriktionen) des Modells erfüllt sind, andererseits die Zielfunktion einen möglichst guten Wert aufweist.

Wie aus der Formulierung deutlich wird, bildet das Modell unmittelbar nur Vollständigkeits- und Kollisionsfreiheitsrestriktionen ab. Sperrungen von Räumen, Lehrern oder Klassen können durch Setzung der rechten Seite der entsprechenden Restriktion in (Zehnder 1) bis (Zehnder 3) auf den Wert 0 sichergestellt werden. Weitere Restriktionen wie das Doppelunterrichtsverbot oder das Springstundenverbot für Klassen bleiben jedoch unberücksichtigt, ebenso mehrstündige Sitzungen sowie Kopplungen und Teilungen. Die Restriktionen (Zehnder 4) werden von Zehnder in das Modell aufgenommen, obwohl sie infolge der strengeren Restriktionen (Zehnder 5) redundant sind, sofern man unterstellt, dass bei korrekter Aufstellung der Unterrichtsverteilung stets

$$\sum_{l=1}^{L} w_{fkl} = p_{fk} \ ; \ \ \forall f = 1, .., F; k = 1, .., K$$

gelten muss. Unter dieser Voraussetzung ist (Zehnder 4) für jede beliebige Kombination eines Faches f mit einer Klasse k identisch mit der Summe über alle Lehrer der f und k betreffenden Restriktionen in (Zehnder 5). Auch kann unter dieser Prämisse die Zahl der Dimensionen ohne Veränderung des Lösungsraumes durch Verschmelzung von Fach-Klasse-Lehrer-Kombinationen zu Unterrichtseinheiten auf drei reduziert werden, da die gegenseitige Zuordnung von Fächern, Lehrern und Klassen bereits durch die Unterrichtsverteilung bzw. die Restriktionen (Zehnder 5) fixiert und daher nicht Gegenstand des hiesigen Entscheidungsproblems ist.

Trotz seiner offensichtlichen Mängel in der Abbildungsgenauigkeit des Problems kann das Modell von Zehnder als Pionierleistung aufgefasst werden, die zahlreiche Wissenschaftler – direkt oder indirekt – auf die Möglichkeit des Einsatzes von Methoden der Mathematischen Programmierung für die Lösung von Setzungsproblemen aufmerksam gemacht hat.

Obwohl die meisten Ansätze die Beziehungen zwischen den Modellvariablen sowohl in der Zielfunktion als auch in den Restriktionen linear abbilden, sind die Möglichkeiten der mathematischen Modellierung nicht auf lineare Zusammenhänge beschränkt. Ferland und Roy [1985] etwa bilden ein Teilproblem der Veranstaltungssetzung an Universitäten als quadratisches Zuordnungsproblem ab. Boronico [2000] verwendet ein Modell, in dem einige der Restriktionen kubisch formuliert sind.

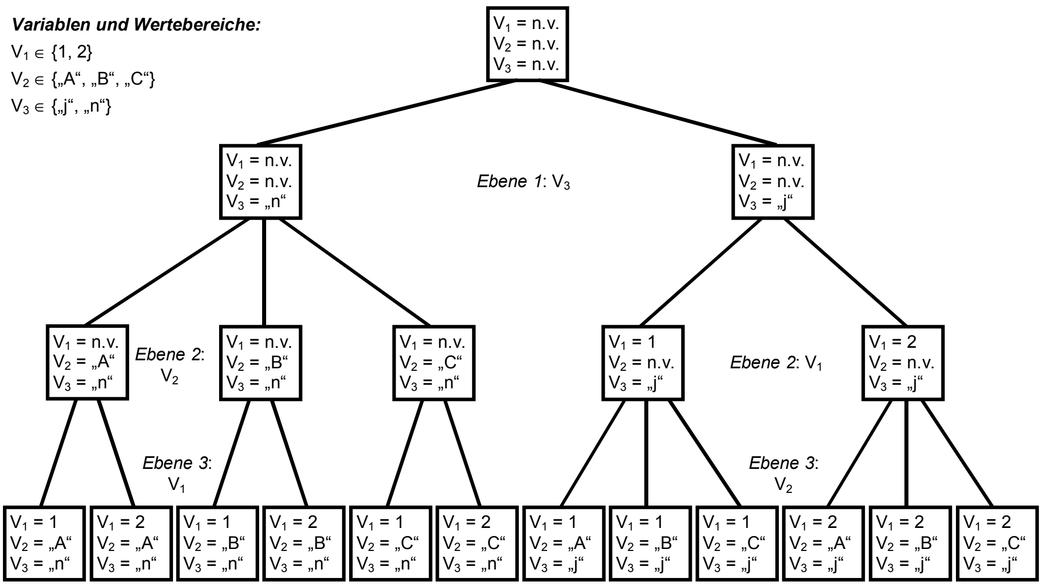
3.2.1.1.4 Naturwissenschaftliche Modelle

Eine Reihe von Autoren hat zur Abbildung von Setzungsproblemen Modelle definiert, die weniger auf logischen oder mathematischen Formulierungen als auf naturwissenschaftlichen Analogien beruhen. Diese Modelle lassen sich in zwei Kategorien einteilen: genetische Modelle und neuronale Modelle. Die erste Kategorie ist stark an biologischen Vererbungsvorgängen orientiert, die zweite an Struktur und Funktionsweise von Nervensystemen und Gehirnen. Da beide Modellvarianten eng an spezielle algorithmische Schemata gekoppelt sind, werden sie erst an entsprechender Stelle gemeinsam mit den auf ihnen beruhenden Verfahrenstypen Genetischer Algorithmus (vgl. Abschnitt 3.2.4.4) und Neuronale Netzwerke (vgl. Abschnitt 3.2.4.5) näher erläutert.

3.2.1.2 Verfahrenstypologie

Beinahe jedes existierende Verfahren zur Lösung eines Setzungsproblems kann als Verfahren zur Untersuchung von Entscheidungsbäumen beschrieben werden, obwohl generell durchaus andere Ansätze zur Lösung diskreter Entscheidungsprobleme existieren.[22] Ein allgemeines Beispiel für einen solchen Entscheidungsbaum gibt Abbildung 3.9. Der oberste Knoten des Entscheidungsbaumes bildet den Startpunkt der Analyse. Er wird daher im Folgenden als Startknoten bezeichnet. Jeder Knoten unterhalb des Startknotens korrespondiert mit der Entscheidung über den Wert einer einzelnen diskreten Variable, jeder Endknoten auf der untersten Ebene des Baumes mit genau einer Lösung des Modells. Der Aufbau des Baumes ist variabel. Er hängt davon ab, in welcher Reihenfolge die Variablen gesetzt werden, wobei für verschiedene Zweige unterschiedliche Reihenfolgen gewählt werden können. Zumindest die Endknoten, häufig aber auch vorgelagerte Knoten, unterliegen einer ein- oder mehrdimensionalen Bewertung, die einen Maßstab für ihre Zulässigkeit und/oder ihren tatsächlichen (bei Endknoten) oder ihren im besten Falle zu erwartenden (bei vorgelagerten Knoten) Zielbeitrag darstellt.

Abbildung 3.9: Beispiel eines dreistufigen Entscheidungsbaumes



Die Größe des Entscheidungsbaumes wächst in Abhängigkeit von der Zahl der zu betrachtenden Variablen enorm schnell. Für ein, gemessen an der Dimension praktischer Setzprobleme, unrealistisch einfaches Beispiel mit $n = 50$ Entscheidungsvariablen mit jeweils nur zwei möglichen Werten existieren bereits $2^{50} = 1.125.899.906.842.620$ Endknoten. Es ist daher für reale Problemgrößen unmöglich, den vollständigen Baum zu erzeugen und alle Endknoten zu bewerten. Die Herausforderung an jeden Lösungsansatz des Setzungsproblems besteht deshalb darin, auf Basis einer geschickten Konstruktion nur weniger Pfade des Baumes gute oder sogar optimale bzw. nicht dominierte Lösungen zu erzeugen.

[22] Als Beispiel seien hier die bei Zimmermann [1992], S.127, erwähnten Schnittebenenverfahren auf Basis nicht-ganzzahliger Lösungen für die ganzzahlige Optimierung genannt, die im Bereich der Setzung aufgrund ihrer im Vergleich etwa zu Branch-and-Bound-Verfahren geringeren Effizienz keine Bedeutung haben.

Die Menge der für das Setzungsproblem an Schulen oder für verwandte Probleme entwickelten Lösungsverfahren lässt sich anhand folgender wichtiger, alternativ oder ergänzend anzuwendender Unterscheidungen kategorisieren:

1. *Exakte vs. heuristische Verfahren* (vgl. Bardadym [1996], Reeves [1996], Müller-Merbach [1976]): Exakte Verfahren zeichnen sich dadurch aus, dass sie das Auffinden einer optimalen zulässigen Lösung des Modells garantieren, sofern eine solche Lösung existiert. Nicht garantiert ist jedoch, dass eine optimale oder auch nur zulässige Lösung in vertretbarer Zeit gefunden wird, da im schlechtesten Fall sämtliche Endknoten des Entscheidungsbaumes erzeugt und bewertet werden müssen (vgl. Unterkapitel 2.3). Zwar ist die Wahrscheinlichkeit für das Eintreten dieses Falles äußerst gering, doch gelten exakte Verfahren für Anwendungen mit komplexer Struktur und hoher Variablenzahl häufig als zu langsam. Heuristische Verfahren garantieren demgegenüber zwar weder das Erreichen einer optimalen noch einer zulässigen Lösung, erfüllen jedoch, je nach Verfahren mehr oder weniger weitreichend, die Eigenschaft, dass sie *in der Regel* schnell eine gute bis sehr gute Lösung präsentieren können. Allerdings kann die Frage, wie gut eine von einer Heuristik erzeugte Lösung tatsächlich ist, zumeist nicht beantwortet werden, da das bestmögliche Ausmaß der Zielerreichung unter gleichzeitiger Wahrung der Zulässigkeit nur über ein exaktes Verfahren ermittelt werden kann.
2. *Konstruktive (Eröffnungs-) vs. iterative Verfahren*: Diese Unterscheidung wird von Müller-Merbach [1976] nur für heuristische Verfahren getroffen, kann jedoch verallgemeinert werden. Konstruktive Verfahren erzeugen einen Teil des Entscheidungsbaums, indem sie, beginnend am Startknoten, durch stufenweise Auswahl und Wertzuweisung jeweils einer Variablen aus bereits generierten Knoten neue Knoten ableiten. Ziel ist es, einen zulässigen Endknoten zu erreichen, wobei häufig zugleich eine möglichst gute Zielbewertung der gefundenen Lösung angestrebt wird. Iterative Verfahren setzen demgegenüber voraus, dass bereits ein oder mehrere mit Hilfe eines konstruktiven Verfahrens erzeugte zulässige Endknoten vorliegen. Diese werden durch Variation einzelner oder mehrerer Variablenzuweisungen in neue Endknoten des Entscheidungsbaumes, sog. Nachbarlösungen, überführt in der Hoffnung, zulässige Lösungen mit verbesserter, möglichst guter Zielbewertung identifizieren zu können.

Die hier aufgezeigten Gliederungskriterien stellen mögliche Grobklassifikationen auf oberster Ebene dar. Für nachgelagerte Ebenen bieten Müller-Merbach [1976] allgemein und Bardadym [1996] bezogen auf Verfahren der Stundenplansetzung an Schulen und Universitäten detailliertere Aufstellungen, die sich an Entwurfsentscheidungen innerhalb der Verfahrensobertypen orientieren. Dabei bezieht Bardadym neben Heuristiken und exakten Verfahren auch interaktive Verfahren in seine Betrachtung mit ein, die im wesentlichen Funktionalitäten zur Unterstützung der Handsetzung bzw. der manuellen Nachbearbeitung automatisch erzeugter Pläne beinhalten. Sie werden in diesem Unterkapitel nicht weiter betrachtet, da hier der Fokus auf den algorithmischen Kern automatisierter Verfahren gerichtet ist, während die Computerunterstützung manueller Prozesse bereits Gegenstand des Unterkapitels 3.1 war.

Unter den oben genannten Kategorisierungen eignet sich die Unterscheidung zwischen konstruktiven und iterativen Lösungsverfahren besonders gut für die Darstellung grundsätzlicher algorithmischer Vorgehensweisen. Sie soll daher im Folgenden als Haupt-Verfahrenstypologie verwendet werden.

Um die spätere Darstellung einzelner Lösungsansätze zu erleichtern, werden die Aufbauprinzipien konstruktiver und iterativer Verfahren zur Untersuchung von Entscheidungsbäumen in den folgenden beiden Unterabschnitten näher erläutert. Dabei wird auf wesentliche jeweils zu treffende Entwurfsentscheidungen eingegangen. Der Einfachheit halber wird für die Darstellung unterstellt, dass die innerhalb des jeweiligen Modells betrachtete Zielfunktion eindimensional ist, was der Praxis aller mir bekannten automatisierten Setzverfahren entspricht. Eine Erweiterung auf mehrdimensionale Zielgrößen und die Betrachtung nicht dominierter statt optimaler Lösungen ist jedoch leicht möglich. Die Darstellung ist auf den Minimierungsfall ausgelegt, auf den jedes Maximierungsproblem durch Multiplikation der jeweiligen Zielfunktion mit (-1) zurückgeführt werden kann.

Um Missverständnisse zu vermeiden, sei an dieser Stelle betont, dass sich die in den beiden folgenden Unterabschnitten präsentierten Schemata konstruktiver und iterativer Lösungsverfahren in der Darstellungsweise nicht bzw. nur teilweise mit den Beschreibungen decken, die die Autoren der verschiedenen Einzelansätze jeweils liefern. Sie sind daher nicht als Zusammenfassungen dieser Ansätze zu verstehen, sondern vielmehr als typische Grundmuster, auf die sich die jeweiligen Algorithmen durch entsprechende Anpassung der Formulierung zurückführen lassen.

3.2.1.2.1 Konstruktive Verfahren

Folgende formale Notation soll für die Beschreibung der generellen Vorgehensweise konstruktiver Verfahren zugrundegelegt werden:

V	:=	Menge aller diskreten Entscheidungsvariablen des betrachteten Modells	
W_v	:=	Wertebereich der Variablen v;	$\forall\, v \in V$
$w(v)$	:=	zugewiesener Wert der Variablen v;	$\forall\, v \in V$
K	:=	Menge erzeugter Knoten des Entscheidungsbaumes	
$e(k)$	:=	Ebene des Knotens k innerhalb des Baumes, wobei $e(k) \in \{0, 1, ..., \lvert V\rvert\}$;	$\forall\, k \in K$
$mk(k)$	:=	Mutterknoten (= direkt vorgelagerter Knoten) des Knotens k;	$\forall\, k \in K$: $e(k) > 0$
GV_k	:=	Menge am Knoten k bereits gesetzter Variablen;	$\forall\, k \in K$

EK_k := Menge aller unterhalb des Knotens k liegenden Endknoten des Entscheidungsbaumes; $\forall\, k \in K$

$w_k(v)$:= Wert der Variablen v am Knoten k; $\forall\, k \in K,\, v \in GV_k$

$nv(k)$:= vom Knoten k ausgehend als nächste zu setzende Variable $v \in V \setminus GV_k$; $\forall\, k \in K$

AW_k^{nv} := Menge bereits abgearbeiteter Werte von $nv(k)$; $\forall\, k \in K$

$f(k)$:= Bewertung des Knotens k, wobei

$$f(k) = \begin{cases} \text{für jeden } k \text{ nachgelagerten Endknoten minimale erreichbare Zielbewertung,} & \text{falls k zulässig und } e(k) < |V| \\ \text{tatsächliche Zielbewertung,} & \text{falls k zulässig und } e(k) = |V| \\ M\ (= \text{Platzhalter für } +\infty) & \text{falls } k \text{ unzulässig} \end{cases} ; \qquad \forall\, k \in K$$

k_{best} := bester bisher gefundener zulässiger Endknoten

f_{best} := $f(k_{best})$

OK := Menge offener Knoten $k \in K$

Der schematische Ablauf konstruktiver Verfahren ist in Abbildung 3.10 dargestellt. Er beruht auf der beim Startknoten beginnenden sukzessiven Generierung und Bewertung neuer Knoten des Entscheidungsbaumes, bis entweder für das Verfahren kein zulässiger Endknoten mit verbessertem Zielbeitrag mehr erreichbar ist oder ein anderes Abbruchkriterium erreicht wird.

Abbildung 3.10: Ablaufschema konstruktiver Lösungsverfahren

Definiere neuen Knoten sk: *** *Generiere Startknoten*
$e(sk) := 0$
$GV_{sk} := \emptyset$
$nv(sk) :=$ nicht definiert
$K := \{sk\}$
Berechne $f(sk)$ *** *Analysiere Startknoten*
Wenn $f(sk) = M$, **dann** *** *Startknoten unzulässig*
  $k_{best} :=$ nicht definiert
  $f_{best} := M$
  $OK := \emptyset$
sonst wenn $ek_{best} \in EK_{sk}: f(ek_{best}) \leq f(ek),\ \forall\ ek \in EK_{sk}$ direkt identifizierbar, **dann**
  $K := K \cup \{ek_{best}\}$ *** *Optimale Lösung bereits identifizierbar*
  $k_{best} := ek_{best}$
  $f_{best} := f(ek_{best})$
  $OK := \emptyset$
sonst *** *Modell noch nicht gelöst*
  $k_{best} :=$ nicht definiert
  $f_{best} := M$
  $OK := \{sk\}$
Ende Wenn
Solange $OK \neq \emptyset$ **und** kein verfahrensspezifisches Abbruchkriterium erfüllt, **wiederhole**
  Wähle $k \in OK$ *** *Entscheidung 1: Knotenwahl*
  Wenn $nv(k) =$ nicht definiert, **dann**
    Wähle $nv(k) \in V \setminus GV_k$ *** *Entscheidung 2: Variablenwahl*
    $AW_k^{nv} := \emptyset$
  Ende Wenn
  Wähle $w(nv(k)) \in W_{nv(k)} \setminus AW_k^{nv}$ *** *Entscheidung 3: Wertewahl*
  Definiere neuen Knoten nk: *** *Neuer Knoten*
  $e(nk) := e(k) + 1$
  $mk(nk) := k$
  $GV_{nk} := GV_k \cup \{nv(k)\}$
  Für alle $i \in GV_{nk} \setminus \{nv(k)\}$ **wiederhole**
    $w_{nk}(i) := w_k(i)$
  Ende Schleife
  $w_{nk}(nv(k)) := w(nv(k))$
  $K := K \cup \{nk\}$
  Berechne $f(nk)$ *** *Analysiere neuen Knoten*
  Wenn $f(nk) \geq f_{best}$, **dann** *** *Neuer Knoten unzulässig oder zu schlecht*
    keine Aktion
  sonst wenn $e(nk) = |V|$, **dann** *** *Neuer Knoten = neue beste Lösung*
    $k_{best} := nk$
    $f_{best} := f(nk)$
    $OK := OK \setminus \{i \in OK: f(i) \geq f_{best}\}$ *** *Dominierte offene Knoten entfernen*
  sonst wenn $ek_{best} \in EK_{nk}: f(ek_{best}) \leq f(ek),\ \forall\ ek \in EK_{nk}$ direkt identifizierbar, **dann**
    $K := K \cup \{ek_{best}\}$ *** *Nachgelagerter Endknoten = neue beste Lösung*
    $k_{best} := ek_{best}$
    $f_{best} := f(ek_{best})$
    $OK := OK \setminus \{i \in OK: f(i) \geq f_{best}\}$ *** *Dominierte offene Knoten entfernen*
  sonst *** *Neuer Knoten noch nicht abschließend beurteilbar*
    $nv(nk) :=$ nicht definiert
    $OK := OK \cup \{nk\}$
  Ende Wenn
  $AW_k^{nv} := AW_k^{nv} \cup \{w(nv(k))\}$ *** *Aktualisiere Information über alten Knoten*
  Wenn $AW_k^{nv} = W_{nv(k)}$ **oder** nochmalige Betrachtung von k nicht vorgesehen, **dann**
    $OK := OK \setminus \{k\}$
  Ende Wenn
Ende Schleife

Die Gestaltungsmöglichkeiten konstruktiver Verfahren sind unbegrenzt. Sie beinhalten die Regelung folgender zentraler Entwurfsentscheidungen:

1. Strategie der *Knotenwahl*: Welcher erzeugte, aber noch nicht vollständig abgearbeitete Knoten k soll in der jeweils nächsten Verfahrensiteration betrachtet werden?

2. Strategie der *Variablenwahl*: Welche Variable soll, ausgehend von dem gerade betrachteten Knoten k, als nächste gesetzt werden, um Folgeknoten für k zu generieren?

3. Strategie der *Wertewahl*: Welcher Wert soll der unter 2. ausgewählten Variablen an dem neu zu erzeugenden Knoten nk zugewiesen werden?

4. *Bewertungsfunktion*: Nach welcher Vorschrift soll der neu erzeugte Knoten nk bewertet werden?

5. *Abbruchkriterium*: Wann soll ggf. der Iterationsprozess vor der Abarbeitung aller offenen Knoten ($OK = \emptyset$) abgebrochen werden?

Mit Hilfe dieser fünf Entwurfsentscheidungen lassen sich Kategorisierungen innerhalb der konstruktiven Lösungsverfahren bilden. Eine allgemeine, detaillierte Morphologie von Entwurfsentscheidungen findet sich bei Müller-Merbach [1976]. An dieser Stelle soll hingegen nur auf wenige, für die weitere Betrachtung wichtige zentrale Unterscheidungen und Sonderfälle eingegangen werden.

Exakte vs. heuristische konstruktive Verfahren: Exakte Verfahren lassen sich innerhalb der Kategorie der konstruktiven Verfahren anhand zweier Eigenschaften identifizieren. Zum ersten arbeiten sie jeden einmal erzeugten Knoten k vollständig ab, d.h. sie entfernen ihn erst dann aus der Menge OK der offenen Knoten, wenn Nachfolgeknoten für alle möglichen Werte der Variable $nv(k)$ erzeugt und analysiert worden sind ($AW_k^{nv} = W_{nv(k)}$). Zum zweiten terminieren sie prinzipiell erst dann, wenn OK leer und so der Beweis für die Optimalität der besten gefundenen Lösung oder, falls keine zulässige Lösung gefunden wurde, für die Unzulässigkeit des Modells erbracht ist. Ist eine dieser beiden Eigenschaften nicht erfüllt, so liegt ein heuristischer Ansatz vor.

Such- vs. Optimierungsverfahren: Über die Bewertungsfunktion lässt sich steuern, ob das Modell als Such- oder als Optimierungsaufgabe gelöst werden soll. Eine Suchaufgabe ist gelöst, sobald die erste im Sinne des Modells zulässige Lösung gefunden ist. Eine Optimierungsaufgabe hingegen ist erst dann vollständig gelöst, wenn die gefundene zulässige Lösung den bestmöglichen Zielbeitrag aufweist. Viele Modelle nutzen die Bewertungsfunktion zur Relaxation des behandelten Problems, indem sie originäre Problemrestriktionen als Ziele formulieren. In einem solchen Fall erfolgt die Lösung zwar als Optimierungsaufgabe, doch wird statt oder neben den Zielkriterien des zugrunde liegenden Problems die Distanz der Modelllösung zur realen Zulässigkeit minimiert (vgl. de Werra [1985a]). Demgegenüber sind Bewertungsfunktionen für die Lösung von Suchaufgaben lediglich binär:

$$f(k) = \begin{cases} 0 & \text{falls } k \text{ zulässig} \\ M & \text{falls } k \text{ unzulässig} \end{cases} \quad ; \qquad \forall\, k \in K$$

Prioritätsregelverfahren vs. Vorausschauregelverfahren (vgl. Müller-Merbach [1976], Krins [1981]): Diese Unterscheidung bezieht sich auf die Strategien der Variablen- und der Wertewahl. Während Prioritätsregelverfahren eine bereits vor dem Start des Algorithmus‘ anhand vordefinierter Kriterien erzeugte Abarbeitungsreihenfolge der Variablen und/oder Werte umsetzen, berechnen Vorausschauregelverfahren während des Verfahrensablaufs an jedem betrachteten Knoten für jede potenziell zu wählende Variable bzw. ihren Variablenwert eine Dringlichkeitskennzahl und nehmen anhand dieser Kennzahl die Auswahl vor. Die Bezeichnung „Vorausschauregelverfahren“ für dieses Vorgehen orientiert sich an der gängigen Praxis, die Dringlichkeitskennzahl als Prognoseinstrument für die Wahrscheinlichkeit des Erreichens einer zulässigen Lösung oder für die Güte der zu erwartenden Lösung zu verwenden.

Tauschvorgänge: Viele konstruktive Heuristiken beinhalten Tauschregeln, nach denen, ausgehend von einem unzulässigen erzeugten Knoten k, ein neuer Knoten nk mit gegenüber k identischer Variablenauswahl $GV_{nk} = GV_k$, aber variierten Variablenwerten zur Analyse bestimmt wird. Die Erzeugung von nk lässt sich im Rahmen des in Abbildung 3.10 präsentierten Ablaufschemas in folgender Weise darstellen: Nach der Analyse von k wird zunächst der auf dem Pfad p_{nk} zwischen dem Startknoten und nk tiefstgelegene offene Knoten $ok(p_{nk}) \in OK$ ausgewählt. Anschließend werden sukzessive alle Knoten erzeugt, die erforderlich sind, um von $ok(p_{nk})$ aus den Knoten nk zu erreichen, der sich auf derselben Ebene wie k befindet. Alle Knoten i nach $ok(p_{nk})$ und vor nk erhalten die Bewertung $f(i) = -M$ und werden nach Setzung der jeweils nächsten Variable sofort wieder aus der Menge OK der offenen Knoten eliminiert. Der Knoten nk hingegen erfährt die für das Verfahren außerhalb von Tauschvorgängen vorgesehene Bewertung und wird bei Vorliegen der entsprechenden Voraussetzungen bis zu seiner Abarbeitung in die Menge OK aufgenommen.

Identifikation eines optimalen Endknotens von vorgelagerter Ebene aus: Bestimmte Verfahren wie das Branch-and-Bound-Verfahren der (Gemischt-)Ganzzahligen Programmierung (vgl. Abschnitt 3.2.2.2) beinhalten die Möglichkeit, dass im Rahmen der Bewertung eines Knotens k mit $e(k) < |V|$ festgestellt wird, welcher k nachgelagerte Endknoten $ek \in EK_k$ zulässig ist und unter allen in EK_k enthaltenen Knoten die beste Zielbewertung aufweist. Ist dies der Fall und stellt ek eine Verbesserung gegenüber der bisher besten gefundenen Lösung k_{best} dar, so kann ek direkt als neue beste Lösung gespeichert werden, ohne dass k in die Menge der offenen Knoten aufgenommen und der Entscheidungsbaumbereich unterhalb von k weiter untersucht werden muss.

3.2.1.2.2 Iterative Verfahren

Iterative Verfahren unterscheiden sich von konstruktiven Verfahren dadurch, dass bei der Analyse des Entscheidungsbaumes nur Endknoten betrachtet werden, d.h. solche Knoten, an denen jeder Entscheidungsvariable v ein Wert aus ihrem Wertebereich W_v zugewiesen ist. Entsprechend wird vorausgesetzt, dass vor der Ausführung eines iterativen Verfahrens bereits mindestens ein zulässiger Endknoten mit Hilfe eines konstruktiven Verfahrens erzeugt wurde. Für die formale Beschreibung iterativer Verfahren soll folgende Notation verwendet werden:

EK := Menge aller erzeugten Endknoten des Entscheidungsbaumes

$f(k)$:= Bewertung des Knotens k, wobei

$$f(k) = \begin{cases} M \text{ (= Platzhalter für } +\infty) & \text{falls } k \text{ unzulässig} \\ \text{Zielbeitrag von } k, & \text{falls } k \text{ zulässig} \end{cases}; \qquad \forall\, k \in EK$$

k_{best} := bester bisher gefundener zulässiger Endknoten

f_{best} := $f(k_{best})$

OK := Menge offener Knoten $k \in EK$

OK_{alt} := Zwischenspeicherung der Menge OK

N_{OK} := Menge aller mit Hilfe eines oder mehrerer verfahrensspezifischer Transformationsoperatoren aus den in OK enthaltenen offenen Knoten generierbaren neuen Endknoten des Entscheidungsbaumes (sog. Nachbarschaft)

GN := Menge generierter Nachbarlösungen $n \in N_{OK}$

$AnzGN$:= Maximale Anzahl der Elemente in GN

Der schematische Ablauf iterativer Verfahren ist in Abbildung 3.11 dargestellt. Zentrales Verfahrensprinzip ist dabei die „Nachbarschaftssuche“ (vgl. Reeves [1996], Müller-Merbach [1976]), d.h. die Erzeugung neuer Lösungen („Nachbarn“) durch Operationen auf bereits vorhandenen Lösungen. Der Prozess terminiert, wenn ein vorgegebenes Abbruchkriterium, etwa eine Maximalzahl von Iterationen oder eine maximale Laufzeit, erreicht wird.

Abbildung 3.11: Ablaufschema iterativer Lösungsverfahren

```
EK := {mit konstruktivem Verfahren erzeugte zulässige Endknoten des Entscheidungsbaums}
OK := EK                                    *** Menge der Startlösungen
k_best := k ∈ OK: f(k) ≤ f(i), ∀ i ∈ OK
f_best := f(k_best)
Solange kein verfahrensspezifisches Abbruchkriterium erfüllt, wiederhole
    GN := ∅
    Wähle AnzGN ≤ |N_OK \ EK|
    Solange |GN| < AnzGN, wiederhole        *** Erzeuge neue Nachbarknoten
        Generiere mit Hilfe der vorgesehenen verfahrensspezifischen
        Operatoren einen neuen Endknoten nk ∈ N_OK \ EK
        GN := GN ∪ {nk}
        EK := EK ∪ {nk}
        Berechne f(nk)                      *** Bewerte Knoten
    Ende Schleife
    OK_alt := OK
    Solange OK ∩ OK_alt ≠ ∅ und GN ≠ ∅, wiederhole
        Wähle nk ∈ GN: f(nk) = min {f(nk‘): nk‘ ∈ GN}   *** Wähle erzeugten neuen Knoten nk
        GN := GN \ {nk}
        Wenn f(nk) = M, dann                *** Neuer Knoten unzulässig
            keine Aktion
        sonst wenn f(nk) ≥ f_best, dann     *** Neuer Knoten schlechter als beste Lösung
            Wenn Entscheidung für Akzeptanz von nk, dann
                Wähle k ∈ OK ∩ OK_alt       *** Auswahl und Ersatz eines alten Knotens
                OK := OK \ {k}
                OK := OK ∪ {nk}
            sonst
                keine Aktion
            Ende Wenn
        sonst                               *** Neuer Knoten = neue beste Lösung
            k_best := nk
            f_best := f(nk)
            Wähle k ∈ OK ∩ OK_alt           *** Auswahl und Ersatz eines alten Knotens
            OK := OK \ {k}
            OK := OK ∪ {nk}
        Ende Wenn
    Ende Schleife
Ende Schleife
```

Auch der Gestaltung iterativer Lösungsverfahren ist theoretisch keine Grenze gesetzt. Sie beinhaltet die Regelung der Entwurfsentscheidungen:

1. *Mächtigkeit von OK*: Wieviele offene Knoten sollen für die Erzeugung neuer Knoten vorrätig gehalten werden?

2. *Nachbarschaftsdefinition*: Nach welcher Vorschrift sollen neue Knoten aus den in *OK* enthaltenen Lösungen generiert werden?

3. *Mächtigkeit von GN*: Wieviele Nachbarlösungen $AnzGN \leq |N_{OK}|$ sollen für jede Zusammensetzung von *OK* erzeugt werden?

4. *Bewertungsfunktion*: Nach welcher Vorschrift sollen neu erzeugte Knoten bewertet werden?

5. *Akzeptanzregel*: Nach welcher Vorschrift soll die Entscheidung getroffen werden, ob ein neu erzeugter Knoten *nk*, dessen Zielbewertung schlechter als die der besten bisher gefundenen Lösung k_{best} ist, in die Menge *OK* der offenen Knoten aufgenommen wird?

6. *Löschungsregel*: Nach welcher Vorschrift soll der aus der Menge *OK* zu löschende Knoten *k* bestimmt werden, wenn der neu generierte Knoten *nk* in *OK* aufzunehmen ist?

7. *Abbruchkriterium*: Wann soll der Iterationsprozess abgebrochen werden?

Einige der hier aufgeführten Entwurfsentscheidungen werden erst in jüngerer Zeit genutzt. So war es in den 1970er Jahren noch üblich, stets nur einen offenen Knoten *k* zu betrachten ($|OK| = 1$), aus der erzeugten Nachbarschaft *GN* den Knoten *nk* mit der besten Zielbewertung auszuwählen und *k* in *OK* genau dann durch *nk* zu ersetzen, wenn $f(nk) < f_{best} = f(k)$ war (vgl. Müller-Merbach [1976]). Zu regelnde Entscheidungen waren daher lediglich die Nachbarschaftsdefinition, die Zahl der je Verfahrensiteration zu erzeugenden Nachbarlösungen, die Bewertungsfunktion und das Abbruchkriterium für das Verfahren. Neuere Lösungsansätze wie Genetische Algorithmen, Simulated Annealing oder Tabu Search hingegen kennen auch die Möglichkeit, mehr als einen offenen Knoten zur Erzeugung von Nachbarn heranzuziehen und auch solche neuen Lösungen zur weiteren Betrachtung zu akzeptieren, die eine schlechtere Zielbewertung als der beste zuvor gefundene Knoten aufweisen (vgl. Reeves [1996]).

Alle iterativen Lösungsverfahren, mit Ausnahme der praktisch bedeutungslosen vollständigen Enumeration des gesamten Entscheidungsbaumes, sind heuristischer Natur. Es wird, da bereits die Startlösungen im Sinne des Modells zulässig sind, immer die Optimierung einer Zielfunktion angestrebt. Diese wird allerdings, ähnlich wie in konstruktiven Verfahren, häufig zur Problemrelaxation verwendet, so dass originäre Zielsetzungen des betrachteten Problems ganz oder teilweise in den Hintergrund gedrängt werden.

Im Gegensatz zu den meisten konstruktiven Verfahren greifen zahlreiche, gerade jüngere, iterative Lösungsansätze bei der Formulierung verfahrensspezifischer Entscheidungsregeln auf die Möglichkeit einer zufallsbasierten Alternativwahl zurück. Stochastische Ansätze werden dabei insbesondere für die Nachbarschaftsdefinition, aber auch für Akzeptanzentscheidungen genutzt (vgl. Reeves [1996]).

3.2.1.3 Übersicht

Jeder Ansatz für die automatisierte Lösung von Setzungsproblemen kann durch den ihm zugrunde gelegten Modelltyp und den verwendeten Verfahrenstyp klassifiziert werden. Da es jedoch mehrere Modelltypen gibt, die eine gleichwertige Abbildung des Problems erlauben, ist es letztlich der Verfahrenstyp, der Modellbildung und algorithmische Gestaltung eines Lösungsansatzes dominiert. Aus diesem Grund werden die in den nachfolgenden Abschnitten dokumentierten einzelnen Lösungsansätze in erster Linie nach dem jeweils eingesetzten Verfahrenstyp eingeordnet und charakterisiert. Nur für solche Verfahrenstypen, die sich alternativ mehrerer Modelltypen bedienen können, wird eine Feingliederung nach diesen Modelltypen vorgenommen. Einen Überblick der für Setzungsprobleme gebräuchlichen Verfahrenstypen mit Bezugnahme auf den bzw. die jeweils korrespondierenden Modelltypen gibt Tabelle 3.4.

Tabelle 3.4: Verfahrenstypen für die automatisierte Lösung von Setzungsproblemen

verwendeter Modelltyp	konstruktive Verfahrenstypen		iterative Verfahrenstypen
	exakt	heuristisch	heuristisch
graphisch	• exakte Graphenfärbung	• engpassorientierte Zuordnung • andere	
logisch	• exakte Logische Programmierung	• engpassorientierte Zuordnung • heuristische Logische Programmierung • andere	• Hillclimbing • Tabu Search • Simulated Annealing
mathematisch	• Lineare Programmierung • (Gemischt-)Ganzzahlige Programmierung • Lagrange-Relaxation • Goal Programming • Nicht-lineare Programmierung		
naturwissenschaftlich			• Genetische Algorithmen • Neuronale Netze

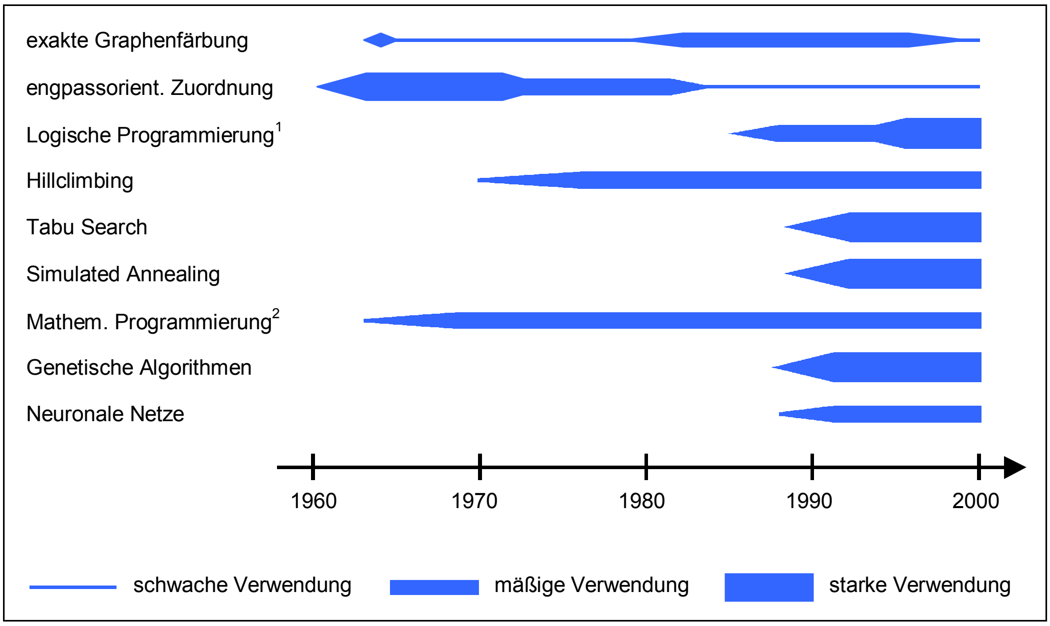
Der Überblick über bisher entwickelter Ansätze zur Lösung von Setzungsproblemen, welcher in den verbleibenden Abschnitten dieses Unterkapitels gegeben wird, erfolgt speziell unter dem Blickwinkel der Fragestellung, inwieweit den Verfahrenstypen, die diesen Ansätzen zugrunde liegen, ein Potenzial für die erfolgreiche Lösung des in Kapitel 2 beschriebenen Setzungsproblems an allgemeinbildenden Schulen innewohnt. Er ist aufgrund dieser Fragestellung von anderen zusammenfassenden Darstellungen zu unterscheiden, die teilweise bereits lange vor dieser Studie erschienen sind.

Ein erster Überblick früher automatisierter Setzverfahren findet sich bereits bei Stahlknecht [1964]. Ebenfalls aus der Anfangszeit der automatisierten Setzung stammen die Zusammenstellungen von Junginger [1968] sowie Bosler und Frangos [1974]. Mit einer vielbeachteten, umfangreichen Bibliographie der bis zum Ende der 1970er Jahre erschienenen Aufsätze schufen Schmidt und Ströhlein [1980] eine Basis für die Forschungsarbeit der folgenden Jahrzehnte. Sie wird ergänzt durch Beschreibungen des jeweils aktuellen Forschungsstandes von Junginger [1982] und Carter [1986], von denen letzterer sich auf Ansätze für die Lösung von Examenssetzungsproblemen konzentriert. Den Fortschritt der nachfolgenden zehn Jahre dokumentieren Carter und Laporte [1996] mit Bezug auf die

Examensproblematik sowie Carter und Laporte [1998] für den Universitätsbereich. Über aktuelle Forschungsrichtungen zum Ende der 1990er Jahre geben zudem die Übersichten von Bardadym [1996] und Burke et al. [1997] Auskunft. Schließlich präsentiert Schaerf [1999] eine aktuelle, umfassende Zusammenfassung von Ansätzen der automatisierten Setzung im Schul-, Universitäts- und Examensbereich.

Die Verwendung der in Tabelle 3.4 aufgelisteten Verfahrenstypen hat weder für alle Typen zur selben Zeit begonnen, noch war und ist sie für alle Typen gleich stark ausgeprägt. Einen groben Überblick über die historische Entwicklung der verschiedenen Typen gibt Abbildung 3.12. Daraus lässt sich erkennen, dass seit Beginn der Forschung in diesem Bereich zu Anfang der 1960er Jahre nicht nur die Breite des Typensortiments, sondern auch die Forschungsaktivität insgesamt stark zugenommen hat. Dies gilt insbesondere für die 1990er Jahre, in denen mehrere neuartige iterative Heuristiktypen erstmalig für die Lösung von Setzungsproblemen adaptiert wurden. Der ohnehin von Beginn an vorherrschende Trend zur Bevorzugung heuristischer gegenüber exakten Verfahrenstypen hat sich durch diese Entwicklung deutlich verstärkt.

Abbildung 3.12: Historische Entwicklung der Verfahrenstypen für die automatisierte Lösung von Setzungsproblemen



[1] Aufteilung exakte vs. heuristische Ansätze etwa 1 : 1
[2] Lineare / (Gemischt-)Ganzzahlige / Nicht-lineare Programmierung / Lagrange-Relaxation / Goal Programming

Wie sehen nun Ansätze der exakten Graphenfärbung, der heuristischen Logischen Programmierung oder des Tabu Search im Setzungsbereich konkret aus und welches Potenzial kann den verschiedenen Verfahrenstypen für die Lösung des in Kapitel 2 charakterisierten Problems der Stundenplansetzung an allgemeinbildenden Schulen jeweils zugeschrieben werden? Diese Frage ist Gegenstand der folgenden Abschnitte.

3.2.2 Ansätze auf Basis exakter konstruktiver Verfahrenstypen

3.2.2.1 Exakte Graphenfärbung

Exakte Verfahren auf Basis von Knotenfärbungsmodellen in Graphen wurden von Boufflet und Nègre [1996] für die Setzung von Examensplänen und von Dowsland [1990] für die Setzung von Veranstaltungsplänen an Universitäten präsentiert. Auf beide Anwendungsbereiche bezieht sich das Verfahren von Kirchgässner [1965]. Cangalovic und Schreuder [1991] beschreiben ein Verfahren, das anwendungsunabhängig für die Setzung von Veranstaltungen variierender Länge konzipiert ist, dessen sinnvolle Verwendung jedoch aufgrund der verfolgten Zielsetzung einer Minimierung des Planungshorizontes am ehesten in der Examensplanung zu sehen ist. Ein explizit auf die Setzung von Schulstundenplänen bezogener exakter Graphenfärbungsansatz wird in der Literatur nicht berichtet.

Die Ansätze von Kirchgässner und Dowsland wurden für Fälle entwickelt, in denen eine für alle Prüflinge bzw. Studierenden konfliktfreie Färbung des Problemgraphen bei vorgegebener Maximalzahl P zu verwendender Farben bzw. Perioden nicht möglich ist. Beide Verfahren verwenden daher eine Kantengewichtung zur Messung potenzieller Konflikte bei Gleichfärbung adjazenter Knoten (vgl. Abschnitt 3.2.1.1.2). Hierauf aufbauend, ermittelt das Verfahren von Kirchgässner einen Plan, indem es die Kantenmenge des Problemgraphen reduziert, bis seine chromatische Zahl, also die Anzahl der für seine Färbung mindestens benötigten Farben, P nicht mehr überschreitet. Hierzu werden alle vollständigen und kritischen Teilgraphen des Problemgraphen identifiziert, die mehr als P Knoten aufweisen, und durch Entfernung jeweils einer Kante zerstört. Das Verfahren minimiert dabei die Summe der Bewertungen aller entfernten Kanten.

Das Verfahren von Dowsland hingegen entfernt zunächst alle Kanten, deren Gewichtung einen vorgegebenen Wert g nicht überschreitet, und erzeugt anschließend eine die Farbanzahl minimierende Lösung des verbleibenden Problemgraphen. Ist die Farbzahl der Lösung größer als P, wird g erhöht, und es wird ein erneuter Färbungsversuch mit abermals reduziertem Graphen gestartet. Der Prozess wird so lange fortgeführt, bis die Zahl der in der Lösung verwendeten Farben P nicht mehr überschreitet. Das Verfahren minimiert so die maximale Zahl entstehender Konflikte für Studierende bei zeitlicher Parallelisierung zweier Lehrveranstaltungen.[23]

Auch das Verfahren von Boufflet und Nègre erzeugt eine Lösung für einen beschränkten Planungshorizont P, doch wird hier von der Existenz einer konfliktfreien Lösung ausgegangen. Zwar wird durch die Wertewahl für die jeweils gerade zu setzende Variable versucht, eine Lösung mit möglichst guter Raumauslastung im Primärkriterium und möglichst wenigen Prüflingen mit drei Prüfungen am selben Tag im Sekundärkriterium zu erzeugen, doch bricht das Verfahren ab, sobald die erste zulässige Lösung gefunden ist. Der Ansatz von Cangalovic und Schreuder hingegen minimiert die Zahl der verwendeten Farben, ohne dass hierfür eine Obergrenze unterstellt wird.

---

[23] Alternativ zu dieser Zielsetzung stellt Dowsland eine Modifikation des Verfahrens vor, mit der statt des maximalen Konfliktes die Summe aller Konflikte für Studierende minimiert werden kann.

Allen genannten Ansätzen ist gemein, dass sie, bedingt durch ihren jeweiligen Anwendungsbereich, nur einen kleinen Teil der für die Setzung an Schulen gültigen Restriktionstypen erfassen. Keines der Verfahren verfolgt eine der für den Schulbereich relevanten Zielsetzungen. Nur Dowsland und Boufflet und Nègre haben ihre Verfahren an realen Problemen getestet, deren Dimensionsgrößen jedoch unterhalb denen eines durchschnittlichen Schulsetzungsproblems anzusiedeln sind. Cangalovic und Schreuder präsentieren Ergebnisse nur für zufallsgenerierte Problemgraphen mit maximal 40 Knoten, Kirchgässner liefert gar nur ein Beispiel mit zehn Graphenknoten.

3.2.2.2 Lineare, Ganzzahlige und Gemischt-ganzzahlige Programmierung

Ansätze, die mit Hilfe mathematischer Modelle der Linearen, Ganzzahligen oder Gemischt-ganzzahligen Programmierung Stundenpläne für allgemeinbildende Schulen zu setzen versuchen, sind ausgesprochen rar. Zwar wurden bereits früh entsprechende Modelle wie das in Abschnitt 3.2.1.1.3 beschriebene Modell von Zehnder formuliert, doch dienten diese ausschließlich der Problemdarstellung oder der Analyse von Zulässigkeitsbedingungen (vgl. Zehnder [1965], Junginger [1972 und 1986], Krins [1981]). Eine Lösung realer Problemstellungen wurde unter Verweis auf die zu erwartende Modellgröße und der daraus gefolgerten unakzeptablen Laufzeiten stets verworfen. Die einzige Ausnahme in dieser Hinsicht stammt von Lawrie [1968 und 1969], der bereits am Ende der 1960er Jahre im Rahmen eines Dekompositionsansatzes ein 0/1-Modell für die zeitliche Parallelisierung von Unterrichtslayoutmustern präsentierte, das, gelöst mit Hilfe des Gomory-Schnittebenenverfahrens (vgl. Zimmermann [1992], S.127ff.), erfolgreich auf ein reales, wenn auch in seiner Struktur stark vom deutschen System abweichendes Schulproblem angewendet wurde.

Auch aus jüngerer Zeit existieren kaum Ansätze, Schulsetzungsprobleme über mathematische Optimierungsmodelle zu lösen. Drexl und Salewski [1997] definieren ein einstufig zu lösendes Totalmodell, welches Ressourcenkapazitäten und Zeitabstandsrestriktionen zwischen den zu erzeugenden Sitzungen berücksichtigt und als Besonderheit die endogene Festlegung der Anzahl und Dauer der Sitzungen für jede Unterrichtseinheit über einen jeweils zu wählenden Unterrichtsmodus beinhaltet. Gelöst werden mit diesem Modell und der Standard-Optimierungssoftware LINDO (vgl. Fourer [1999]) jedoch lediglich kleine simulierte Beispielfälle, um einen Referenz-Zielfunktionswert für die Qualitätsbeurteilung heuristischer Verfahren zu erzeugen.

Der bislang einzige Totalansatz, mit dessen Hilfe es gelungen ist, zulässige und hinsichtlich des angewendeten Modellziels optimale Pläne für praxisnahe Fälle zu erzeugen, stammt von Birbas, Daskalaki und Housos [1997]. Er basiert auf folgendem 0/1-Optimierungsmodell:

Gegeben seien die Mengen I aller Wochentage, J aller Perioden eines Tages, K aller Klassen, L aller Lehrer und M aller Fächer (courses). Sei mit $j1$ die Anzahl der Perioden je Wochentag bezeichnet ($j1 := |J|$). Ferner seien definiert:

M_{kl}	:=	$\{m \in M$: m wird durch Lehrer l in Klasse k unterrichtet$\}$;	$\forall\ k \in K, l \in L$
K_l	:=	$\{k \in K$: Klasse k wird durch Lehrer l unterrichtet$\}$;	$\forall\ l \in L$
L_k	:=	$\{l \in L$: Klasse k wird durch Lehrer l unterrichtet$\}$;	$\forall\ k \in K$
a_k	:=	Gesamtzahl aller Perioden, in denen Klasse k zu unterrichten ist;	$\forall\ k \in K$
b_{kl}	:=	Gesamtzahl aller Perioden, in denen Klasse k von Lehrer l zu unterrichten ist;	$\forall\ k \in K, l \in L$
d_{mk}	:=	Gesamtzahl aller Perioden, in denen Klasse k in Fach m zu unterrichten ist;	$\forall\ k \in K, m \in M$
c_{ijklm}	:=	Bewertung für den Fall, dass Fach m in Klasse k durch Lehrer l in Periode j des Tages i unterrichtet wird;	$\forall\ i \in I, j \in J, k \in K, l \in L, m \in M_{kl}$

Die Modellvariablen $x_{ijklm} \in \{0; 1\}$ seien definiert durch:

$$x_{ijklm} := \begin{cases} 1 & \text{Klasse } k \text{ wird durch Lehrer } l \text{ in Stunde } j \text{ des Tages } i \text{ in Fach } m \text{ unterrichtet} \\ 0 & \text{sonst} \end{cases}; \quad \forall\ i \in I, j \in J, k \in K, l \in L, m \in M_{kl}$$

Dann lassen sich Zielfunktion und Restriktionen des Setzungsmodells mit Birbas, Daskalaki und Housos wie folgt formulieren:

$$\text{minimiere} \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} \sum_{k \in K_l} \sum_{m \in M_{kl}} c_{ijklm} \cdot x_{ijklm} \qquad \text{(Birbas 0)}$$

u.d.N.

$$\sum_{k \in K_l} \sum_{m \in M_{kl}} x_{ijklm} \leq 1 \ ; \qquad \forall\ i \in I, j \in J, l \in L \qquad \text{(Birbas 1)}$$

$$\sum_{l \in L_k} \sum_{m \in M_{kl}} x_{ijklm} = 1 \ ; \qquad \forall\ i \in I, j = 1, .., j1 - 1, k \in K \qquad \text{(Birbas 2a)}$$

$$\sum_{l \in L_k} \sum_{m \in M_{kl}} x_{ijklm} \leq 1 \ ; \qquad \forall\ i \in I, j = j1, k \in K \qquad \text{(Birbas 2b)}$$

$$\sum_{l \in L_k} \sum_{j \in J} \sum_{m \in M_{kl}} x_{ijklm} \leq j1 \ ; \qquad \forall\ i \in I, k \in K \qquad \text{(Birbas 3)}$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{l\in L_k}\sum_{m\in M_{kl}} x_{ijklm} = a_k \quad ; \qquad \forall\, k \in K \qquad \text{(Birbas 4)}$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{m\in M_{kl}} x_{ijklm} = b_{kl} \quad ; \qquad \forall\, k \in K, l \in L \qquad \text{(Birbas 5)}$$

$$\sum_{j\in J}\sum_{k\in K_l}\sum_{m\in M_{kl}} x_{ijklm} \geq 1 \quad ; \qquad \forall\, i \in I, l \in L \qquad \text{(Birbas 6)}$$

$$\sum_{j\in J} x_{ijklm} \leq 1 \quad ; \qquad \forall\, i \in I, l \in L, k \in K_l, m \in M_{kl} \qquad \text{(Birbas 7)}$$ [24]

$$\sum_{i\in I}\sum_{j\in J} x_{ijklm} = d_{mk} \quad ; \qquad \forall\, l \in L, k \in K_l, m \in M_{kl} \qquad \text{(Birbas 8)}$$

Die Restriktionen (Birbas 1) stellen die zeitliche Kollisionsfreiheit für Lehrer sicher, die Restriktionen (Birbas 2a) und (Birbas 2b) diejenige der Klassen. Durch die Gleichheitsrestriktionen (Birbas 2a) wird zudem bestimmt, dass jede Klasse an jedem Tag in jeder Stunde außer der letzten unterrichtet werden *muss*. Sie sichern so einerseits das Springstundenverbot, andererseits die gleichmäßige Länge aller Unterrichtstage für jede Klasse. Vorausgesetzt wird dabei, dass die Stundenzahl jeder Klasse ausreicht, um den Plan an jedem Tag bis mindestens zur vorletzten Periode aufzufüllen. Die Restriktionen (Birbas 3) gewährleisten, dass für jede Klasse für jeden Tag nicht mehr Sitzungen eingeplant werden, als Perioden zur Verfügung stehen. Sie werden von Birbas, Daskalaki und Housos in das Modell aufgenommen, obwohl sie von den Restriktionen (Birbas 1) bis (Birbas 2b) linear abhängig und damit redundant sind.

Die Vollständigkeit des Stundenplans für jede Klasse wird durch die Restriktionen (Birbas 4) hinsichtlich der insgesamt vorgesehenen Stundenzahl und durch (Birbas 5) hinsichtlich der Anzahl der mit jedem Lehrer zu verbringenden Stunden gesichert. Die Restriktionen (Birbas 6) verlangen, dass jeder Lehrer an jedem Wochentag mindestens eine Periode lang unterrichtet. Durch (Birbas 7) wird für jedes Fach in jeder Klasse das Doppelunterrichtsverbot am selben Tag erzwungen. Wiederum die Vollständigkeit des Planes betreffen die Restriktionen (Birbas 8), die sicherstellen, dass jedes Fach mit der für sie vorgesehenen Periodenzahl im Plan berücksichtigt wird. Durch (Birbas 8) werden die Restriktionstypen (Birbas 4) und (Birbas 5) redundant, da sie von (Birbas 8) linear abhängig sind.

Neben den durch die o.g. Modellrestriktionen repräsentierten Problemrestriktionen Vollständigkeit, Kollisionsfreiheit, Springstundenverbot und Doppelunterrichtsverbot werden auch Sperrungen berücksichtigt, indem alle Variablen, die die gesperrte Klasse bzw. den gesperrten Lehrer oder Raum und die Periode(n) der Sperrung betreffen, auf den Wert 0 fixiert werden. Manuelle Fixierungen von Sitzungen zu bestimmten Zeiten werden von

[24] Die Restriktionen (Birbas 7) sind für den Fall formuliert, dass jedes Fach m in maximal fünf einperiodigen Sitzungen unterrichtet wird. Sind mehr als fünf Sitzungen vorgesehen, so wird die rechte Seite der für m gültigen Restriktionen auf 2 gesetzt, so dass an einem Tag maximal zwei Sitzungen stattfinden können.

Birbas, Daskalaki und Housos zwar nicht direkt einbezogen, doch können diese durch Festlegung der betreffenden Variablen auf den Wert 1 ebenfalls leicht berücksichtigt werden. Nicht abgebildet werden die Restriktionstypen Randstunden, Wegezeiten und freie Tage für Lehrer, wobei letztere, wie die Restriktionen (Birbas 6) zeigen, explizit nicht gewollt sind. Die Raumdimension bleibt aus dem Modell vollkommen ausgeklammert, so dass sich die Zuordnung, auch wenn zur Darstellung fünf Indizes verwendet werden, auf die zwei Dimensionen Unterrichtseinheit (Klasse-Lehrer-Fach-Kombination) und Zeit (Tag und Periode) beschränkt.

Die Zielfunktion (Birbas 0) minimiert die Summe der Bewertungen der Periodenzuordnungen aller im Plan enthaltenen Sitzungen. Die Bewertung wird dabei von Birbas, Daskalaki und Housos so genutzt, dass für Hauptfächer (primary courses) die Zuordnung zu späten Perioden eines Tages, für Nebenfächer (non-primary courses) hingegen die Zuordnung zu frühen Stunden bestraft wird. Da viele Lehrer auf Fächer entweder der einen oder der anderen Kategorie spezialisiert sind, wird durch die Zielfunktion auf diese Weise zugleich die Verringerung der Springstundenzahl für Lehrer begünstigt.

Trotz der offenkundigen Redundanzen innerhalb ihres Modells konnten Birbas, Daskalaki und Housos optimale Ergebnisse für fünf Beispielfälle verschiedener Größen vorlegen, die mit Hilfe der Standard-Optimierungssoftware CPLEX (vgl. Fourer [1999]) erzielt wurden. Leider ist aus der Darstellung der Autoren nicht eindeutig ersichtlich, ob es sich bei den getesteten Fällen um reale oder um konstruierte Beispiele handelt, doch wird zumindest für die ausführlich behandelte Fallstudie beteuert, dass es sich um eine „typical Greek high school“ (Birbas, Daskalaki und Housos [1997], S.1195) handele. Auch machen die Autoren keine Angaben über die Laufzeit der Optimierung. Allerdings kann allein die Tatsache, dass es ihnen überhaupt gelungen ist, 0/1-Modelle für die Stundenplansetzung an allgemeinbildenden Schulen in der Größenordnung von bis zu 10.038 0/1-Variablen und 4.538 Restriktionen innerhalb einer offensichtlich überschaubaren Zeitspanne zu lösen, schon als Beleg dafür gewertet werden, dass sich für die Lösung komplexer kombinatorischer Probleme die (Gemischt-)Ganzzahlige Programmierung dank fortschreitender Entwicklung der am Markt erhältlichen Optimierungssoftware mehr und mehr zu einer ernstzunehmenden Alternative gegenüber den dieses Feld bislang beherrschenden heuristischen Verfahren entwickelt.

Auch im Bereich der Veranstaltungssetzung an Universitäten finden sich Anwendungen der Ganzzahligen und der Linearen Programmierung. Schon früh entwickelte Akkoyunlu [1973] ein lineares Modell zur kollisionsfreien Veranstaltungssetzung, das mit einem modifizierten Simplexalgorithmus[25] so gelöst werden kann, dass alle Variablen automatisch entweder den Wert 0 oder 1 annehmen. Die Modifikation besteht dabei darin, nur Matrixelemente mit dem Wert 1 als potenzielle Pivotelemente zuzulassen. Das Modell bildet ausschließlich die Problemrestriktionen der Kollisionsfreiheit als Modellrestriktionen ab, die Vollständigkeitsrestriktionen hingegen werden relaxiert und mit hohem Gewicht in die Zielfunktion aufgenommen, welche außerdem die Minimierung der Summe

[25] Eine Darstellung des allgemeinen Simplexalgorithmus gibt Domschke [1998], S.19ff.

von Bewertungen einzelner Periodenzuordnungen anstrebt. Weitere Problemrestriktionen werden nicht berücksichtigt. White [1975] adaptiert den von Lawrie [1969] beschriebenen Gomory-Ansatz für die Veranstaltungssetzung, berichtet jedoch keine Verfahrensergebnisse. Shih und Sullivan [1977] haben 0/1-Optimierungsmodelle sowohl für die Unterrichtsverteilung als auch für die Setzung an Universitätsfakultäten formuliert, jedoch nur das Modell zur Unterrichtsverteilung implementiert.

Bereits aus jüngerer Zeit stammt der Ansatz von Dinkel, Mote und Venkataramanan [1989], die ein lineares Modell für die simultane Lösung von Unterrichtsverteilungs- und Setzungsproblem im Universitätsbereich entwickelt haben. Darin wird das Gesamtproblem der gegenseitigen Zuordnung von (Unter-)Kursen, Dozenten verschiedener Abteilungen, Räumen und Zeiten als vierstufiges Netzwerk abgebildet, in dem für jede zu erzeugende Sitzung ein Fluss zu bestimmen ist, welcher von einer Quelle über jeweils einen Knoten der Ebenen Abteilung, Dozenten-Fach-Kombination und Raum-Zeit-Kombination hin zu einer Senke läuft. Die Vollständigkeit der Zuordnung wird auf jeder Ebene durch Knotenbilanzgleichungen gesichert, wobei auf der Raum-Zeit-Ebene eine Relaxierungsmöglichkeit vorgesehen ist. Verfügbarkeitsbeschränkungen der Ressourcen werden durch Kapazitierung der Netzwerkflüsse an den betroffenen Knoten in das Modell einbezogen.

Nicht als Modellrestriktionen berücksichtigt werden von Dinkel, Mote und Venkataramanan die Problemrestriktionen der Kollisionsfreiheit für Dozenten und der Vollständigkeit. Sie werden relaxiert und fließen, gemeinsam mit einer Reihe von Zuordnungspräferenzen und Raumnutzungsbewertungen, in eine gewichtete lineare Zielfunktion ein. Der Grund für die Relaxierung dieser wichtigen Restriktionen ist die Unimodularität, die sich durch die Relaxierung für das Modell ergibt und die bewirkt, dass die optimale Lösung des linearen Modells automatisch ganzzahlig ist. Auf den Anstoß eines Branch-and-Bound-Prozesses kann daher verzichtet werden. Bestätigt wurde die Effizienz dieser Vorgehensweise durch die sekundenschnelle Lösung mehrerer, mit fünf Universitätsabteilungen, 750 Dozenten-Fach-Kombinationen, 16 Perioden und sieben verschiedenen Raumgrößen sehr großer Praxisfälle, bei der sich in keinem Fall eine Dozenten-Kollision ergab. Ein Grund für dieses gute Ergebnis liegt möglicherweise darin, dass für die jeweils ca. 300 zuzuordnenden Unterkurse über 150 Dozenten zur Verfügung standen, also je Dozent im Durchschnitt nicht mehr als zwei Unterkurse anfielen.

Brunetta und De Poli [1997] präsentieren ein 0/1-Modell, welches im Rahmen der vollständigen und kollisionsfreien Setzung aller Veranstaltungen einer Universitätsabteilung auch eine globale Raumkapazitätsbeschränkung und ein Doppelunterrichtsverbot abbildet. Die Zielfunktion maximiert die Summe der Bewertungen einzelner Periodenzuordnungen. Obwohl es ihnen gelungen ist, für eine mit nur 102 zu setzenden Stunden sehr kleine, jedoch immerhin für einen Praxisfall generierte Modellinstanz eine Lösung im Totalansatz mit Hilfe der Standardsoftware CPLEX in unter vier Sekunden zu erzielen, bevorzugen Brunetta und De Poli eine vierstufige Dekomposition, die einerseits nach Jahrgängen, andererseits nach der setztechnischen Komplexität der Veranstaltungen unterscheidet. Hintergrund ist die Feststellung der Autoren, dass im Totalmodell nicht für alle Veranstaltungen die jeweils gewünschte Raumkategorie sichergestellt werden konnte. Dieser Mangel wäre möglicherweise durch explizite Einbeziehung der Raumkategorisierung in das Modell leicht zu beheben gewesen.

Stallaert [1997] berichtet über den praktischen Einsatz eines mit ca. 200 Variablen ebenfalls recht kleinen 0/1-Modells für die Lösung des Teilproblems der Setzung von Hauptkursen (core courses) einer Universitätsfakultät, macht jedoch keine Angaben über die verwendete Lösungssoftware oder die benötigte Laufzeit.

Weitere erfolgreiche Anwendungen der Ganzzahligen oder der Linearen Programmierung wurden für die dem Setzungsproblem vorgelagerte Unterrichtsverteilung an Schulen (vgl. Tillett [1975]) bzw. Universitäten (vgl. Hultberg und Cardoso [1997], McClure und Wells [1984], Breslaw [1976]) und für Schichtplanungsprobleme aus anderen Dienstleistungsbereichen (vgl. Jaumard, Semet und Vovor [1998], Beaumont [1997]) entwickelt, die sich jedoch in ihrer Struktur stark von Setzungsproblemen unterscheiden und deshalb hier nicht näher betrachtet werden sollen. Ein Ansatz für die Zuordnung von Raumgruppen zu bereits zeitlich fixierten Unterrichtseinheiten auf Basis eines Linearen Programms findet sich bei Gosselin und Truchon [1986].

Für die Lösung (Gemischt-)Ganzzahliger Optimierungsmodelle stehen heute mehrere leistungsfähige kommerzielle Softwaresysteme der Mathematischen Programmierung zur Verfügung, die – wie das bereits mehrfach genannte CPLEX – laufend weiterentwickelt werden und über Schnittstellen an andere Softwaresysteme wie Tabellenkalkulationsprogramme oder Datenbanken angebunden werden können (vgl. Fourer [1999]). Für die Modelllösung verwenden diese Systeme LP-basierte Branch-and-Bound-Algorithmen, die auf entsprechende LP-Solver zurückgreifen. Letztere basieren standardmäßig auf dem Simplex-Verfahren, doch bieten in jüngerer Zeit einige Programmpakete hierzu einen Interior-Point-Algorithmus als Alternative, der allerdings nur bei Vorliegen bestimmter Modelleigenschaften eine höhere Effizienz als der Simplex-Algorithmus aufweist (vgl. für eine Gegenüberstellung Suhl [2000]).

Einen Überblick über die Charakteristika LP-basierter Branch-and-Bound-Verfahren gibt Tabelle 3.5. Sowohl für die Knoten-, als auch für die Variablen- und Wertewahl stehen verschiedene Regeln zur Verfügung, wobei Variablen- und Wertewahl i.d.R. zu einer sog. Branching-Heuristik gekoppelt sind. Kernelement des Branch-and-Bound ist die Lösung einer sog. LP-Relaxation LP(nk) an jedem neu definierten Knoten nk des Entscheidungsbaumes. Dabei ist LP(nk) mit dem für die jeweilige Problemstellung formulierten (gemischt-)ganzzahligen Ausgangsmodell MIP identisch, mit Ausnahme der beiden folgenden wesentlichen Unterschiede:

1. Diejenigen Entscheidungsvariablen x, denen an Knoten nk bereits ein Wert $w(x)$ aus ihrem jeweiligen Wertebereich zugewiesen ist, werden im zu lösenden Modell LP(nk) auf $w(x)$ fixiert.
2. Für alle in MIP enthaltenen Variablen mit Ganzzahligkeitsbedingung – im Folgenden kurz als ganzzahlige Variablen bezeichnet – wird die Ganzzahligkeitsbedingung aufgehoben (relaxiert). Dies bedeutet, dass eine ganzzahlige

Variable x, deren kleinster zulässiger Wert $ug(x)$ und größter zulässiger Wert $og(x)$ ist, in der Lösung zu LP(nk) beliebige reelle Werte aus dem Intervall $[ug(x), og(x)]$ annehmen kann. LP(nk) ist daher ein Lineares Programm, das mit Hilfe des Simplex-Algorithmus‘ gelöst werden kann.

Ist die optimale Lösung von LP(nk) ganzzahlig, d.h. weisen alle ganzzahligen Variablen ganzzahlige Werte aus ihrem jeweiligen Wertebereich auf, so ist durch die Wertekonstellation dieser Variablen automatisch der bestmögliche zulässige Endknoten im Entscheidungsbaumbereich unterhalb von nk identifiziert. Erweist sich hingegen LP(nk) wegen der darin enthaltenen Variablenfixierungen als unzulässig, so existiert unterhalb von nk kein zulässiger Endknoten. Ist schließlich der Zielfunktionswert z(LP(nk)), der der Bewertung $f(nk)$ des Knotens nk entspricht, schlechter als die Bewertung f_{best} der besten bisher gefundenen zulässigen Lösung k_{best}, so existiert unterhalb von nk kein zulässiger Endknoten, der hinsichtlich seiner Bewertung eine Verbesserung gegenüber k_{best} aufweist. In allen drei Fällen erübrigt sich die Suche nach einer zulässigen Lösung im Bereich des Entscheidungsbaumes unterhalb von nk und der Knoten braucht nicht in die Menge OK der offenen Knoten aufgenommen zu werden. Eine ausführliche Darstellung des Branch-and-Bound-Verfahrens und des Simplex-Algorithmus‘ findet sich bei Domschke und Drexl [1998].

Tabelle 3.5: Charakterisierung des exakten konstruktiven Verfahrenstyps *Branch-and-Bound für die (Gemischt-)Ganzzahlige Programmierung*
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.1)

<table>
<tr><th>Entwurfsentscheidung</th><th>Ausprägung</th></tr>
<tr><td>Knotenwahlstrategie</td><td>offen; häufige Strategien sind:
• zuletzt erzeugter Knoten (LIFO),
• Knoten mit bester Bewertung (best bound),
• Knoten mit kleinster Summe der Verletzungen der Ganzzahligkeitsbedingungen durch die Werte der ganzzahligen Variablen</td></tr>
<tr><td>Variablenwahlstrategie</td><td rowspan="2">offen; werden i.d.R. zu einer gemeinsamen „Branching-Heuristik“ verknüpft; häufige Strategien sind:
• Setze die fraktionelle Variable, d.h. ganzzahlige Variable mit nicht ganzzahligem Wert, mit dem besten Zielfunktionskoeffizienten auf den nächsthöheren ganzzahligen Wert.
• Setze die fraktionelle Variable mit dem schlechtesten Zielfunktionskoeffizienten auf den nächstniedrigeren ganzzahligen Wert.
• Wähle diejenige fraktionelle Variable, deren Wert am weitesten vom nächsten ganzzahligen Wert entfernt liegt und runde sie nach kaufmännischer Regel ganzzahlig.</td></tr>
<tr><td>Wertewahlstrategie</td></tr>
<tr><td>Bewertungsfunktion</td><td>Zielfunktion des (gemischt-)ganzzahligen Modells; Berechnung über die Lösung des zu dem aktuellen Entscheidungsbaumknoten gehörenden LP-Relaxation</td></tr>
<tr><td>Abbruchkriterium</td><td>$|OK| = \emptyset$</td></tr>
<tr><td>Besonderheit</td><td>Weist die Lösung einer LP-Relaxation LP(k), die an einem nicht auf der untersten Ebene des Entscheidungsbaumes liegenden Knoten k betrachtet wird, für alle ganzzahligen Variablen ganzzahlige Werte auf, so ist damit automatisch der bezüglich des unterhalb von k gelegenen Entscheidungsbaumbereichs optimale Endknoten identifiziert.</td></tr>
</table>

3.2.2.3 Goal Programming

Eine besondere Variante der Mathematischen Programmierung stellt das sog. Goal Programming dar (vgl. Meyer und Hansen [1996], S.52ff.). Es dient dazu, mit Hilfe von Über- oder Unterschreitungsvariablen den Abstand zwischen einer Funktion der Entscheidungsvariablen und einem gewünschten Zielwert zu minimieren. Werden z.B. im Rahmen eines linearen oder ganzzahligen Modells n nichtnegative Entscheidungsvariablen x_j mit $j = 1, .., n$ verwendet, so kann der Wunsch, die Summe dieser Entscheidungsvariablen möge möglichst den Wert b annehmen, durch die Goal-Restriktion

$$\sum_{j=1}^{n} x_j + d^- - d^+ = b$$

modelliert werden. Dabei stellen $d^+ \geq 0$ und $d^- \geq 0$ kontinuierliche nicht-negative Variablen dar, die die Über- bzw. Unterschreitung des Zielwerts b messen. Die Bedeutung des durch die Goal-Restriktion formulierten Punktziels im Rahmen der Gesamtzielsetzung wird durch Erweiterung der zu minimierenden Zielfunktion $z_{ohne\ Goal} = f(x_1, .., x_n)$ in der Form

$$\text{minimiere } z_{mit\ Goal} = z_{ohne\ Goal} + w \cdot (d^- + d^+)$$

berücksichtigt, wobei $w > 0$ die Gewichtung des betrachteten Goals im Verhältnis zur restlichen Zielfunktion darstellt. Soll das Goal nur einseitig erreicht werden, also b lediglich nicht unterschritten bzw. nicht überschritten werden, so kann dies durch die Restriktion

$$\sum_{j=1}^{n} x_j + d^- \geq b \qquad \text{im ersten Fall bzw.} \qquad \sum_{j=1}^{n} x_j - d^+ \leq b$$

im zweiten Fall modelliert werden. In die Zielfunktion wird dann nur die jeweils verwendete Variable d^- bzw. d^+ mit dem Gewicht w aufgenommen. Durch die Einbeziehung der mit den Goal-Restriktionen verbundenen Über- und/oder Unterschreitungsvariablen wird ein zugrundeliegendes ganzzahliges Modell wegen des kontinuierlichen Wertebereichs dieser Variablen in ein gemischt-ganzzahliges Modell überführt.

Die Möglichkeiten des Goal Programming wurden für die Stundenplanerstellung bislang vor allem im Universitätsbereich genutzt. Badri et al. [1998] präsentieren ein gemischt-ganzzahliges Modell zur simultanen gegenseitigen Zuordnung von Dozenten, Kursen und Zeiten, in dem Goal-Restriktionen für die Abbildung von Präferenzen der Dozenten für bestimmte Kurse und Zeiten, aber auch für die Relaxation der Vollständigkeitsrestriktionen und der Restriktionen hinsichtlich der Erfüllung der Stundenverpflichtungen der einzelnen Dozenten verwendet werden. Die zu minimierende Zielfunktion besteht in der Summe der nach Restriktionstyp gewichteten Über- und Unterschreitungsvariablen aller Goal-Restriktionen, wobei den für Relaxationen verwendeten Über- und Unterschreitungsvariablen ein besonders hohes Gewicht zugewiesen wird. Der von Badri et al. mit Hilfe ihres Modells und eines nicht näher spezifizierten modifizierten Simplex-Verfahrens gelöste Praxisfall ist allerdings mit nur 31 zu setzenden Kursen sehr klein und kann daher nicht als Beleg für die allgemeine Tauglichkeit des Verfahrens gewertet werden. Dieselbe Aussage gilt für einen früheren Ansatz von Badri [1996], in dem Setzung und Unterrichtsverteilung in zwei getrennten Goal-Programming-Modellen behandelt werden.

Ebenfalls in die Kategorie der gemischt-ganzzahligen Ansätze gehört das Modell von Croucher [1984], welches für Problemfälle der Veranstaltungssetzung entwickelt wurde, in denen neben Restriktionen bezüglich Vollständigkeit, Kollisionsfreiheit und Sperrungen auch solche der minimalen und maximalen Anzahl an Sitzungen je Periode sowie der zeitlichen Reihenfolge zwischen verschiedenen Veranstaltungen zu beachten sind. Goal-Restriktionen werden hier verwendet, um unerwünschte zeitliche Kollisionen zwischen ausgewählten Veranstaltungspaaren nach Möglichkeit zu vermeiden. Auch der von Croucher behandelte Beispielfall ist mit nur 14 zu setzenden Kursen sehr klein.

Weitere Anwendungen des Goal-Programming finden sich im Bereich der Erzeugung von Unterrichtsverteilungen für Universitätsinstitute (vgl. Schniederjans und Kim [1987], Joiner [1980], Harwood und Lawless [1975], Lee und Clayton [1972]). Auch hier sind die bislang getesteten Beispielfälle durch eine eher geringe Größe gekennzeichnet. Anwendungen auf die Stundenplansetzung an Schulen oder die Examenssetzung sind mir nicht bekannt geworden.

3.2.2.4 Lagrange-Relaxation

Durch Relaxation ausgewählter Problemrestriktionen, also den Verzicht auf ihre Formulierung als Modellrestriktionen und die ersatzweise Formulierung als Zielgröße, kann unter Umständen der Lösungsprozess mathematischer Modelle erheblich beschleunigt werden. Dies ist z.B. dann der Fall, wenn durch die Relaxation erreicht werden kann, dass das Modell unimodular ist, also bereits die Lösung der zugehörigen LP-Relaxation ganzzahlige Werte für ganzzahlige Variablen aufweist, so dass kein Branch-and-Bound-Prozess durchlaufen werden muss (vgl. Dinkel, Mote und Venkataramanan [1989]). Eine spezielle Methode zur Modell-Entschärfung ist die Lagrange-Relaxation (vgl. Fisher [1981], Geoffrion [1974]). Mit ihr kann ein Restriktionenblock des Typs

$$A \cdot x = b \text{ (bzw. } \leq b, \geq b\text{)},$$

in dem A eine $m \times n$-Koeffizientenmatrix, x der n-dimensionale Vektor der Entscheidungsvariablen und b ein m-dimensionaler Rechte-Seite-Vektor ist, relaxiert werden, indem er aus dem Restriktionenbereich des Modells herausgelassen und stattdessen in die zu minimierende Zielfunktion $z_{ohne\ Relaxation}(x)$ durch die Erweiterung

$$z_{mit\ Relaxation}(x) = z_{ohne\ Relaxation}(x) + u \cdot (A \cdot x - b)$$

integriert wird. Dabei stellt $u \in R$ (bzw. ≥ 0, ≤ 0) eine m-dimensionale Gewichtung (Vektor der sog. Lagrange-Multiplikatoren) dar. Durch den Wertebereich von u wird sichergestellt, dass für den Vergleich der optimalen Zielfunktionswerte z^{opt} des relaxierten und des unrelaxierten Modells $z^{opt}{}_{mit\ Relaxation} \leq z^{opt}{}_{ohne\ Relaxation}$ gilt. Die Lösung des relaxierten Modells liefert daher eine Zielfunktionsuntergrenze für das unrelaxierte Modell.

Die konkrete Festlegung der Gewichtung u stellt jedoch ein besonderes Problem dar, da durch u die Zielfunktion immer dann verzerrt wird, wenn die linke Seite einer Restriktion $i \in \{1, .., m\}$ von ihrer rechten Seite b_i abweicht. Dieses Problem wirkt sich insbesondere dann negativ aus, wenn die Abweichung im Sinne des unrelaxierten Modells zulässig ist. Daher wird i.d.R. die Lösung des relaxierten Modells mit verschiedenen Werten von u mehrmals wiederholt, um so einen möglichst hohen optimalen Zielfunktionswert zu erreichen und damit eine möglichst scharfe Untergrenze für das unrelaxierte Modell. Eine Erläuterung verschiedener Verfahren zur Bestimmung von u, insbesondere des sog. Subgradienten-Verfahrens, findet sich bei Fisher [1981].

Den umfassendsten Lösungsansatz auf Basis der Lagrange-Relaxation hat Tripathy [1980 und 1984] für die Veranstaltungssetzung einer Universitätsfakultät entwickelt. Basis ist hier ein 0/1-Optimierungsmodell, in dem das für den betrachteten Anwendungsfall zwingende Verbot der zeitlichen Kollision von Veranstaltungen für die Studierenden in die Zielfunktion integriert wird. Der verbleibende Restriktionenblock des so relaxierten Modells erfasst nur die Restriktionstypen Vollständigkeit und Einhaltung der Raumkapazitäten und ist daher unimodular. Der von Tripathy implementierte Algorithmus besteht aus drei Kernkomponenten:

1. einer Adaption des von Bray und Witzgall [1968]) entwickelten Out-of-Kilter-Algorithmus‘ für die Lösung des relaxierten Modells bei gegebenen Werten der Lagrange-Multiplikatoren,
2. einem Subgradienten-Verfahren zur Modifikation der Lagrange-Multiplikatoren und
3. einem Branch-and-Bound-Verfahren zur schrittweisen Beseitigung von Verletzungen relaxierter Kollisionsfreiheitsrestriktionen. Durch diese Komponente soll sichergestellt werden, dass der Lösungsprozess gegen eine auch für das unrelaxierte Problem zulässige Lösung konvergiert. Dabei wird als Branching-Heuristik eine Regel verwendet, nach der aus der relaxierten Restriktion mit der größten Unzulässigkeit diejenige Variable auf 1 gesetzt wird, die aufgrund ihres hohen Zielbeitrags und der geringen Anzahl der für die zugehörige Veranstaltung noch verfügbaren Perioden am höchsten bewertet wird. Die Knotenauswahl innerhalb des Entscheidungsbaumes erfolgt nach der LIFO-Regel, d.h. der zuletzt erzeugte offene Knoten wird als erster abgearbeitet.

Gesteuert wird der Lösungsprozess über das Branch-and-Bound-Verfahren. Ein Branching wird dabei immer dann vorgenommen, wenn nach zehnmaliger wechselseitiger Iteration von Out-of-Kilter-Algorithmus und Modifikation der Lagrange-Multiplikatoren die Lösung des relaxierten Modells aufgrund von Verletzungen der relaxierten Restriktionen für das unrelaxierte Modell unzulässig ist.

Um die Lösung des Gesamtproblems zu erleichtern, wird es in zwei sukzessiv zu lösende Teilprobleme dekomponiert, wobei in der ersten Stufe alle doppelstündigen, in der zweiten Stufe alle einstündigen Veranstaltungen nach o.g. Schema gesetzt werden. Die Setzung wird zusätzlich durch die Zusammenfassung von Studierenden mit homogenen Studienplänen zu Studierendengruppen, von Fächern mit identischen Teilnehmermengen zu Fächergruppen und von gegenseitig austauschbaren Räumen zu Raumgruppen vereinfacht. Mit seinem Verfahren konnte Tripathy einen Praxisfall mit 33 Studierendengruppen, 94 Fächergruppen und 5 Raumgruppen innerhalb weniger Minuten auf einem CDC 7600-Rechner lösen.

Ein weiterer Lagrange-Ansatz im Universitätsbereich wurde von Carter [1989] für das Teilproblem der Raumzuordnung zu bereits zeitlich fixierten Unterrichtseinheiten entwickelt. Anwendungen im Bereich der Examensplanung wurden von Arani, Karwan und Lotfi [1988]) sowie von Balakrishnan, Lucena und Wong [1992] vorgestellt. Beide Ansätze beschränken sich dabei auf das Teilproblem, Gruppen von bereits in einer vorgelagerten Lösungsstufe zeitlich parallelisierten Examina jeweils einer konkreten Periode des Prüfungszeitraumes zuzuordnen. Dabei setzen Arani, Karwan und Lotfi die Lagrange-Relaxation ähnlich wie Tripathy im Rahmen eines Branch-and-Bound-Verfahrens ein, während Balakrishnan, Lucena und Wong sie zur Berechnung von Zielfunktionsuntergrenzen im Rahmen eines iterativen Zuordnungsalgorithmus‘ verwenden.

Anwendungen der Lagrange-Relaxation auf die Stundenplansetzung an Schulen sind mir bislang nicht bekannt geworden.

3.2.2.5 Nicht-lineare Optimierung

Mathematische Modelle, die nicht-lineare Terme in der Zielfunktion und/oder einigen Restriktionen aufweisen, sind ausgesprochen schwierig zu lösen und werden daher weitgehend gemieden.[26] Ein nicht-linearer Ansatz für das Stundenplanerstellungsproblem einer Universität wurde von Boronico [2000] vorgeschlagen. Dabei wird das Gesamtproblem in drei Teilprobleme dekomponiert, die sukzessive gelöst werden. In der ersten Stufe werden Kurse in Unterkurse aufgeteilt, wobei die Anzahl der Unterkurse eines Kurses von der vorhandenen Dozentenkapazität und von der insgesamt für den Kurs zu erwartenden Teilnehmerzahl abhängt. In der zweiten Stufe werden Unterkurse, die zeitlich parallel stattfinden sollen, zu Gruppen zusammengefasst, wobei die Summe der zu erwartenden Konflikte für die Studierenden minimiert wird. In der dritten Stufe werden die so erzeugten Kursgruppen konkreten Perioden zugeordnet und gleichzeitig die Zuordnung der Dozenten zu den Veranstaltungen bestimmt. Hier wird versucht, Überbelastungen einzelner Dozenten möglichst zu vermeiden.

Während die erste Stufe noch als 0/1-Modell mit linearer Struktur formuliert und gelöst wird, werden die zweite und die dritte Stufe durch jeweils ein 0/1-Modell mit nicht-linearer Struktur repräsentiert und mit Hilfe eines Gradienten-Verfahrens gelöst. Als Basisdaten für

[26] Vgl. Zimmermann [1992], S.208ff. für verschiedene Lösungsverfahren. Nash [1998] gibt einen Überblick über kommerzielle Softwarepakete für die Nicht-lineare Programmierung.

die Planerstellung dienen die aus Simulationsläufen gewonnenen prognostizierten Teilnehmerzahlen der einzelnen Kurse und zu erwartenden Kurskonflikte für die Studierenden. Mit seinem Verfahren konnte Boronico innerhalb nicht näher spezifizierten Zeitspanne einen Plan für einen mit 18 Kursen überschaubaren, aber durchaus realen Praxisfall erzeugen.

Für die Stundenplansetzung an Schulen existieren meines Wissens bislang keine Ansätze auf Basis nicht-linearer mathematischer Modelle.

3.2.2.6 Implizite Enumeration mit Logischer Programmierung

Eine Alternative zur Mathematischen besteht in der Logischen Programmierung. Logische Sprachen wie PROLOG, CHIP (vgl. van Hentenryck [1991]) oder Oz (vgl. Henz und Würtz [1996]) unterstützen standardmäßig die Lösung von Restriktionserfüllungsproblemen (CSP)[27] durch implizite Enumeration nach dem LIFO-Prinzip (vgl. Tabelle 3.6). Die Beschneidung des Entscheidungsbaumes erfolgt dabei dadurch, dass im Rahmen der Analyse eines neu erzeugten Knotens *nk* neben der Zulässigkeitsprüfung der bereits festgelegten Variablenwerte gleichzeitig eine Vorausprüfung der Restriktionen (constraint propagation) hinsichtlich unzulässiger Werte für diejenigen Variablen erfolgt, die noch keine Zuweisung erhalten haben. Diese Werte werden für die Betrachtung der Nachfolgeknoten von *nk* temporär aus den Wertebereichen der betroffenen Variablen entfernt, so dass sich die weitere Enumeration auf die (noch) nicht gelöschten Werte beschränkt.

Tabelle 3.6: Charakterisierung des exakten konstruktiven Verfahrenstyps
Implizite Enumeration mit Logischer Programmierung
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.1)

Entwurfsentscheidung	***Ausprägung***
Knotenwahlstrategie	zuletzt erzeugter Knoten (LIFO)
Variablenwahlstrategie	offen
Wertewahlstrategie	offen
Bewertungsfunktion	Standard = Zulässigkeitsprüfung; weitergehende Bewertungsfunktionen können/müssen selbst definiert werden
Abbruchkriterium	$\|OK\| = \varnothing$
Besonderheit	Im Rahmen der Zulässigkeitsprüfung werden die Wertebereiche der noch nicht festgelegten Variablen um als unzulässig identifizierte Werte reduziert.

Während auf diese Weise die implizite Enumeration von Entscheidungsbäumen für CSPs aktiv unterstützt wird, gilt dies nicht in gleichem Maße für die Lösung von Optimierungsmodellen. Hierzu muss der Entwickler eine Prozedur für die Berechnung der Bewertungsfunktion programmieren. Obere Bewertungsgrenzen f_{best}, die mit Hilfe dieser Prozedur an zulässigen erzeugten Endknoten des Entscheidungsbaumes gewonnen werden, können als

[27] Eine Definition von Restriktionserfüllungsproblemen findet sich in Abschnitt 3.2.1.1.1.

Restriktionen in das CSP eingefügt werden, um so die weitere Suche auf solche Bereiche des Baumes einzuschränken, die eine Verbesserung gegenüber f_{best} versprechen.

Obwohl das skizzierte implizite Enumerationsverfahren als exaktes Verfahren für die optimale Lösung von Setzungsproblemen eingesetzt werden kann, wurde diese Möglichkeit bislang wenig ausgenutzt. Der in dieser Hinsicht vermutlich am weitesten gehende Ansatz stammt von Henz und Würtz [1996], die sich mit der Setzung von Veranstaltungsplänen einer Hochschule befasst haben. Sie formulieren ein logisches Modell, in dem jede Entscheidungsvariable für die Startperiode genau eines Kurses steht. Einbezogen werden zahlreiche Restriktionen, die neben der Vollständigkeit und Kollisionsfreiheit u.a. Sperrungen, Fixierungen, freie Tage für Dozenten, Raumkapazitäten und Mittagspausen betreffen. Gelöst wird das Modell mit Hilfe eines Branch-and-Bound-Verfahrens, das auf der standardmäßigen LIFO-Enumeration basiert. Darin wird für die Variablenwahl das sog. First-Fail-Prinzip verwendet, ein Engpasskriterium, nach dem diejenige Variable als nächste zu setzen ist, die aktuell den kleinsten Wertebereich aufweist. Die Wertewahl für die zu setzende Variable erfolgt in zeitlicher Reihenfolge der Perioden, wobei die erste zu testende Periode zufällig bestimmt wird. Als zu minimierende Bewertungsfunktion wird die Anzahl der Sitzungen nach einer Mittagspause vorgeschlagen. Henz und Würtz erheben nicht den Anspruch, innerhalb einer akzeptablen Zeitspanne eine optimale Lösung präsentieren zu können. Sie betonen stattdessen die Möglichkeit, die Optimierung abzubrechen, wenn eine hinreichende Lösungsqualität erreicht ist. Mit ihrem Verfahren haben die Autoren für eine Hochschule mit 91 Kursen à drei bzw. sechs Perioden, 34 Dozenten, sieben Räumen und 180 Perioden à 15 Minuten eine hinreichend gute Lösung innerhalb von ca. 10 Minuten auf einer Sun Sparc 20 mit 60 MHz erzeugt. Die erste zulässige Lösung wurde in weniger als einer Minute gefunden.

Guéret et al. [1996] erstellen ein Logisches Programm für die Lösung des Setzungsproblems einer Universität, verzichten jedoch auf eine Optimierung. Stattdessen wird die Enumeration, die an das standardmäßige LIFO-Prinzip anknüpft, bei Erreichen der ersten zulässigen Lösung gestoppt. Das Verfahren verwendet für die Variablenwahl das Kriterium der maximalen Länge der durch die Variable repräsentierten Veranstaltung, bei mehreren Veranstaltungen gleicher Länge das First-Fail-Prinzip als Sekundärkriterium. Der Variablenwert, der die Startperiode der Veranstaltung angibt, wird in zwei Stufen bestimmt. Zuerst wird derjenige Tag ausgewählt, an dem die summierte Gesamtlänge der diesem Tag bereits zugeordneten Veranstaltungen der betroffenen Studierendengruppe am kürzesten ist. Sodann wird innerhalb des gewählten Tages die früheste zulässige Startperiode zugewiesen. Die Autoren konnten mit diesem Verfahren innerhalb weniger Sekunden auf einem nicht näher spezifizierten Computer einen Veranstaltungsplan für ein Institut mit 42 Dozenten und 160 Studierenden erzeugen, die sich zu insgesamt 91 Veranstaltungen zu treffen hatten. Die gemeinsame Planung zweier Institute mit 165 Veranstaltungen gelang erst nach der Relaxation einiger Restriktionen. Als Grund vermuten die Autoren, dass das Modell ohne Relaxation unzulässig war.

Einen weiteren Ansatz der Logischen Programmierung für die Setzung von Veranstaltungsplänen präsentieren Deris et al. [1997] sowie Deris, Omatu und Ohta [2000]. Auch hier beschränkt sich der Algorithmus auf das Auffinden einer zulässigen Lösung mit Hilfe der LIFO-Enumeration und einer engpassorientierten Variablenwahl.[28] Im Unterschied zu den vorgenannten Ansätzen werden jedoch sowohl Variablen für die zeitliche als auch Variablen für die räumliche Zuordnung der Veranstaltungen definiert. Für die Wertewahl werden zielgerichtete Kriterien bestimmt, die die Zuordnung von Veranstaltungen zu möglichst frühen Perioden und zu solchen Räumen begünstigen, die in Ausstattung, Lage und Sitzplatzkapazität dem jeweiligen Bedarf am ehesten entsprechen. Beide Aufsätze berichten von der erfolgreichen Lösung großer Setzungsprobleme mit mehreren hundert zu erzeugenden Sitzungen in bis zu 32 Minuten auf einem Pentium-100 MHz-PC.

Über die bereits genannten Beiträge hinaus, wurden Logische Programme für die enumerative Erzeugung zulässiger Veranstaltungspläne von Fahrion und Dollansky [1992] sowie von Erben und Keppler [1996] entwickelt. Letztere verwenden das damit gewonnene Ergebnis als Startlösung für einen Genetischen Algorithmus, um so eine verbesserte Lösungsqualität zu erzielen (vgl. Abschnitt 3.2.4.4). Eine erfolgreiche Anwendung im Bereich der Examensplanung stammt von Boizumault, Delon und Peridy [1996].

Obwohl die Methodik der Logischen Programmierung in jüngerer Zeit eine zunehmende Verbreitung gewonnen hat, wurde ihre Kernidee, nach jeder Wertzuweisung einer Variablen die Wertebereiche der noch ungesetzten Variablen über eine Restriktionsanalyse um unzulässige Werte zu reduzieren, bereits in der Frühzeit der automatisierten Setzung verfolgt. Dies zeigt der Lösungsansatz, welcher von Gotlieb [1963] für das in Unterkapitel 2.3 vorgestellte vereinfachte Setzungsproblem VSP entwickelt und von Csima und Gotlieb [1964] implementiert wurde.[29] Darin werden nach jeder vorgenommenen Zuordnung einer Klasse-Lehrer-Kombination (k, l) zu einer Periode p die gemeinsamen Verfügbarkeiten $v_p^{\kappa l}$ für alle Klassen $\kappa \in K$ und Lehrer l in Periode p sowie $v_p^{k\lambda}$ für Klasse k und alle Lehrer $\lambda \in L$ in Periode p auf den Wert 0 reduziert, so dass weitere Zuordnungen von Klasse-Lehrer-Kombinationen, die Klasse k oder Lehrer l betreffen, zu Periode p ausgeschlossen werden. Anhand der ebenfalls in Unterkapitel 2.3 beschriebenen Hall-Bedingungen wird nach jeder Zuordnung überprüft, ob es noch ungesetzte Klasse-Lehrer-Kombinationen gibt, die aufgrund einer Gleichheit der Anzahl der für sie jeweils noch verfügbaren Perioden mit der Anzahl der ihnen noch zuzuordnenden Perioden als nächstes gesetzt werden müssen. Gibt es derartige Klasse-Lehrer-Kombinationen, so werden sie in der folgenden Verfahrensiteration den für sie verfügbaren Perioden zugeordnet. Andernfalls wird per Zufallsauswahl eine zulässige Zuordnung bestimmt und vorgenommen. Für den Fall, dass nach einer Zuordnung eine Verletzung der Hall-Bedingungen festgestellt wird, sieht das Verfahren ein Backtracking bis zur letzten zulässigen Zuordnung vor. Es orientiert sich demnach hinsichtlich der Knotenwahlstrategie im Entscheidungsbaum am LIFO-Prinzip.

---

28 In Deris et al. [1997] werden an mehreren Stellen irreführende Begriffe wie „Branching“, „Bounding“ und „optimale Lösung“ verwendet, die die Verwendung eines optimierenden Verfahrens suggerieren. Dies ist jedoch nicht der Fall, wie aus der näheren Betrachtung der beigefügten Pseudocodes und der sehr ähnlichen, aber klarer formulierten Darstellung in Deris, Omatu und Ohta [2000] deutlich wird.

29 vgl. hierzu auch die Anmerkungen in Duncan [1965].

Da sich die Überprüfung der Hall-Bedingungen nach jeder Setzung als sehr aufwendig erwies, konnten mit Hilfe des Gotlieb-Verfahrens nur extrem kleine Beispielfälle mit neun Lehrern, neun Klassen und neun Perioden gelöst werden. Gotlieb schlägt daher eine Dekomposition des Setzungsproblems in ein Wochen- und mehrere Tagesprobleme vor. Dabei beinhaltet das Wochenproblem die vollständige Aufteilung der zu erzeugenden Sitzungen auf die Unterrichtstage unter Beachtung der zeitlichen Kapazitäten der Lehrer und Klassen, jedes Tagesproblem hingegen die exakte zeitliche Fixierung der durch die Lösung des Wochenproblems vorgegebenen Sitzungen innerhalb eines Unterrichtstages. Das oben beschriebene Lösungsverfahren eignet sich insbesondere für die Lösung der Tagesprobleme, kann jedoch nach Auffassung Gotliebs mit geringer Modifikation auch auf das Wochenproblem angewendet werden.

Die Lösung realer Probleme mit Hilfe eines auf dem Gotlieb-Verfahren basierenden Ansatzes und einer Wochen-Tage-Dekomposition gelang jedoch erst Lions [1966b und 1967], der hierzu die Engpassprüfung über die Hall-Bedingungen durch die Anwendung der Ungarischen Methode zur Matrixreduktion (vgl. Zimmerman [1992], S.113ff.) ersetzte und die Möglichkeit der Vorab-Fixierung nutzte, um einige zuvor nicht berücksichtigte Restriktionen, z.B. die Beachtung von Fachraumanforderungen, einzubeziehen. Allerdings werden in dem Ansatz von Lions nur die Tagesprobleme mit Hilfe des modifizierten Gotlieb-Ansatzes gelöst, während für das Wochenproblem eine konstruktive Heuristik verwendet wird, die sich sowohl an einem Engpasskriterium als auch an Präferenzkriterien hinsichtlich der Verteilung des Unterrichts auf die verschiedenen Wochentage orientiert. Obwohl Lions für mehrere Schulen vollständige oder fast vollständige Stundenpläne erzeugen konnte, wurde sein Ansatz – mit Ausnahme eines nur anhand von Zufallsproblemen getesteten modifizierten Ansatzes der Matrixreduktion von Johnston und Hoare [1975] – nicht weiterentwickelt. Zum originären Gotlieb-Verfahren erschienen lediglich noch eine generalisierende, eher theoretische Betrachtung der Zulässigkeitsbedingungen durch Lions [1969] und eine Untersuchung zulässiger Tauschvorgänge innerhalb eines Stundenplanes durch Dempster [1968], die jedoch offenbar keine Anwendungsbedeutung erlangt haben.

Während den in diesem Abschnitt berichteten Ansätzen gemein ist, dass sie eine Form der impliziten Enumeration wählen, die das Auffinden zumindest einer zulässigen Lösung garantieren, sofern es eine gibt, wird von anderen Autoren der Enumerationsprozess so modifiziert, dass diese Garantie nicht mehr gilt. Diese letztgenannten Ansätze können daher, obwohl sie ebenfalls zur Logischen Programmierung gerechnet werden, nicht mehr als auf einem exakten Verfahrenstyp basierend betrachtet werden. Sie werden in Abschnitt 3.2.3.3 behandelt.

3.2.2.7 Beurteilung

Die in den vorangegangenen Abschnitten vorgestellten Lösungsansätze auf Basis exakter konstruktiver Verfahrenstypen sind hinsichtlich ihres Potenzials für die erfolgreiche Lösung des Setzungsproblems an allgemeinbildenden Schulen unterschiedlich zu beurteilen (vgl. Tabelle 3.7).

Tabelle 3.7: Beurteilung *exakter konstruktiver Lösungsansätze*
(zur Erläuterung der Kriterien vgl. Tabelle 2.11, S.42)

Verfahrenstyp / *Kriterium*	exakte Graphenfärbung		(Gemischt-) Ganzzahlige Programmierung[1]		Nicht-lineare Programmierung		Implizite Enumeration mit Log. Programmierung	
	bisherige Ansätze	Potenzialeinschätzung	bisherige Ansätze	Potenzialeinschätzung	bisherige Ansätze[3]	Potenzialeinschätzung	bisherige Ansätze	Potenzialeinschätzung
Berücksichtigung der Restriktionstypen								
• Vollständigkeit (R-01):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.
• Kollisionsfreiheit (R-02):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.
• Sperrungen (R-03):	ja	mögl.	ja	mögl.	nein	mögl.	ja	mögl.
• Springstundenverbot für Klassen (R-04):	nein	schw.	indir.[2]	mögl.	nein	mögl.	nein	mögl.
• Doppelunterrichtsverbot (R-05):	nein	schw.	ja	mögl.	nein	mögl.	nein	mögl.
• Randstunden (R-06):	nein	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
• Fixierungen (R-07):	nein	mögl.	ja	mögl.	nein	mögl.	ja	mögl.
• Wegezeiten (R-09):	nein	schw.	nein	mögl.	nein	mögl.	nein	mögl.
• Freie Tage für Lehrer (R-12):	nein	mögl.	nein	mögl.	ja	mögl.	ja	mögl.
Wahrscheinlichkeit einer zulässigen Lösung bzgl. aller o.g. Restriktionstypen:		mittel - hoch		hoch		hoch		hoch
Berücksichtigung der Zielsetzungen								
• Gleichlange Klassentage (Z-01):	nein	mögl.	indir.[2]	mögl.	nein	mögl.	nein	mögl.
• Früher Unterrichtsbeginn (Z-02):	nein	mögl.	indir.[2]	mögl.	nein	mögl.	nein	mögl.
• Springstundenmin. für Lehrer (Z-06):	nein	schw.	ja	mögl.	nein	mögl.	nein	mögl.
• Vertretungsbereitschaft (Z-10):	nein	mögl.	ja	mögl.	nein	mögl.	nein	mögl.
• Fächerbeziehungen (Z-12):	nein	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
Handhabung der Mehrfachzielsetzung:	Einzelziel[4]	offen	Zielgew.	offen	Zielgew.[4]	offen	Einzelziel[4]	offen
Laufzeit:	kurz - mittel	unakzeptabel	kurz - mittel	lang	n.v.	unakzeptabel	kurz - mittel[5]	lang

Abkürzungen: mögl. = möglich, schw. = schwierig, indir. = indirekt, f. Doz. = für Dozenten, Zielgew. = Zielgewichtung, n.v. = nicht verfügbar

Anmerkungen:
1 inkl. Goal Programming, Lagrange-Relaxation
2 vgl. das Modell in Birbas, Daskalaki und Housos [1997], in dem jede Klasse in jeder Periode eines Tages mit Ausnahme der letzten aktiv sein muss
3 nur Boronico [2000]
4 unter Einbeziehung anderer als der hier genannten Zielsetzungen
5 Zeit bis zur Erzielung der ersten zulässigen Lösung

Fragwürdig sind die Erfolgsaussichten exakter Graphenfärbungsverfahren, da im Schulbereich Zielsetzungen und Restriktionen anfallen, die die sinnvolle Nutzung spezieller Methoden der Graphentheorie, z.B. zur Identifikation und ggf. Reduktion der chromatischen Zahl eines Graphen, weitgehend unmöglich machen. Einige Restriktionen wie das Springstundenverbot für Klassen oder das Doppelunterrichtsverbot lassen sich zudem in einem graphischen Modell nicht ohne weiteres abbilden. Die Verwendung eines graphischen Ansatzes liefe daher auf die Implementation eines weitgehend modellunabhängigen Verfahrens der impliziten Enumeration hinaus, das jedoch ohne eine Verknüpfung mit

Algorithmen der Mathematischen oder Logischen Programmierung kaum in die Lage versetzt werden kann, Unzulässigkeiten frühzeitig zu entdecken und so eine langwierige Tiefensuche in unzulässigen Bereichen des Entscheidungsbaumes zu vermeiden. Für reale Probleme müsste daher mit unakzeptablen Laufzeiten gerechnet werden.

Wesentlich besser erscheinen die Aussichten, mit Hilfe der Ganzzahligen oder Gemischt-ganzzahligen Programmierung zum Erfolg zu gelangen. Durch die Formulierung eines entsprechenden Modells lässt sich eine hohe Abbildungsgenauigkeit des Problems realisieren, da sich alle Problemrestriktionen in Form linearer Gleichungen oder Ungleichungen abbilden lassen (vgl. hierzu auch die Modelle in Kapitel 4). Einziger Schwachpunkt ist die Unmöglichkeit der mehrdimensionalen Abbildung des Zielkatalogs. Sie zwingt – wie auch in allen anderen in der Praxis eingesetzten Verfahren – zur Vornahme einer Zielgewichtung oder die Beschränkung auf ein Einzelziel. Allerdings kann durch Formulierung entsprechender Modellrestriktionen für jedes der angestrebten Ziele ein Mindestniveau gesichert werden.

Das größte Hindernis für den Einsatz der (Gemischt-)Ganzzahligen Programmierung besteht nicht in der Formulierung eines geeigneten Modells, sondern in der für seine Lösung benötigten Laufzeit, die wegen der NP-Vollständigkeit des Setzungsproblems bis heute weitläufig als unmöglich gilt.[30] Allerdings wird dieses Pauschalurteil durch die in jüngerer Zeit berichteten Erfolge (vgl. Birbas, Daskalaki und Housos [1997], Brunetta und De Poli [1997], Dinkel, Mote und Venkataramanan [1989]), die nicht zuletzt aufgrund der hohen Leistungsfähigkeit heute verfügbarer Standard-Optimierungssoftware beruhen, mehr und mehr in Frage gestellt. Dies gilt umso mehr, als für den Fall, dass ein (gemischt-) ganzzahliges Modell nicht in akzeptabler Zeit zu lösen ist, zwei Methoden für seine Vereinfachung zur Verfügung stehen: die Dekomposition des Modells in mehrere Teilmodelle und die Relaxation von Restriktionenblöcken. Mit beiden Methoden sind jedoch Gefahren verbunden. So kann eine Dekomposition dazu führen, dass wegen der Unzulässigkeit eines Teilmodells keine für das Gesamtmodell zulässige Lösung gefunden werden kann. Eine Relaxation kann bewirken, dass die ermittelte Lösung des relaxierten Modells für das reale Problem unzulässig ist.

Für eine Relaxation stehen die unterschiedlichen Konzepte der Lagrange-Relaxation und des Goal-Programming zur Verfügung. Von diesen erscheint zunächst die Lagrange-Relaxation als die attraktivere, da sie den Restriktionenblock des Modells vollständig von den relaxierten Restriktionen befreit und so die Lösung des Modells erheblich erleichtert. Dieser Vorteil wird jedoch durch die Notwendigkeit der iterativen Bestimmung der Lagrange-Multiplikatoren und die erforderlichen Maßnahmen zur Sicherstellung einer auch für das unrelaxierte Modell zulässigen Lösung stark relativiert. Wird mit Hilfe des Goal-Programming relaxiert, ist wegen des Ersatzes der herausgenommenen Restriktionen durch entspre-

[30] Stellvertretend für diese nach meiner Erfahrung weit verbreitete Auffassung sei an dieser Stelle auf die Feststellung verwiesen, welche Drexl und Salewski [1997] nach der Formulierung ihres 0/1-Modells für die Setzung von Schulstundenplänen und dem Hinweis auf die NP-Vollständigkeit des Problems getroffen haben (ebd. S.199): „Therefore, the only line of attack for tackling practical problem sizes comprising thousands of binary variables [..] is provided by approximation [= heuristic; Hilbert-Siekmann] methods.“

chende Goal-Restriktionen der Vereinfachungseffekt für die Modelllösung geringer, doch entfällt hier die umständliche Bestimmung von Lagrange-Multiplikatoren, da nur einmalig eine Goal-Gewichtung für die Zielfunktion festzulegen ist. Wird dabei den Über- bzw. Unterschreitungsvariablen ein überragendes Gewicht gegenüber den originären Zielkriterien eingeräumt, so ist die optimale Lösung des relaxierten Modells für das unrelaxierte Modell nur dann unzulässig, wenn es für letzteres keine zulässige Lösung gibt.

Erheblich schwieriger zu lösen als ganzzahlige Modelle mit linearer Struktur sind in jedem Fall nicht-lineare Modelle. Da sich das Setzungsproblem an allgemeinbildenden Schulen mit Hilfe ausschließlich linearer Variablenbeziehungen darstellen lässt, sollte daher auf die Verwendung nicht-linearer Formulierungen verzichtet werden.

Ein weiterer Verfahrenstyp, welchem zunehmende Bedeutung zukommt, ist die Logische Programmierung. Die von ihr verwendeten Prüfungsroutinen der Constraint Propagation stehen in direkter Konkurrenz zu den LP-Solvern innerhalb der (Gemischt-)Ganzzahligen Programmierung. Die Frage, welche dieser beiden Alternativen bei der Gestaltung eines Branch-and-Bound-Verfahrens zu bevorzugen ist, ist nicht leicht zu beantworten. Darby-Dowman und Little [1998] haben in vergleichenden Leistungstests mit Optimierungsmodellen für die Gruppenbildung bei Golfturnieren, die Einsatzplanung von Busfahrern und die Maschinenbelegungsplanung keine generelle Hierarchie zwischen Logischer und Ganzzahliger Programmierung identifizieren können. Allerdings sind die von den Autoren behandelten Fälle eher klein, so dass eine Schlussfolgerung für das deutlich größere Setzungsproblem an Schulen riskant erscheint. Insgesamt muss festgestellt werden, dass die Menge der bislang mit exakten logischen und mathematischen Ansätzen in Bezug auf Setzungsprobleme gesammelten Erfahrungen noch viel zu gering ist, als dass sich auf ihrer Basis ein fundiertes oder gar letztgültiges Urteil über das Potenzial dieser Ansätze fällen ließe.

3.2.3 Ansätze auf Basis heuristischer konstruktiver Verfahrenstypen

Heuristische konstruktive Verfahren werden eingesetzt, um in möglichst kurzer Zeit eine zulässige Modelllösung mit, soweit eine entsprechende Funktion definiert ist, möglichst guter Bewertung zu erzeugen. Für die gewonnene Lösung sind dabei zwei mögliche Verwendungen denkbar. Die erste besteht in einer direkten Umsetzung in die Realität. Dazu ist es erforderlich, dass die konstruktive Heuristik alle wichtigen Restriktionen und Zielsetzungen des Problems umfassend berücksichtigt. Die zweite mögliche Verwendung einer konstruktiv-heuristisch erzeugten Lösung stellt ihr Gebrauch als Startlösung für eine iterative Heuristik dar, die sie durch verschiedene Operationen in eine Lösung mit deutlich verbesserter Qualität zu überführen versucht. Konstruktive Heuristiken, die als Basis einer iterativen Heuristik dienen, sind häufig sehr einfach konzipiert und lassen eine Reihe von Restriktionen unberücksichtigt, die erst in die iterative Heuristik einbezogen werden.

Da sich konstruktive Heuristiken der zweiten Kategorie bei gleichzeitig einfacherer Struktur im Grunde derselben Verfahrensprinzipien bedienen wie die der ersten Kategorie, kann sich die Betrachtung in diesem Abschnitt auf solche Verfahren beschränken, deren Ergebnisse zur direkten Umsetzung in die Praxis bestimmt sind. Auf konstruktive Heuristiken, die als Basis einer iterativen Heuristik dienen, wird daher erst und nur, soweit erforderlich, in Zusammenhang mit der Erläuterung der jeweils auf sie aufbauenden iterativen Heuristik in Abschnitt 3.2.4 eingegangen.

3.2.3.1 Engpassorientierte Zuordnung auf Basis einfacher logischer Modelle

Einer der frühesten Ansätze zur computerisierten Setzung von Schulstundenplänen stammt von Appleby, Blake und Newman [1961]. Seine Grundidee besteht in der Verwendung von Engpassgrößen als Prioritätskriterien für die sukzessive Zuordnung von Unterrichtseinheiten zu Perioden. Der Ablauf des Verfahrens gestaltet sich wie folgt: Aus der Menge aller noch ungesetzten Unterrichtseinheiten wird diejenige Einheit u ausgewählt, die die kleinste Differenz zwischen der Anzahl P der unter Berücksichtigung des bereits vorhandenen Teilplans für u noch zulässigen Perioden und der Anzahl N der u insgesamt zuzuordnenden Perioden aufweist und somit den größten Engpass darstellt. Ist nun $P - N$ gleich 0, so besteht hinsichtlich der Periodenwahl kein Freiheitsgrad und es werden die als zulässig identifizierten Perioden zugeordnet. Ist $P - N$ größer als 0, müssen aus den P zulässigen Perioden N ausgewählt werden. Es werden dann diejenigen N Perioden zugeordnet, die die spätere Setzung weiterer noch ungesetzter Unterrichtseinheiten am wenigsten einschränken. Ist jedoch $P - N$ kleiner als 0, kann keine zulässige Zuordnung mehr vorgenommen werden und das Verfahren bricht ab.

Das Lösungsverfahren von Appleby, Blake und Newman weist eine sehr einfache Struktur auf. Es basiert auf einem logischen Modell, in dem die zu generierenden Sitzungen der Unterrichtseinheiten die Variablen, die zuzuordnenden Perioden die möglichen Variablenwerte darstellen. Das Verfahren implementiert zwar mit der schwierigkeitsorientierten Periodenauswahl eine Vorausschauregel, die ihm eine hohe Zahl erfolgreicher Setzungen ermöglicht, doch verfolgt es lediglich einen einzigen Lösungspfad im Entscheidungsbaum und enthält keinerlei Vorkehrung für den Fall, dass eine Unterrichtseinheit nicht mehr zulässig gesetzt werden kann. Der erzeugte Plan kann daher unvollständig sein. Desweiteren werden durch das Verfahren lediglich Restriktionen der Kollisionsfreiheit für Lehrer und Klassen als solche berücksichtigt. Die Einhaltung des Doppelunterrichtsverbots wird durch bevorzugte Auswahl von Perioden an Tagen, an denen das betroffene Fach (noch) nicht unterrichtet wird, zwar angestrebt, aber nicht erzwungen. Doppelstunden werden so zugeordnet, dass sie keine großen Pausen überdecken. Andere Restriktionstypen und Zielsetzungen spielen in dem Verfahren keine Rolle, und die Raumdimension bleibt vollständig ausgeklammert. Trotz dieser Einschränkungen kann der Ansatz von Appleby, Blake und Newman als Pionierleistung der computergestützten Stundenplansetzung angesehen werden, an die zahlreiche spätere Autoren mit Erweiterungen und Anpassungen an die jeweilige Problemstellung angeknüpft haben.

Ähnlich strukturiert wie der Ansatz von Appleby, Blake und Newman [1961] ist das Verfahren von Junginger [1969], der jedoch zusätzlich Sperrungen und die Kollisionsfreiheit für – vorab zugeordnete – Räume einbezieht und das Doppelunterrichtsverbot als verbindlich durchsetzt. Ibrahim [1972] bestimmt die zu setzende Unterrichtseinheit und die Periode simultan über ein kombiniertes Engpasskriterium. Der Ansatz von Gunzenhäuser und Junginger [1964] hingegen operiert periodenweise. Hier werden, beginnend mit der ersten, jeder Periode so viele noch ungesetzte Unterrichtseinheiten wie zulässig möglich zugeordnet, wobei die Unterrichtseinheiten in willkürlicher Reihenfolge der Klassen und innerhalb einer Klasse nach einem engpassorientierten Bewertungskriterium ausgewählt werden, das verschiedene Restriktionen berücksichtigt. Ebenfalls periodenweise geht Keel [1972] vor, jedoch werden die Perioden hier nicht in zeitlicher Reihenfolge, sondern nach dem Prioritätskriterium der kleinsten Anzahl in einer Periode verfügbarer Räume abgearbeitet. Die engpassorientierte Auswahl der zu setzenden Unterrichtseinheit erfolgt ohne Differenzierung nach Klassen.

Weitere engpassorientierte Setzheuristiken mit nur einem Lösungspfad wurden für die Veranstaltungssetzung an Universitäten von Heinrich [1984], Barham und Westwood [1978], Keil und Meditsch [1976] und Rössler [1969] sowie für die Setzung von Examensplänen von Junginger und Schlipphack [1974], Foxley und Lockeyer [1968], Wood [1968], Zehnder [1965] und Cole [1964] entwickelt. Sie sind den hier beschriebenen Ansätzen für die Stundenplansetzung an Schulen vergleichbar, doch ist im Universitäts- bzw. Examensbereich die Gefahr, dass ein durch ein engpassorientiertes Verfahren erzeugter Plan unvollständig ist, aufgrund der im Vergleich zum Schulproblem einfacheren Problemrestriktionen tendenziell geringer (vgl. Abschnitt 2.5.3).

Bereits früh entstanden Bestrebungen, die Flexibilität des Setzverfahrens und damit die Wahrscheinlichkeit für das Auffinden einer vollständigen Lösung durch die Ermöglichung von Tausch- oder Verschiebungsvorgängen zu erhöhen. Berghuis, van der Heiden und Bakker [1964] betten in ihr Setzungsprogramm für Schulen eine Verschiebungsroutine ein, welche aufgerufen wird, wenn im ersten Setzversuch nicht alle geforderten Sitzungen erzeugt werden können. Diese Routine identifiziert für jede noch offene Sitzung s_{offen} eine konfligierende Sitzung s_{schieb} im bereits gesetzten Teilplan und versucht, diese in eine andere Periode zu verschieben, um so Platz für zu s_{offen} schaffen. Die Auswahl von s_{schieb} erfolgt dabei nach dem Kriterium der minimalen „Widerstandszahl“ (resistance), die wesentlich von der zeitlichen Verfügbarkeit des betroffenen Lehrers abhängt. Allerdings wird s_{schieb} zugunsten von s_{offen} nur dann tatsächlich verschoben, wenn, ggf. durch weitere Verschiebungen anderer Sitzungen, die Zulässigkeit des Teilplans gewahrt werden kann. Ist eine solche Verschiebungskette nicht möglich, wird ein neues s_{schieb} bestimmt, und ein neuer Verschiebungsversuch wird gestartet. Der Prozess endet für eine offene Sitzung, wenn nach einer Kette von Operationen entweder die Verschiebung einer bereits erzeugten Sitzung zulässig möglich ist oder aber alle potenziellen Partner bis zu einem festgelegten Höchstwert der Widerstandszahl vergeblich getestet wurden.

Vergleichbare Ansätze der engpassorientierten Plansetzung mit Tauschroutine für die Einordnung ansonsten nicht zulässig setzbarer Unterrichtseinheiten wurden für den Schulbereich von Vejsada [1976], Bosler und Frangos [1974] und, parallel zu Berghuis, van der Heiden und Bakker [1964], von Barraclough [1965] entwickelt. Lediglich die Idee einer Tauschroutine, jedoch keine Implementation präsentiert de Gans [1981], dessen engpassorientiertes Setzverfahren ohne Tausch (Version 2) dennoch im Test mit 21 Schulen eine Zuordnungsrate von durchschnittlich 97,3% aller zu verplanenden Sitzungen erzielt.

Lazak [1966a, 1966b, 1969a, 1969b] präsentiert einen Setzalgorithmus für Schulen, der für die Bestimmung einer im bereits gesetzten Teilplan enthaltenen Sitzung s_{raus} zum Austausch gegen eine konfligierende, in den gegenwärtigen Teilplan nicht zulässig integrierbare Sitzung s_{rein} einen Kollisionsindex KO(s_{raus}) verwendet. Dieser zählt die Häufigkeit, mit der s_{raus} bereits in früheren Tauschoperationen aus dem gesetzten Teilplan herausgenommen wurde, und kann als Maß für das Konfliktpotenzial von s_{raus} verstanden werden. Neben KO(s_{raus}) wird die Nummer IS(s_{raus}) des Iterationsschrittes, in dem s_{raus} zuletzt in den gesetzten Teilplan hinein gelangt ist, bei der Auswahl berücksichtigt. IS(s_{raus}) und KO(s_{raus}) werden so zu einem Auswahlkriterium verknüpft, dass die Sitzung s_{raus} nur dann tatsächlich herausgenommen wird, wenn sie früh während des Planungsprozesses erzeugt wurde und in früheren Tauschrunden selten verdrängt wurde. Die Einbeziehung von IS(s_{raus}) in das Auswahlkriterium resultiert dabei aus der Annahme Lazaks, dass Setzungen im Sinne einer Konvergenz des Verfahrens umso „richtiger“ seien, je später im Verfahrensablauf sie vorgenommen wurden. Eine Bestätigung dieser These durch Lösung realer Setzungsprobleme blieb Lazak trotz umfangreicher Programmtests mit Hilfe künstlich erzeugter Fälle (vgl. Lazak [1969a und 1969b]) jedoch schuldig.

Für die Veranstaltungssetzung an Universitäten präsentieren Brittan und Farley [1971] eine Adaption des Ansatzes von Barraclough [1965]. Einen anderen Weg beschreitet der Ansatz von Almond [1969], der auf einer Vorlage von Almond [1966] beruht. Darin werden die zu setzenden Unterrichtseinheiten (courses) nach einem Gewichtungsfaktor geordnet, der sich – je nach Entwurfsentscheidung – an dem jeweiligen Konfliktpotenzial und/oder an anderen Zielgrößen orientiert. Die Setzung erfolgt nach Ordnung der Unterrichtseinheiten, wobei Doppelstunden vor Einzelstunden gesetzt werden. Die Auswahl der Perioden erfolgt in Reihenfolge benutzerseitig definierter Präferenzen. Können für eine Unterrichtseinheit u nicht genügend zulässig verfügbare Perioden gefunden werden, werden zunächst einige einfachere Restriktionen relaxiert und der Setzversuch wird wiederholt. Kann u auch dann nicht gesetzt werden, wird der Prozess abgebrochen und mit einer neuen Reihenfolge der Unterrichtseinheiten erneut gestartet, wobei den im vorigen Durchlauf als schwierig erwiesenen Unterrichtseinheiten eine höhere Priorität eingeräumt wird. Der Wechsel des Lösungspfades erfolgt hier demnach nicht über eine Tauschprozedur, sondern über einen Neustart vom Startknoten des Entscheidungsbaumes aus. Direkt angelehnt an den Ansatz von Almond sind die Arbeiten von Yule [1967] und Selim [1982 und 1983], ein ähnliches Verfahren findet sich bei Loo, Goh und Ong [1986]. Für die Stundenplansetzung an Schulen sieht Genrich [1966] bei periodenweiser Setzung und engpassorientierter Folge der Unterrichtseinheiten sowohl wiederholte Verfahrensdurchläufe mit veränderter Start-Priorisierung der Unterrichtseinheiten als auch eine einfache Tauschprozedur zwischen Sitzungen vor.

3.2.3.2 Engpassorientierte Zuordnung auf Basis graphischer Modelle

Eine Reihe konstruktiv-heuristischer Lösungsansätze für Setzungsprobleme wurde auf Basis graphischer Modelle entwickelt. Die am häufigsten gewählte Modellvariante ist dabei das Knotenfärbungsmodell im ungerichteten Graphen (vgl. Abschnitt 3.2.1.1.2). Sie wird von Mausser und Magazine [1996], Balakrishnan [1991], Mehta [1981], Wood [1969] und Welsh und Powell [1967] für die Setzung von Examensplänen, von Kiaer und Yellen [1992] für die Setzung an Universitäten genutzt. Als einzige verwenden Kiaer und Yellen in ihrem Ansatz eine Kantengewichtung, um das Ausmaß unvermeidbarer Konflikte bei Gleichfärbung adjazenter Knoten zu messen.

Alle genannten Ansätze betrachten stets den zuletzt erzeugten Knoten des Entscheidungsbaumes zuerst (LIFO-Strategie) und bedienen sich bei der Variablenauswahl eines Engpasskriteriums. So bestimmen Balakrishnan [1991], Wood [1969], Welsh und Powell [1967] und Mehta [1981], der sich eines von Brelaz [1979] entwickelten Algorithmus' bedient, die zu setzende Variable nach der Prioritätsregel des höchsten Grades des zugehörigen Graphenknotens. Der Grad eines Graphenknotens v gibt dabei an, wieviele Nachbarknoten er enthält, die eine von v unterschiedliche Färbung erhalten müssen. Kiaer und Yellen [1992] definieren und vergleichen verschiedene Dringlichkeitsmaße. Dabei liefert eine Funktion der bei Färbung eines Graphenknotens zu erwartenden Kosten, die durch Gleichfärbung mit einem oder mehreren Nachbarknoten entstehen, das beste Ergebnis hinsichtlich der Gesamtsumme der Veranstaltungskonflikte für die Studierenden. Demgegenüber orientieren sich Mausser und Magazine [1996] bei der Variablenwahl an der minimalen Anzahl für einen Knoten unter Beachtung der bereits vorgenommenen Färbungen noch konfliktfrei wählbarer Farben.

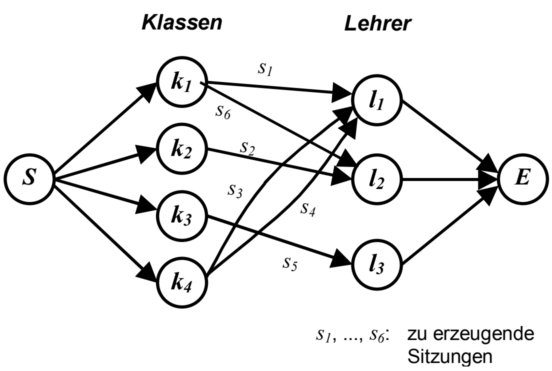
Die Wertewahl für die jeweils betrachtete Variable erfolgt bei Mehta [1981] derart, dass ihr die kleinste zulässige Periode zugewiesen wird, wobei nach oben zunächst keine Grenze gesetzt ist. Stellt sich am Ende des Färbungsprozesses heraus, dass die Zahl der zur Sicherung der Konfliktfreiheit benötigten Farben die für das Problem zulässige Maximalzahl P überschreitet, werden alle Graphenknoten mit einer Farbe größer P so umgefärbt, dass die Summe der für Prüflinge entstehenden Konflikte möglichst gering bleibt. In den Verfahren von Balakrishnan [1991], Wood [1969] und Welsh und Powell [1967] ist die Wertewahl der Variablenwahl vorgeschaltet. Hier werden, bei Periode 1 beginnend, periodenweise alle noch ungefärbten Graphenknoten in Reihenfolge oben beschriebener Prioritätsregel betrachtet und, soweit dies zulässig ist, mit der gerade betrachteten Periode gefärbt. Dabei ist die Zahl der verfügbaren Perioden nach oben offen, doch gibt es keine Regelung für den Fall, dass die bei konfliktfreier Färbung benötigte Periodenzahl den Wert P überschreitet. Das Verfahren von Kiaer und Yellen [1992] weist der gerade betrachteten Variable diejenige Periode zu, die die geringsten Konflikte für die Studierenden verursacht. Mausser und Magazine [1996] schließlich definieren eine Vorausschauregel, die unter den zulässigen Variablenwerten denjenigen auswählt, der zu der geringsten Einschränkung für die spätere Färbung noch ungefärbter Nachbarknoten führt.

Als einziger der bisher beschriebenen heuristischen Knotenfärbungsansätze implementieren Mausser und Magazine eine Backtracking-Prozedur, die es erlaubt, bei Unzulässigkeit eines Suchpfades auf einen anderen Pfad zu wechseln. Alle anderen Verfahren konstruieren nur einen einzigen Suchpfad, der entweder aufgrund der recht einfachen Zulässigkeitsanforderungen von vornherein zu einer zulässigen Lösung führt oder dessen unzulässiges Ergebnis nachträglich durch Umfärbung von Knoten in eine zulässige Lösung überführt wird (vgl. Mehta [1981]).

Ebenfalls zu den Knotenfärbungsansätzen für die Setzung von Veranstaltungsplänen gehört das Verfahren von Selim [1988]. Grundidee dieses Ansatzes ist die Spaltung von Graphenknoten, bis die chromatische Zahl des Problemgraphen auf die maximale Periodenzahl P reduziert ist. Die Spaltung eines Knotens impliziert dabei die Zerlegung des in der betroffenen Unterrichtseinheit enthaltenen Kurses in zwei Unterkurse, die zu unterschiedlichen Zeiten gegeben werden. Da jedoch eine solche Aufteilung von Unterrichtseinheiten im Zusammenhang der Setzungsproblematik an Schulen nicht zulässig ist, ist eine sinnvolle Adaption des Ansatzes von Selim nicht möglich.

Explizit für die Plansetzung an Schulen entwickelt wurde der Lösungsansatz von deWerra [1971], der von Ostermann und de Werra [1982] aufgegriffen und weiterentwickelt wird. Er basiert nicht auf einer Knotenfärbung, sondern auf einem Netzwerkflussmodell. Die Setzung erfolgt periodenweise, wobei für jede Periode ein eigenes Netzwerk betrachtet wird. Dieses stellt den Unterrichtsfluss von einem künstlichen Startknoten S über Klassen und Lehrer zu einem künstlichen Endknoten E dar (vgl. Abbildung 3.13). Jede Klasse ist mit dem Startknoten, jeder Lehrer mit dem Endknoten verknüpft. Sitzungen, die durch die Unterrichtsverteilung zwar vorgesehen, aber noch nicht erzeugt sind, werden durch eine Kante repräsentiert, die die betroffene Klasse und den betroffenen Lehrer verbindet, wobei implizit unterstellt wird, dass keine Unterrichtseinheit mehr als eine Klasse oder mehr als einen Lehrer bindet.

Abbildung 3.13: Netzwerk für die Setzung einer Periode
(Darstellung in Anlehnung an deWerra [1971])



Der Stundenplan für die betrachtete Periode wird durch eine Prozedur erzeugt, die Pfade in dem zugehörigen Netzwerk so auswählt, dass jede Kante des Netzwerkes auf maximal einem Pfad liegt. Auf diese Weise wird die Kollisionsfreiheit der Lösung sichergestellt. Da jeder Pfad eindeutig durch die Sitzung identifiziert wird, über deren Kante er läuft, ist die Auswahl der Netzwerkpfade identisch mit der Entscheidung, welche Sitzungen in der betrachteten Periode stattfinden sollen. Diese werden nach dem Vorausschaukriterium des geringsten Freiheitsgrades bestimmt, d.h. der geringsten Zahl für eine unverplante Sitzung noch verfügbarer Perioden. Bedingung ist jedoch, dass die so erzeugte Lösung den maximalen Netzwerkfluss, d.h. die maximal mögliche Anzahl auszuwählender Pfade, realisiert. Wie die meisten Knotenfärbungsansätze sehen auch Ostermann und de Werra in ihrem Verfahren keine Backtracking-Prozedur vor und nehmen so in Kauf, dass nicht alle durch die Unterrichtsverteilung vorgegebenen Sitzungen erzeugt werden können, der Plan also im Ergebnis unvollständig bleibt.

3.2.3.3 Heuristische Ansätze der Logischen Programmierung

Im Abschnitt 3.2.2.6 wurde die Logische Programmierung als ein Instrument vorgestellt, mit dessen Hilfe sich ein implizites Enumerationsverfahren auf Basis einer LIFO-Strategie für die Lösung von Setzungsproblemen formulieren und implementieren lässt. Nicht alle Ansätze der Logischen Programmierung nutzen jedoch diese Möglichkeit in einer Weise, die das Auffinden einer zulässigen oder sogar optimalen Lösung garantiert. Kang, von Schoenberg und White [1991] bilden das Setzungsproblem an Universitäten als Restriktionserfüllungsproblem (CSP)[31] ab und unterscheiden dabei zwischen „harten“ Restriktionen, z.B. Kollisionsfreiheit, Doppelunterrichtsverbot, Sperrungen und Fixierungen, und „weichen“ Restriktionen, z.B. Verwendung der bevorzugten Gebäude, Räume und Perioden sowie Verbot zweier Sitzungen desselben Dozenten in direkt aufeinander folgenden Perioden. Die Setzung erfolgt durch schrittweise Erzeugung der durch die Unterrichtsverteilung geforderten Sitzungen über die Identifikation jeweils zulässiger Zeiten und Räume. In die Zulässigkeitsprüfung fließen dabei sämtliche harten und weichen Restriktionen ein. Kann eine Sitzung s aufgrund der aktuellen Konstellation des bereits gesetzten Teilplans nicht zulässig erzeugt werden, werden die weichen Restriktionen in Reihenfolge aufsteigender Bedeutung eine nach der anderen aufgehoben, bis entweder eine Generierung von s möglich ist oder alle weichen Restriktionen vergeblich aufgehoben wurden. Tritt letzterer Fall ein, versucht der Algorithmus, durch Verschiebungen bereits erzeugter Sitzungen Raum für s zu schaffen. Gelingt auch dies nicht, wird s in einer Liste nach Abschluss der automatischen Setzung manuell einzuplanender Sitzungen gespeichert und der Algorithmus mit der nächsten Sitzung fortgesetzt.

Das Verfahren von Kang, von Schoenberg und White verwendet die Restriktionsprüfung nicht zum Zwecke der Constraint Propagation, also der Einschränkung der Wertebereiche noch ungesetzter Variablen, sondern es prüft stets nur die Zulässigkeit der aktuell vorgenommenen Setzung und nimmt so eine Erhöhung des Risikos in Kauf, in einen unzulässigen Lösungspfad hineinzulaufen. Die Lösungszeit wird für ein großes Beispiel von über 2.000 zu erzeugenden Sitzungen mit 25 Minuten auf einer Amdahl 2880 angegeben. Das Verfahren wird in späteren Beiträgen von Kang und White [1992] sowie Cheng et al.

[1996] wieder aufgegriffen, erfährt eine größere Ergänzung jedoch erst durch White und Zhang [1998], die es als Generator für die Startlösung eines Tabu Search-Verfahrens zur iterativen Lösungsverbesserung verwenden (vgl. Abschnitt 3.2.4.2). Ein Verfahren, das ähnlich wie der Ansatz von Kang, von Schoenberg und White Tauschroutinen zur Vermeidung von Unzulässigkeiten einsetzt, präsentiert David [1998] für die Setzung von Examensplänen. Auch hier wird bei Scheitern aller Tauschversuche die betroffene Sitzung für nicht automatisch generierbar erklärt und der manuellen Setzung anheim gegeben. Elmohamed, Coddington und Fox [1998] nutzen ein regelbasiertes Expertensystem für eine konstruktive Setzheuristik, mit der zulässige Startlösungen für verschiedene iterative Heuristiken erzeugt werden. Anwendungsgebiet ist die Veranstaltungssetzung an Universitäten.

Ebenfalls in den Universitätsbereich ist der Beitrag von Lajos [1996] einzuordnen. Auch hier wird auf ein Backtracking innerhalb des Entscheidungsbaumes verzichtet. Stattdessen wird mit Hilfe einer engpassorientierten Variablenwahl und der Constraint Propagation versucht, sukzessiv so viele Sitzungen wie möglich zulässig in den Stundenplan zu integrieren. Für den Fall, dass sich dies als unmöglich erweist, wird eine virtuelle Periode definiert, der eine sonst nicht generierbare Sitzung zugewiesen wird. Lajos konnte innerhalb von zehn Minuten auf einer Sun10/20 Workstation einen Veranstaltungsplan mit ca. 2.500 Sitzungen erzeugen, macht jedoch keine Angaben über die Zahl der verbliebenen manuell zu verplanenden Sitzungen.

Monfroglio [1988], der einen Ansatz für die Setzung von Schulstundenplänen entwickelt hat, kombiniert eine engpassorientierte Setzheuristik mit einer Tauschroutine, die drohende Unzulässigkeiten durch Verschiebungen im bereits gesetzten Teilplan zu beseitigen versucht. Kann für eine zu erzeugende Sitzung *s* trotz Verschiebung anderer Sitzungen keine zulässig wählbare Periode gefunden werden, wird *s* dennoch einer Periode zugeordnet, und die hierdurch entstehenden Restriktionsverletzungen werden als unausweichlich akzeptiert. Demnach wird stets eine vollständige, aber nicht unbedingt eine zulässige Lösung des Modells garantiert. Monfroglio gelang es, innerhalb von 38 Minuten auf einem 8086-PC einen akzeptablen („*well formed*"; Monfroglio [1988], S.20) Stundenplan für einen Testfall mit sechs Klassen, 23 Lehrern und zwölf Räumen zu erzeugen. Dabei flossen als Restriktionen die Vollständigkeit und Kollisionsfreiheit, das Doppelunterrichtsverbot und gleichmäßig lange Unterrichtstage der Klassen in das Modell ein. Die Raumzuordnung zu den Unterrichtseinheiten wurde als gegeben unterstellt. Zur Beschleunigung des Lösungsverfahrens zeigt Monfroglio die Möglichkeit der Parallelisierung von Restriktionsprüfungen auf. In Monfroglio [1996] verwendet der Autor seine Heuristik für die Dekodierung von Chromosomen innerhalb eines Genetischen Algorithmus' (vgl. Abschnitt 3.2.4.4).

Yoshikawa [1996] hat ein Verfahren entwickelt, welches Setzungsprobleme an Schulen betrifft, für die die Restriktionstypen Vollständigkeit, Kollisionsfreiheit, Doppelunterrichtsverbot bzw. minimaler Zeitabstand zwischen Sitzungen eines Faches und die Begren-

[31] Für eine Definition vgl. Abschnitt 3.2.1.1.1.

zung der täglichen Arbeitsbelastung für Lehrer relevant sind. Setzprinzip ist, ähnlich wie bei Lajos [1996], die durch eine Constraint Propagation unterstützte engpassorientierte Zuordnung ohne Backtracking. Nicht zulässig erzeugbare Sitzungen werden auch hier zunächst übergangen, jedoch in einem zweiten Verfahrensdurchlauf nachträglich in den Plan integriert. Dabei wird für jede dieser Sitzungen die Periode gewählt, deren Zuordnung die am geringsten bewertete Unzulässigkeit verursacht. Die so gewonnene Lösung wird als Startlösung einer Hillclimbing-Heuristik verwendet (vgl. Abschnitt 3.2.4.1)

3.2.3.4 Andere heuristische konstruktive Ansätze

Neben der engpassorientierten Setzung, die unter allen heuristischen konstruktiven Lösungsansätzen eindeutig die Hauptströmung darstellt, wurden einige Ansätze entwickelt, die Stundenpläne nach anderen Prinzipien zu konstruieren versuchen. So erzeugt Broder [1964] Examenspläne mit Hilfe einer Heuristik, die durch geschickte Variablen- und Wertewahl versucht, die Summe der durch den Plan implizierten Prüflingskonflikte möglichst gering zu halten. Dazu werden die einzelnen Prüfungseinheiten zunächst nach absteigender Summe potenzieller Prüflingskonflikte mit allen anderen Prüfungseinheiten sortiert und anschließend in dieser Reihenfolge gesetzt. Die Auswahl der jeweils zuzuordnenden Periode erfolgt nach dem Kriterium der geringsten Summe der durch die Zuordnung tatsächlich verursachten Prüflingskonflikte mit bereits gesetzten Prüfungseinheiten. Der Ansatz berücksichtigt nur die Vollständigkeitsrestriktion, so dass eine zulässige Lösung immer erreicht werden kann.

Ein präferenzgesteuertes Setzverfahren für die Veranstaltungsplanung an Universitäten präsentiert Tripathy [1992]. Darin werden Regeln der Variablen- und Wertewahl so verwendet, dass in dem unter Beachtung der Restriktionstypen Vollständigkeit und Kollisionsfreiheit erstellten Veranstaltungsplan diejenigen (Unter-)Kurse, an denen die meisten Studierenden beteiligt sind, den durch den Benutzer am meisten präferierten Perioden zugeordnet werden.

Für die Setzung von Schulstundenplänen hat Simma [1970] ein zielorientiertes Verfahren entwickelt, das lediglich Vollständigkeits- und Kollisionsfreiheitsrestriktionen berücksichtigt und die Raumdimension ausklammert. Das Verfahren läuft in zwei Stufen ab: Zunächst wird nach einer nicht näher spezifizierten Regel eine vorgegebene Anzahl an Kombinationsmustern identifiziert, die angeben, welche Lehrer-Klasse-Kombinationen derselben zunächst noch unbenannten Periode konfliktfrei zugeordnet werden können. Anschließend wird auf dem Weg der impliziten Enumeration eine Teilmenge der erzeugten Kombinationsmuster ausgewählt und so zu einem Stundenplan zusammengesetzt, dass die durch die Unterrichtsverteilung gegebenen Stundenanforderungen je Lehrer und Klasse durch den Plan genau erfüllt werden. Dabei wird eine nicht näher definierte Zielfunktion maximiert, die ein Ausdruck der Planqualität ist. Erreicht diese nicht das vom Benutzer geforderte Mindestniveau, so werden neue Kombinationsmuster erzeugt und den bereits bestehenden hinzugefügt. Anschließend wird ein neuer Plan gesetzt. Der Prozess wird so lange iteriert, bis die geforderte Mindestqualität erzielt ist. Eine Anwendung des Ansatzes von Simma auf reale Probleme wurde bislang nicht berichtet.

Ebenfalls für den Schulbereich präsentiert Fujino [1965] ein Setzverfahren, welches im Gegensatz zu den oben beschriebenen Ansätzen sowohl bei der Variablen- wie auch bei der Wertewahl nach dem Zufallsprinzip bei Gleichwahrscheinlichkeit der jeweiligen Alternativen vorgeht. Allerdings konnte dieses Verfahren für den von Fujino berichteten, aufgrund der Verwendung von Unterrichtsmustern (teaching patterns) mit der deutschen Praxis nur bedingt vergleichbaren Beispielfall nur 869 (= 94,5%) von insgesamt 921 Unterrichtsstunden erfolgreich zuordnen. Ein weiterer stochastischer Ansatz wurde in jüngerer Zeit von Drexl und Salewski [1997] konzipiert. Darin werden Periode für Periode so viele Sitzungen wie jeweils zulässig möglich in den Plan aufgenommen, wobei die Reihenfolge der Sitzungen zufallsbedingt ist[32]. Die Wahrscheinlichkeit für die Auswahl einer Sitzung s hängt dabei von ihrer Bewertung im Rahmen einer Prioritätsregel ab, z.B. den Kosten, die der Benutzer der Zuordnung von s zur gerade betrachteten Periode zumisst. Das Verfahren von Drexl und Salewski wurde bislang nur anhand zufallsgenerierter Probleme getestet, und es ist unklar, welche Restriktionen und Ziele bei diesem Test im Einzelnen berücksichtigt wurden. Auch dieses Verfahren garantiert selbst bei bekannter Lösbarkeit des Problems nicht, dass ein vollständiger Plan konstruiert werden kann.

3.2.3.5 Beurteilung

Die Aussichten, mit Hilfe einer konstruktiven Setzheuristik einen real zulässigen Stundenplan für eine allgemeinbildende Schule zu erzeugen, sind als äußerst schlecht zu beurteilen (vgl. Tabelle 3.8). Obwohl alle für diesen Bereich berichteten Ansätze Restriktionen bzgl. Randstunden, Wegezeiten zwischen Gebäuden oder freier Tage für Lehrer aus der Planerstellung ausklammern und nur Berghuis, van der Heiden und Bakker [1964] ein Springstundenverbot für Klassen einbeziehen, konnte keiner der Autoren für eine reale Schule einen vollständigen Stundenplan vorlegen. Diese Aussage gilt nicht nur für solche Ansätze, die sich mit der Verfolgung eines einzigen Lösungspfades begnügen, sondern auch für diejenigen, die ein Tausch- oder Backtrackingverfahren beinhalten, und schließt die Ansätze ein, welche mit Hilfe Logischer Programmiersprachen implementiert wurden. Da die für den Universitätsbereich und die Examensplanung entwickelten Verfahren jenen für den Schulbereich sehr ähnlich sind, ist auch von einer Übernahme von Verfahrensprinzipien aus diesen beiden Bereichen wenig Erfolg zu erwarten.

Obwohl keine der relevanten Zielgrößen in den berichteten Ansätzen, mit Ausnahme von Simma [1970], explizit gemessen wird, sondern Ziele allenfalls in die Variablen- bzw. Wertewahl integriert werden, ist die Formulierung einer Bewertungsfunktion, in die eine oder mehrere dieser Zielgrößen eingehen, durchaus denkbar. Sie kann jedoch nur dann sinnvoll verwendet werden, wenn das Verfahren die Möglichkeit beinhaltet, den beschrittenen Lösungspfad im Entscheidungsbaum zu wechseln. Dies erscheint wegen der hohen

[32] Die hier skizzierte Vorgehensweise betrifft Phase 2 des Algorithmus‘ TPGRM von Drexl und Salewski [1997]. Die vorgeschaltete Phase 1, welche zur automatischen Aufteilung des Stundenpensums der einzelnen Unterrichtseinheiten auf Einzel-, Doppel- oder Dreifachstunden eingesetzt wird und ebenfalls stochastisch konzipiert ist, wird hier nicht weiter betrachtet, da in dieser Arbeit unterstellt wird, dass Anzahl und Dauer der zu einer Unterrichtseinheit gehörenden Sitzungen bereits durch die Unterrichtsverteilung gegeben ist (vgl. Definition 2.3).

Wahrscheinlichkeit auftretender Unzulässigkeiten ohnehin unerlässlich. Die Potenzialeinschätzung für die konstruktiven Heuristiken ist daher von vornherein auf Verfahren beschränkt, die einen solchen Wechsel vorsehen.

Tabelle 3.8: Beurteilung *heuristischer konstruktiver Lösungsansätze*[33]
(zur Erläuterung der Kriterien vgl. Tabelle 2.11, S.42)

Kriterium	**bisherige Ansätze**	**Potenzialeinschätzung[1]**
Berücksichtigung der Restriktionstypen		
• Vollständigkeit (R-01):	ja	möglich
• Kollisionsfreiheit (R-02):	ja	möglich
• Sperrungen (R-03):	ja	möglich
• Springstundenverbot für Klassen (R-04):	ja[2, 3]	möglich
• Doppelunterrichtsverbot: (R-05):	ja	möglich
• Randstunden (R-06):	nein	möglich
• Fixierungen (R-07):	ja	möglich
• Wegezeiten (R-09):	nein	möglich
• Freie Tage für Lehrer (R-12):	nein	möglich
Wahrscheinlichkeit einer zulässigen Lösung bzgl. aller o.g. Restriktionstypen:		gering
Berücksichtigung der Zielsetzungen		
• Gleichlange Klassentage (Z-01):	nein	möglich
• Früher Unterrichtsbeginn (Z-02):	nein[3]	möglich
• Springstundenminimierung für Lehrer (Z-06):	nein	möglich
• Vertretungsbereitschaft (Z-10):	nein	möglich
• Fächerbeziehungen (Z-12):	nein	möglich
Handhabung der Mehrfachzielsetzung:	keine Bewertungsfunktion; Ziele werden teilw. bei Variablen- bzw. Wertewahl berücksichtigt	Zielgewichtung oder -hierarchie
Laufzeit:	kurz - mittel	kurz - lang

Anmerkungen:
[1] Potenzialeinschätzung für Verfahren, die in der Lage sind, den Lösungspfad zu wechseln (z.B. durch Tausch oder Backtracking)
[2] nur von Berghuis, van der Heiden und Bakker [1964] explizit einbezogen
[3] teilweise durch bevorzugte Ausnutzung früher Perioden unterstützt

Sofern der Definition einer konstruktiven Heuristik ein logisches Modell zugrunde gelegt wird,[34] gibt es keinen Grund zu bezweifeln, dass sämtliche Restriktionen durch das Verfahren berücksichtigt werden können. Die größte Schwierigkeit für den Erfolg eines solchen Verfahrens liegt daher nicht in einer mangelhaften Abbildungsgenauigkeit des ihm zugrunde liegenden Modells, sondern vielmehr in seiner Unfähigkeit, durch vorgenommene Setzungen verursachte Unzulässigkeiten frühzeitig zu erkennen und so eine aufwendige

[33] Im Gegensatz zu den Beurteilungen der exakten konstruktiven Ansätze (vgl. Abschnitt 3.2.2.7) und der heuristischen iterativen Ansätze (vgl. Abschnitt 3.2.4) wird bei der Beurteilung der heuristischen konstruktiven Ansätze nicht nach verschiedenen Verfahrenstypen differenziert. Der Grund für dieses Vorgehen liegt in der großen Ähnlichkeit, die diesen Ansätzen trotz unterschiedlicher Modellierungs- und Implementationsvarianten hinsichtlich der verwendeten Algorithmen zu Eigen ist.

[34] Zur Schwierigkeit der hinreichend genauen Abbildung mit Hilfe graphischer Modelle vgl. die Abschnitte 3.2.1.1.2 und 3.2.2.1.

Tiefensuche in unzulässigen Bereichen des Entscheidungsbaumes zu vermeiden. Das verbreitete Instrument der vorausschauenden engpassorientierten Auswahl der zu setzenden Unterrichtseinheiten und/oder der zuzuordnenden Perioden reicht zudem nicht aus, um mit ausreichender Wahrscheinlichkeit die Verfolgung eines unzulässigen Lösungspfades zu verhindern.

Ein Weg für die Entschärfung der Unzulässigkeitsproblematik besteht in der weitreichenden Relaxation von Problemrestriktionen durch das verwendete Modell. Dieser Weg birgt jedoch das hohe Risiko in sich, dass sich die aus dem Modell gewonnene Lösung für die Realität als untauglich erweist. Um diesem Risiko zu begegnen, haben zahlreiche Autoren iterative Heuristiken entwickelt, die Lösungen, welche mit Hilfe konstruktiver Heuristiken erzeugt wurden, durch schrittweise Modifikation zu verbessern versuchen. Sie sollen im nächsten Abschnitt näher beleuchtet werden.

3.2.4 Ansätze auf Basis heuristischer iterativer Verfahrenstypen

3.2.4.1 Hillclimbing

In der Systematik der iterativen Heuristiken stellen Hillclimbing-Verfahren die einfachste Verfahrenskategorie dar. Sie sind daran zu erkennen, dass sie stets nur einen offenen Lösungsknoten k betrachten, aus diesem je Iterationsschritt genau einen zulässigen Nachbarknoten nk erzeugen und anschließend genau dann $k := nk$ setzen, wenn die Bewertung $f(nk) < f(k)$ ist (vgl. Tabelle 3.9). Ist hingegen $f(nk) \geq f(k)$, wird entweder eine neue Nachbarlösung nk' aus k erzeugt und verglichen oder das Verfahren bricht ab, weil bereits alle Nachbarlösungen untersucht sind oder eine maximale Verweildauer am Knoten k erreicht wurde. k wird dann als „lokal optimale“ Lösung ausgegeben.

Tabelle 3.9: Charakterisierung des iterativen Verfahrenstyps *Hillclimbing*
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.2)

Entwurfsentscheidung	***Ausprägung***
|*OK*|	= 1
Nachbarschaftsdefinition	offen; häufig einzelne oder paarweise Modifikation von Variablenwerten
AnzGN	= 1
Bewertungsfunktion	offen; häufig Maß für den Abstand zur Zulässigkeit
Akzeptanzregel	$f(nk) < f(k)$
Löschungsregel	eindeutig, da $\|OK\| = 1$
Abbruchkriterium	$f(nk) \geq f(k) \wedge (N_{OK} \setminus EK = \emptyset \vee$ max. Iterationszahl an k erreicht)

Die Verbreitung reiner Hillclimbing-Algorithmen innerhalb der Stundenplanung ist wegen ihrer starren Verfolgung des Optimierungsziels relativ begrenzt. Für den Bereich der Examensplanung haben jedoch White und Chan [1978] einen Ansatz vorgelegt, der nach Konstruktion einer kollisionsfreien Aufteilung der Prüfungseinheiten auf die verfügbaren Perioden als Ausgangslösung eine iterative Heuristik anwendet, die die Summe der im

Prüfungsplan enthaltenen Sekundärkonflikte (second order conflicts), d.h. Zuordnungen verschiedener Prüfungen desselben Prüflings zu direkt aufeinanderfolgenden Perioden, verringert.[35] Dies geschieht in drei Phasen: Zunächst werden durch paarweisen Tausch die Prüfungsperioden permutiert. Anschließend werden einzelne Prüfungen kollisionsfrei von ihrer bisherigen in eine andere Prüfungsperiode verschoben. In der dritten Phase schließlich werden die Periodenzuordnungen der Prüfungseinheiten paarweise getauscht bzw. verschoben. Jede Phase endet, wenn durch die jeweilige Verfahrensvorschrift die Summe der Sekundärkonflikte nicht mehr weiter reduziert werden kann. Eine Anpassung dieses Verfahrens für die simultane Planung von Tages- und Abendprüfungen findet sich bei White und Haddad [1983].

Ein weiterer Hillclimbing-Ansatz zur Examensplanung wird von Desroches, Laporte und Rousseau [1978] für einen besonderen Anwendungsfall vorgestellt, in dem der Planungshorizont aus einer Menge von Prüfungstagen besteht, die jeweils genau zwei Prüfungsperioden, Vormittag und Nachmittag, umfassen. Auch hier besteht das Hauptziel darin, die Summe der Sekundärkonflikte für die Prüflinge zu minimieren, wobei benachbarten Perioden desselben Tages ein höheres Gewicht zukommt als solchen an aufeinanderfolgenden Tagen. Als weiteres Ziel wird angestrebt, möglichst viele Prüfungen am Vormittag stattfinden zu lassen. Ähnlich wie bei White und Chan werden mehrere Iterationsphasen durchlaufen, um die zuvor erzeugte vollständige und kollisionsfreie Ausgangslösung durch Reallokation einzelner Prüfungseinheiten und Permutation der Prüfungsperioden hinsichtlich der genannten Ziele zu verbessern.

Eine Weiterentwicklung des Ansatzes von Desroches, Laporte und Rousseau findet sich in Laporte und Desroches [1984].[36] Darin wird die Modellierung um die Einbeziehung einer beliebigen Anzahl von Perioden je Prüfungstag und um die Berücksichtigung – zuvor nicht enthaltener – räumlicher Beschränkungen erweitert. Statt der Summe von Sekundärkonflikten wird eine Kostenfunktion zu minimieren versucht, die über alle Prüflinge die gewichtete zeitliche Nähe zwischen den Prüfungseinheiten des jeweiligen Prüflings summiert und diese „Nähekosten“ (proximity cost) mit einer weiteren Größe addiert, welche die Abneigung des Benutzers gegenüber der Zuordnung einzelner Prüfungseinheiten zu bestimmten Prüfungslisten erfasst (aversion cost). Allerdings ist in diesem Ansatz die mehrstufige iterative Heuristik durch eine einstufige Heuristik auf Basis simpler Verschiebungen einzelner Prüfungseinheiten ersetzt. Dies mag auf die veränderte Modellstruktur zurückzuführen sein, in der der Unterscheidung zwischen benachbarten Perioden desselben Tages und solchen aufeinanderfolgender Tage eine verringerte Bedeutung zukommt. Die Autoren des Ansatzes geben selbst keine Begründung für diese Modifikation an.

---

35 Von der *Summe* der Konflikte zwischen Prüfungseinheiten wird im Folgenden stets gesprochen, wenn jeder Konflikt eines einzelnen Prüflings addiert wird. Hingegen wird von der *Anzahl* der Konflikte gesprochen, wenn als ein Konflikt der Fall zweier Prüfungseinheiten gezählt wird, deren Mengen teilnehmender Prüflinge sich überschneiden.

36 Eine Zusammenfassung dieser Weiterentwicklung geben Eiselt und Laporte [1987].

Hansen und Vidal [1995] haben das von Laporte und Desroches [1984] beschriebene Verfahren für die Erstellung landesweiter Examenspläne für dänische Sekundarschulen modifiziert. Dabei wurden neben zahlreichen anderen Anpassungen insbesondere in der konstruktiven Heuristik zur Erzeugung einer Startlösung kostenorientierte Verfahrensregeln durch zufallsorientierte ersetzt und so die Wahrscheinlichkeit für das schnelle Auffinden einer zulässigen Ausgangslösung erhöht. Weitere iterative Heuristiken für die Erstellung von Examensplänen wurden von Mausser und Magazine [1996] auf Basis deterministischer sowie von Ross und Corne [1995] auf Basis stochastischer Verfahrensentscheidungen entwickelt.

Für die Veranstaltungssetzung an Universitäten stellen Aubin und Ferland [1989] einen Ansatz vor, der auf einer Vorlage von Ferland und Roy [1985] basiert. Darin wird das Setzungsproblem als quadratisches Zuordnungsproblem formuliert, in dem jede mögliche Zuordnung einer Unterrichtseinheit zu einer Periode durch eine 0/1-Variable repräsentiert wird. Einziger unrelaxierter Restriktionstyp in dieser Formulierung ist die Vollständigkeit. Die Restriktionstypen Kollisionsfreiheit für Dozenten und Studierende, Ressourcensperrungen und Begrenzung von Raumkapazitäten werden hingegen relaxiert und zu einer gewichteten Zielfunktion integriert. Gelöst wird das quadratische Zuordnungsproblem mit Hilfe einer von Carlson und Nemhauser [1966] entwickelten Hillclimbing-Heuristik, die, beginnend mit einer beliebigen, einfach zu erzeugenden Startlösung, durch iterative Verschiebungen jeweils einer Zuordnung den Plan so lange modifiziert, bis auf diese Weise keine Verbesserung des Zielfunktionswertes mehr zu erzielen ist.

Eine Besonderheit des Ansatzes von Aubin und Ferland besteht darin, dass das Setzungsproblem wechselseitig mit dem Problem der Aufteilung der Studierenden auf Unterkurse (vgl. Abschnitt 2.5.1) betrachtet wird. Sobald eine neue Lösung des Setzungsproblems vorliegt, wird durch Verschiebungen einzelner Studierender zwischen zusammengehörigen Unterkursen versucht, die Zahl der Konflikte für die Studierenden weiter zu reduzieren. Gelingt dies, wird die iterative Setzheuristik erneut gestartet, um durch zeitliche Verschiebungen von Sitzungen abermals eine Konfliktreduktion zu erzielen. Die wechselseitige Lösung von Setzungsproblem und Gruppierungsproblem wird so lange iteriert, bis keine weitere Verbesserung der Zielfunktion mehr erreicht werden kann. Mit Hilfe dieses Verfahrens konnten Aubin und Ferland weitgehend konfliktfreie Veranstaltungspläne erzeugen. Eine Weiterentwicklung der Methodik der Periodenverschiebung auf allgemeinerer Basis findet sich bei Ferland und Lavoie [1992], doch wurde diese nur anhand zufallsgenerierter Probleme getestet.

Ähnlich wie Aubin und Ferland betrachten Sampson und Freeland [1995] bzw. Sampson, Freeland und Weiss [1995] die Teilprobleme der Veranstaltungssetzung und der Aufteilung der Studierenden auf Unterkurse in einem integrierten Ansatz, wobei jedoch die Unterkurszuteilung nicht als sequenzielles, sondern als Unterproblem der Veranstaltungssetzung behandelt wird. Auch hier wird eine Hillclimbing-Heuristik auf Basis einzelner Periodenverschiebungen eingesetzt, um den Studierenden den Besuch der von ihnen bevorzugten Veranstaltungen weitestgehend zu ermöglichen.

Den einzigen Ansatz zur Erstellung von Schulstundenplänen mit Hilfe eines Hillclimbing-Verfahrens liefert Krins [1981]. Dieser erzeugt über eine engpassorientierte konstruktive Heuristik zunächst eine Startlösung, die jeder Periode-Klasse-Kombination (= Variablen) entweder eine Lehrer-Fach-Raum-Kombination oder eine Freistunde als Wert zuordnet. Dabei werden lediglich die Restriktionen der Vollständigkeit und der Kollisionsfreiheit für Klassen berücksichtigt. Alle übrigen Restriktionen sowie eine Reihe von Zielsetzungen werden nach ihrer Bedeutung gewichtet und zu einer Inkompatibilitätsfunktion (= Bewertungsfunktion) addiert, die als Maß für die Entfernung der beurteilten Lösung von der realen Akzeptanzfähigkeit dient. Mit der so konstruierten Startlösung beginnend, versucht die iterative Heuristik, schrittweise aus der jeweiligen aktuellen Lösung *k* durch eine Reihe von Tauschvorgängen zwischen jeweils zwei Zuordnungen eine Nachbarlösung *nk* zu generieren, die eine geringere Inkompatibilität als *k* aufweist. Das Verfahren bricht ab, wenn für jede Zuordnung im zuletzt erzeugten Stundenplan gilt, dass sie entweder alle Restriktionen hinreichend erfüllt oder aber durch ihre Modifikation eine Planverbesserung bereits vergeblich versucht wurde. Mit seinem Verfahren konnte Krins für mehrere Schulen innerhalb weniger Minuten auf einer CDC CYBER 76 Lösungen erzeugen, doch wiesen diese stets einige verbliebene Unzulässigkeiten auf, so dass manuelle Nachkorrekturen unvermeidlich waren.

3.2.4.2 Tabu Search

Eine Möglichkeit, das mit Hillclimbing-Heuristiken verbundene „Hängenbleiben“ in einem lokalen Minimum zu umgehen, besteht in der Verwendung eines Tabu Search-Verfahrens (vgl. Tabelle 3.10).[37]

Tabelle 3.10: Charakterisierung des iterativen Verfahrenstyps *Tabu Search*
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.2)

Entwurfsentscheidung	***Ausprägung***
$\|OK\|$	= 1
Nachbarschaftsdefinition	offen; häufig einzelne oder paarweise Modifikation von Variablenwerten
AnzGN	$= \|N_{OK}\|$
Bewertungsfunktion	offen; häufig Maß für den Abstand zur Zulässigkeit
Akzeptanzregel	$f(nk) = \min\{f(nk'): nk' \in GN \wedge nk'$ nicht tabu$\} \wedge$ (nk nicht tabu $\vee f(nk) <$ Grenzkriterium A)
Löschungsregel	eindeutig, da $\|OK\| = 1$
Abbruchkriterium	max. Iterationszahl ohne Verbesserung von f_{best} bzw. generelle max. Iterationszahl erreicht

Der Unterschied zum Hillclimbing besteht vor allem in zwei Punkten. Zum einen werden, ausgehend vom aktuellen Knoten *k*, in einem Iterationsschritt nicht nur eine, sondern i.d.R. alle auf Basis der verwendeten Nachbarschaftsdefinition konstruierbaren Nachbarlösungen

[37] Eine umfassende, allgemeine Einführung in die Methodik des Tabu Search gibt Glover [1989 und 1990].

erzeugt. Zum anderen wird jene Nachbarlösung, die unter allen Nachbarlösungen von *k* den niedrigsten Zielfunktionswert aufweist, immer als Ersatz für *k* akzeptiert, und zwar auch dann, wenn *k* selbst einen noch niedrigeren Zielfunktionswert aufweist. Ausgenommen werden dabei allerdings solche Nachbarlösungen, die aufgrund ihres Eintrags in eine sog. Tabu-Liste verboten sind. Diese Liste enthält alle akzeptierten Knoten der letzten $n > 0$ Iterationen. Sie dient dazu, Zyklen im Lösungsprozess zu vermeiden. Alternativ zu konkreten Lösungen können in der Tabu-Liste allerdings auch Nachbarschaftsoperationen gespeichert sein, die für die Erzeugung der in den letzten *n* Iterationen akzeptierten Lösungen verwendet wurden. In diesem Fall werden neue Nachbarlösungen ausgeschlossen, wenn sie durch Verwendung einer der in der Tabu-Liste gespeicherten Operationen erzeugt wurden. Dies gilt auch, wenn sie nicht mit zuvor generierten Lösungen identisch sind.

Um nun durch die Tabu-Liste die Verfolgung guter Lösungspfade nicht zu blockieren, ist es üblich, ein Grenzkriterium (aspiration function) *A* zu definieren und einen aufgrund eines entsprechenden Eintrags in der Tabu-Liste eigentlich verbotenen Nachbarknoten doch zu akzeptieren, sofern seine Bewertung unterhalb von *A* liegt. Häufig wird *A* dabei mit dem Wert der besten bislang gefundenen Lösung gleichgesetzt.[38] Das Verfahren terminiert, wenn nach einer vorgegebenen Anzahl an Iterationen keine Verbesserung gegenüber der besten gefundenen Lösung mehr erzielt werden kann oder wenn die generell vorgegebene maximale Iterationszahl erreicht ist.

Das erste Tabu Search-Verfahren für die Lösung von Setzungsproblemen wurde von Hertz [1991] entwickelt und sowohl für die Veranstaltungssetzung als auch für die Examenssetzung eingesetzt. Eine Lösung wird dabei als im Sinne des Verfahrens zulässig betrachtet, wenn sie neben der (implizit unterstellten) Vollständigkeit sämtliche Sperrungen und Fixierungen berücksichtigt. Als zulässig hinsichtlich des Problems wird sie eingestuft, wenn sie zusätzlich keine zeitlichen Kollisionen für Studierende (bzw. Prüflinge) und Dozenten (bzw. Prüfer) aufweist und Wegezeiten zwischen Gebäuden angemessen berücksichtigt. Die Bewertung jeder verfahrensmäßig zulässigen Lösung erfolgt durch eine gewichtete Summe ihrer Verletzungen der relaxierten Problemrestriktionen. Nachbarlösungen sind alle Lösungen, die durch zeitliche Verschiebung einer einzelnen Sitzung erzeugt werden können. Mit seinem Verfahren konnte Hertz den Veranstaltungsplan für eine große Universitätsfakultät mit 288 Kursen, 143 Dozenten, 1.729 Studierenden und 67 auf 15 Gebäude verteilten Räumen so lösen, dass nur 69 Kurskollisionen für Studierende entstanden und 116 Verletzungen von Wegezeitenrestriktionen vorlagen. Ähnlich erfolgreich verlief die Setzung des Examensplans einer anderen Universitätsinstitution mit 1.851 Prüfungseinheiten. Eine Adaption seines Ansatzes für ein azyklisches Setzungsproblem präsentiert Hertz [1992].

Ebenfalls für ein Problem der Veranstaltungssetzung entwickelt wurde der Ansatz von White und Zhang [1998], in dem Tauschoperationen bezüglich der Zeit- und Raumzuordnungen jeweils zweier Sitzungen für die Nachbarschaftsdefinition verwendet werden.

[38] Die Verwendung von *A* in dieser Form macht nur Sinn, wenn die Tabu-Liste nicht konkrete Lösungen, sondern bestimmte Nachbarschaftsoperationen enthält, da das Wiederauffinden einer bereits untersuchten Lösung nie zu einer Verbesserung der aktuell besten Bewertung führen kann.

Auch hier wird die Zielfunktion zur weitgehenden Problemrelaxation, u.a. bezüglich Kollisionsfreiheit, Raumkapazitäten und Fixierungen, verwandt, so dass am Ende des Lösungsprozesses mit verbliebenen Unzulässigkeiten gerechnet werden muss. Allerdings wird dieses Risiko durch die Zuhilfenahme der heuristischen Logischen Programmierung (vgl. Abschnitt 3.2.3.3) bei der Konstruktion der Startlösung erheblich gemindert. Mit ihrem Verfahren ist es White und Zhang gelungen, einen mit 262 Kursen relativ großen Fall ohne verbleibende Unzulässigkeiten hinsichtlich der relaxierten Problemrestriktionen zu lösen. Für zwei weitere, kleinere Fälle gelang dies jedoch nicht. In einem Fall brachte das Tabu Search-Verfahren nicht einmal eine Verbesserung gegenüber der Startlösung. Einen weiteren Ansatz für die Setzung von Examensplänen, der auf ähnlichen Prinzipien basiert und mit ähnlichen Risiken belastet ist wie die vorgenannten, haben Boufflet und Nègre [1996] vorgestellt.

Die früheste Anwendung des Tabu Search im Bereich der Stundenplansetzung an Schulen stammt von Costa [1994]. Sie enthält eine Relaxation der Restriktionstypen Kollisionsfreiheit und Springstundenverbot für Lehrer (nicht für Klassen!) sowie einiger weiterer Restriktionstypen, u.a. bezüglich Raumvorbereitungszeiten und der Kapazität der Schulmensa. Nicht relaxiert werden die Restriktionstypen Vollständigkeit, Fixierungen, Sperrungen, Wegezeiten und gleichmäßige Verteilung des Unterrichts eines Faches über die Woche. Auch werden bestimmte unerwünschte Fächerfolgen vermieden und eine Mittagspause berücksichtigt. Unklar ist, inwieweit das nicht explizit einbezogene Springstundenverbot für Klassen durch eine Gleichheit von je Klasse zu setzender Stundenzahl und der Zahl verfügbarer Perioden implizit gesichert ist. Wie in den vorgenannten Ansätzen, werden auch hier die relaxierten Restriktionen zu einer gewichteten linearen Zielfunktion integriert. Die Nachbarschaftsdefinition erfolgt über die zeitliche Verschiebung einzelner Sitzungen. Die von Costa für zwei Schulen mit 32 bzw. 12 Klassen erzeugten Lösungen wiesen Unzulässigkeiten hauptsächlich hinsichtlich der geforderten Springstundenfreiheit für Lehrer auf, doch erschienen diese akzeptabel, zumal von Hand keine bessere Lösung erzeugt werden konnte.

Weitere Tabu Search-Ansätze wurden von Colorni, Dorigo und Maniezzo [1998] für die Setzung italienischer sowie von Alvarez-Valdes, Martin und Tamarit [1996] für die Setzung spanischer allgemeinbildender Schulen entwickelt. Wright [1996] behandelt den Fall einer englischen Schule, in der einige Unterrichtseinheiten in vierzehntägigem Rhythmus anfallen. Ebenfalls anhand italienischer Beispielfälle wurde das Verfahren von Schaerf [1996] getestet, das als Besonderheit eine Kombination von Tabu Search und stochastischem Hillclimbing aufweist. Dabei wird das Tabu Search gestartet, wenn mit Hilfe der Hillclimbing-Routine, die jeweils *eine* zufällig gewählte Tauschoperationen ausführt und bei Nicht-Verschlechterung der Bewertungsfunktion akzeptiert, über eine vorgegebene Iterationszahl hinweg keine Verbesserung mehr erzielt werden kann. Umgekehrt wird zum stochastischen Hillclimbing zurückgekehrt, wenn das Tabu Search stagniert. Der Wechsel zwischen beiden Verfahren wird fortgesetzt, bis insgesamt keine Verbesserung mehr zu erreichen ist. Mit seinem Ansatz konnte Schaerf in ein- bis knapp fünfstündiger Laufzeit auf einer Silicon Graphics INDY Workstation für drei Praxisfälle Stundenpläne erzeugen, die eine bessere Bewertung als das jeweils per manueller Setzung erzielte Vergleichsergebnis aufwiesen, wobei in die Bewertungsfunktion sowohl Restriktionsrelaxationen als auch originäre Zielsetzungen einflossen.

3.2.4.3 Simulated Annealing

Ein stochastisches Verfahren der Nachbarschaftssuche ist das von Kirkpatrick, Gelatt und Vecchi [1983]) entwickelte Simulated Annealing (vgl. Tabelle 3.11).[39]

Tabelle 3.11: Charakterisierung des iterativen Verfahrenstyps ***Simulated Annealing***
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.2)

Entwurfsentscheidung	***Ausprägung***
|*OK*|	= 1
Nachbarschaftsdefinition	offen; häufig einzelne oder paarweise Modifikation von Variablenwerten
AnzGN	= 1
Bewertungsfunktion	offen; häufig Maß für den Abstand zur Zulässigkeit
Akzeptanzregel	$f(nk) < f(k)$, sonst Akzeptanz mit Wahrscheinlichkeit $p = e^{-(f(nk) - f(k)) / T}$
Löschungsregel	eindeutig, da |*OK*| = 1
Abbruchkriterium	max. Iterationszahl ohne Verbesserung von f_{best} bzw. generelle max. Iterationszahl erreicht

Ähnlich wie beim Hillclimbing, wird auch durch das Simulated Annealing je Iteration stets nur eine Nachbarlösung *nk* des aktuellen Lösungsknotens *k* betrachtet, doch wird diese mit einer Wahrscheinlichkeit $p \in (0; 1]$ auch dann akzeptiert, wenn sie eine schlechtere Bewertung als *k* aufweist. *p* wiederum bestimmt sich nach dem Abstand der Bewertungen von *k* und *nk* durch die Formel

$$p = e^{-[f(nk) - f(k)] / T},$$

wobei $T > 0$ einen Steuerungsparameter, die sog. Auskühltemparatur (cooling temperature) darstellt. *T* wird in der Startphase des Lösungsprozesses zunächst auf einen hohen Wert festgelegt, der dazu führt, dass selbst ein Knoten *nk*, der eine große Bewertungsdifferenz gegenüber *k* aufweist, mit hoher Wahrscheinlichkeit als neuer offener Knoten übernommen wird. Im Verlauf des Lösungsprozess wird *T* jedoch schrittweise gesenkt, so dass mehr und mehr nur noch solche Nachbarlösungen eine hohe Akzeptanzwahrscheinlichkeit erhalten, die eine sehr geringe oder keine Bewertungsdifferenz zur aktuellen Lösung aufweisen. Als Senkungsformel hat dabei die exponentielle Auskühlung (exponential cooling) eine hohe Verbreitung gefunden, die nach jeweils $n > 0$ Verfahrensiterationen *T* mit einem konstanten Faktor $\alpha \in (0; 1)$ multipliziert und *T* so zu Beginn des Verfahrens schnell, im spä-

[39] Eine umfassende Darstellung und Kritik des Simulated Annealing gibt Ingber [1993]. Eine kurze Einführung in das Grundkonzept findet sich bei Dowsland [1995].

teren Verlauf hingegen immer langsamer senkt. Das Verfahren bricht ab, wenn nach einer vorgegebenen Zahl von Iterationen keine Verbesserung mehr erzielt werden kann oder ein generelles Iterationslimit erreicht ist.

Für die Stundenplanerstellung an Schulen haben Abramson [1991] und Abramson und Dang [1993] einen ersten Lösungsansatz auf Basis des Simulated Annealing entwickelt. Darin werden, mit Ausnahme der Vollständigkeit, sämtliche Restriktionstypen relaxiert und zu einer gewichteten linearen Zielfunktion integriert. Abramson [1991] regelt die Nachbarschaftsdefinition über die zeitliche Verschiebung einzelner Sitzungen, wobei die Auswahl der Sitzung und der neuen Periode zufällig erfolgt. Die Startlösung wird durch zufällige Periodenzuordnung zu jeder Sitzung erzeugt. Obwohl Abramson die mögliche Einbeziehung von bevorzugten Unterrichtszeiten und einer Begrenzung der täglichen Unterrichtsbelastung für Lehrer in die Zielfunktion erwähnt, wurden in den Testläufen des Verfahrens nur Kollisionsfreiheitsrestriktionen berücksichtigt. Die Raumzuordnung zu den Unterrichtseinheiten wurde als bereits gegeben unterstellt. Gelöst wurden neun zufallsgenerierte Probleme sowie ein Praxisfall, wobei in einem der künstlichen Fälle trotz 90-minütiger Laufzeit auf einer Sun 3/60 einige Unzulässigkeiten verblieben. Der Praxisfall konnte innerhalb von 14 Stunden zulässig gelöst werden. Eine Adaption des von Abramson vorgestellten Ansatzes mit einer um verschiedene didaktische und organisatorische Zielsetzungen erweiterten Bewertungsfunktion verwenden Colorni, Dorigo und Maniezzo [1998] für die Setzung italienischer Schulen und berichten, eine Verbesserung gegenüber manuell gesetzten Plänen erzielt zu haben.

Abramson und Dang [1993] führen zwei grundlegende Modifikationen gegenüber dem vorherigen Ansatz von Abramson ein. Zum ersten wird die exogene Benennung eines konkreten Raumes für jede Unterrichtseinheit durch die Spezifikation einer Raumgruppe ersetzt. Hierdurch können die Kollisionsfreiheitsrestriktionen für die einzelnen Räume durch Kapazitätsrestriktionen für die Raumgruppen abgelöst und so die Planungsflexibilität gesteigert werden. Zum zweiten wird in die Nachbarschaftsdefinition als Alternative zur o.g. Periodenverschiebung ein Lehrertausch zwischen jeweils zwei Unterrichtseinheiten eingeführt, soweit dies fachlich vertretbar erscheint. Dabei erfolgt die Entscheidung, welche Operation in der jeweils aktuellen Verfahrensiteration ausgeführt wird, über eine Zufallsauswahl unter Gleichwahrscheinlichkeit beider Alternativen. Im Gegensatz zu Abramson [1991] werden ausschließlich zufallsgenerierte Beispiele betrachtet, die so konstruiert sind, dass die Unterrichtsbelastung jedes Lehrers, jeder Klasse und jedes Raumes genau der Anzahl der Planungsperioden entspricht. Obwohl für die Lösung dieser Fälle spezielle, eigens für den Simulated Annealing-Algorithmus entwickelte Hardware verwendet wurde, konnte nicht immer ein zulässiger Plan erzielt werden. Das Ergebnis lag zwar umso besser, je näher die Abkühlungsrate α an den Wert 1 heranreichte, doch mussten hierfür Laufzeiten von bis zu sieben Stunden in Kauf genommen werden.

Ein weiterer Simulated Annealing-Ansatz für die Setzung von Schulstundenplänen stammt von Dige, Lund und Ravn [1993]. Einbezogen werden darin die Restriktionstypen Vollständigkeit, Kollisionsfreiheit, Doppelunterrichtsverbot, Sperrungen und Springstundenverbot für Klassen und für Lehrer, eine Begrenzung der täglichen Arbeitsbelastung für

Lehrer sowie ein spätester Tagesbeginn und spätestes Tagesende für den Klassenunterricht. Die Raumzuordnung zu Unterrichtseinheiten wird als gegeben unterstellt. Wie bei Abramson werden alle Problemrestriktionen bis auf die Vollständigkeit relaxiert, wobei zwischen „harten" Restriktionen mit hohem Zielgewicht und „weichen" Restriktionen mit geringem Zielgewicht differenziert wird. Die Nachbarschaftsdefinition erfolgt über den Periodentausch jeweils zweier Sitzungen. Das Verfahren beginnt mit einer Startlösung, die das Resultat einer zufälligen, klassenweise unter Einbeziehung der jeweils gültigen Restriktionen erzeugten Periodenzuordnung aller Unterrichtseinheiten ist. Anhand eines Beispielfalles mit 14 Klassen, 29 Lehrern und 32 Räumen stellen die Autoren die Ergebnisse zahlreicher Testläufe mit verschiedenen Parametereinstellungen für T bzw. α sowie verschiedenen Strategien für die Auswahl der zu tauschenden Sitzungen vor. Sie kommen zu dem Schluss, dass eine adaptive Parameteranpassung, bei der eine Senkung von T von der Entwicklung der Lösungsbewertung abhängig gemacht wird, die beste Strategie darstelle, da sie relativ unsensibel gegenüber den Initialisierungswerten von T und α sei. Die Zeit bis zur Erreichung einer hinsichtlich zumindest der harten Restriktionen zulässigen Lösung geben Dige, Lund und Ravn mit ca. sechs Stunden auf einem 80386 bzw. 80486 PC mit 20 bis 40 MHz an.

Simulated Annealing-Ansätze wurden auch für die Setzung von Examensplänen entwickelt und anhand praktischer Fälle erfolgreich getestet (vgl. Bullnheimer [1998], Thompson und Dowsland [1996], Dowsland [1995], Ross und Corne [1995], Johnson [1990]). Sie bedienen sich ähnlicher Prinzipien wie die bereits erläuterten Verfahren zur Lösung des Schulproblems, kommen jedoch aufgrund der Tatsache, dass neben der Vollständigkeit i.d.R. lediglich Kollisionsfreiheits- und Raumkapazitätsrestriktionen berücksichtigt werden müssen, ohne Relaxation aus, so dass die Bewertungsfunktion für die Verwirklichung originärer Zielsetzungen wie die Minimierung von Sekundärkonflikten, die möglichst frühe Durchführung von Examina mit hoher Teilnehmerzahl oder die aus Sicht der Prüflinge möglichst gleichmäßige Verteilung der Prüfungen über den Planungshorizont eingesetzt werden kann.

Für die Veranstaltungssetzung an Universitäten haben Elmohamed, Coddington und Fox [1998] ein Simulated Annealing-Verfahren implementiert und sich dabei, analog zu den Ansätzen für die Setzung an Schulen, einer weitreichenden Problemrelaxation bedient. Eine Besonderheit dieses Ansatzes besteht in der Verwendung eines Expertensystems, mit dessen Hilfe Unzulässigkeiten innerhalb eines (Teil-)Planes ermittelt werden können. Es wird sowohl für die Generierung einer (unvollständigen) Startlösung als auch für die Identifikation von günstigen Tauschoperationen zwischen den Raum- und/oder Periodenzuordnungen jeweils zweier Sitzungen eingesetzt. Da die Autoren auch die Vollständigkeit relaxieren, ist für sie der Anteil der bei Verfahrensabschluss tatsächlich im Plan enthaltenen Sitzungen an allen zu erzeugenden Sitzungen ein wesentliches Gütekriterium für das Verfahren. Für drei große Beispielfälle mit bis zu 13.653 Studierenden, 1.200 Professoren, 509 Räumen und 3.839 Sitzungen ist es ihnen mit Hilfe einer von der Entwicklung der Bewertungsfunktion abhängigen Annealingstrategie, die phasenweise auch zu einer Erhöhung des Wertes von T (reheating) führt, gelungen, innerhalb von zehn bis 20 Stunden auf einer Unix-Workstation einen vollständigen und für Dozenten und Räume, nicht aber für die Studierenden, kollisionsfreien Plan zu erzeugen. Weitere, vergleichbare Ansätze des Simulated Annealing für den Universitätsbereich wurden von Dowsland [1990 und 1993] entwickelt.

3.2.4.4 Genetische Algorithmen

Wesentlich komplexer strukturiert als die oben geschilderten Verfahren des Hillclimbing, Tabu Search oder Simulated Annealing sind die sog. Genetischen Algorithmen (vgl. Tabelle 3.12).[40]

Tabelle 3.12: Charakterisierung des iterativen Verfahrenstyps
Genetischer Algorithmus
(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.2)

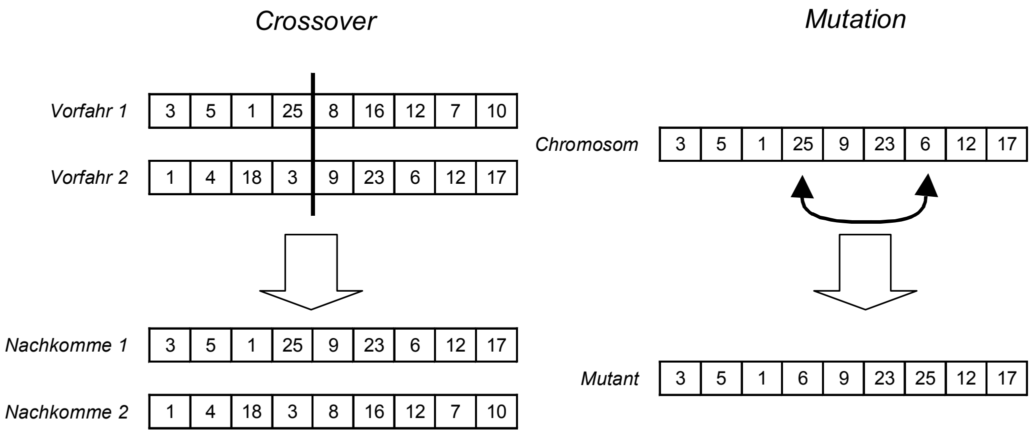
Entwurfsentscheidung	***Ausprägung***
|*OK*|	= $N >> 1$
Nachbarschaftsdefinition	über einen oder mehrere der genetischen Operatoren **Crossover** (Rekombination zweier alter Lösungen zu einer neuen Lösung), **Mutation** (Modifikation einer Lösung) und **Reparation** (Wiederherstellung der Zulässigkeit nach Durchführung eines Crossover oder einer Mutation)
AnzGN	≥ 1
Bewertungsfunktion	offen; häufig Maß für den Abstand zur Zulässigkeit; häufig als zu maximierende Fitness-Funktion (fitness function) formuliert
Akzeptanzregel / Löschungsregel	Nachkommen ersetzen ihre Eltern, Mutanten ihr Ausgangschromosom; Löschung außerdem implizit über **Reproduktion**
Abbruchkriterium	max. Iterationszahl ohne Verbesserung von f_{best} bzw. generelle max. Iterationszahl erreicht

Genetische Algorithmen basieren auf einem naturwissenschaftlichen Modell, das eng an die Genetik angelehnt ist. Darin wird jede erzeugte Lösung als ein Chromosom betrachtet, das sich aus einer Sequenz von Genen zusammensetzt, von denen wiederum jedes einen Wert, Allele genannt, aus einem vorgegebenen diskreten Wertebereich annimmt. Jedes Gen stellt demnach eine Entscheidungsvariable innerhalb des Modells dar. Im Unterschied zu den vorgenannten iterativen Lösungsverfahren, halten Genetische Algorithmen stets nicht nur einen, sondern eine größere Zahl N offener Knoten des Entscheidungsbaumes als „Chromosomen-Population".

Als (genetische) Nachbarschaftsoperation steht zunächst die Rekombination zweier Eltern-Chromosomen aus der Population zu zwei oder einem Nachkommen, Crossover genannt, zur Verfügung. Alternativ oder ergänzend kommt die nach einer verfahrensindividuellen Regel gestaltete Modifikation einzelner Chromosomen, die Mutation, in Frage (vgl. Abbildung 3.14).

[40] Eine ausführliche Einführung in Genetische Algorithmen geben Goldberg [1989], Davis [1991] und Reeves [1997].

Abbildung 3.14: Beispiele für Crossover und Mutation



Zwei Chromosomen werden zufällig aus der Population ausgewählt und vor dem fünften Gen miteinander gekreuzt.

Ein Chromosom wird zufällig aus der Population ausgewählt und durch Vertauschen der Allelen von Gen 4 und Gen 7 mutiert.

Sofern durch Crossover oder Mutation ungewollte Unzulässigkeiten entstehen, können genetische Reparationsoperatoren definiert sein, die diese Unzulässigkeiten beseitigen und so die Überlebensfähigkeit der Nachkommen steigern. Diese wird durch die Bewertungsfunktion (fitness function) gemessen, die i.d.R. als Maximierungsgröße ausgelegt ist. Sie wird oft für die Relaxation einer Reihe von Restriktionstypen verwandt, kann aber auch originäre Zielkriterien enthalten. Chromosomen, die aus einer genetischen Operation entstanden sind, ersetzen ihre direkten Vorfahren. Wird bei einem Crossover nur ein Nachkomme generiert, ersetzt dieser i.d.R. den Elternteil mit der geringeren Fitness. Meist werden in einer Verfahrensiteration nicht alle verfügbaren genetischen Operatoren auf jedes Chromosom angewendet, sondern es werden anhand voreingestellter Wahrscheinlichkeiten oder Regeln ein oder mehrere Operatoren sowie ein oder mehrere Chromosomen ausgewählt. Nach Abschluss aller Nachbarschaftsoperationen einer Iteration wird durch ein Zufallsexperiment, Reproduktion genannt, eine neue Generation erzeugt, die an die Stelle der bestehenden Population tritt. Dies geschieht durch N-maliges Ziehen von Chromosomen aus der bestehenden Population, wobei standardmäßig die Auswahlwahrscheinlichkeit jedes Chromosoms dem Quotienten aus seiner Fitness und der summierten Fitnesswerte aller Chromosomen der bestehenden Population entspricht. Die fittesten Chromosomen haben so die höchste Überlebenswahrscheinlichkeit. Das Ziehen erfolgt mit Zurücklegen, so dass einige Chromosomen der alten Generation in der neuen mehrmals, andere hingegen gar nicht vertreten sein können.

Genetische Algorithmen haben sich in den vergangenen zehn Jahren zum wohl populärsten Ansatz für die Lösung von Setzungsproblemen entwickelt. Ihren Einsatz im Schulbereich haben erstmals Colorni, Dorigo und Maniezzo [1991, 1992 und 1998] berichtet. In ihrem Ansatz werden die Chromosomen als Matrizen definiert. Jede Zeile einer solchen Chromosomen-Matrix steht für einen Lehrer. Die Zahl der Gene in jeder Zeile entspricht der Anzahl der verfügbaren Planungsperioden. Die Allele eines beliebigen Gens t in Zeile l

bezeichnet die Klasse, die Lehrer *l* in Periode *t* unterrichtet, oder aber eine Freistunde oder eine andere Beschäftigung des Lehrers *l* in *t*. Eine Lösung gilt als zulässig, wenn sie sowohl vollständig als auch kollisionsfrei ist und keine Springstunden für die Klassen aufweist. Letztere können allerdings zumindest in dem 1998 ausführlich geschilderten Fall[41] nur dann auftreten, wenn der Plan für die betreffende Klasse gleichzeitig Kollisionen enthält.

Die von Colorni, Dorigo und Maniezzo betrachtete lineare, gewichtete Fitness-Funktion bezieht sowohl Unzulässigkeiten in Bezug auf die o.g. Restriktionen als auch Zielvorstellungen didaktischer und organisatorischer Natur ein, z.B. die gleichmäßige Verteilung der Unterrichtsstunden eines Faches in einer Klasse über die Woche, die Begrenzung der täglichen Unterrichtsbelastung der Lehrer sowohl nach unten wie nach oben und die Minimierung der Springstundenzahl für Lehrer. Als Basis für die Startpopulation wird der an die Verhältnisse des zu planenden Schuljahres angepasste Stundenplan des Vorjahres verwandt. Der Crossover-Operator sortiert zunächst die Matrixzeilen der beiden zu kreuzenden Chromosomen absteigend nach einer zeilenweise definierten lokalen Fitness des ersten Chromosoms. Die Kreuzung erfolgt dann hinter der k_1-ten Zeile, d.h. es werden die ersten k_1 Zeilen dem einen, die restlichen dem anderen Elternteil entnommen. Da durch diese Operation Unzulässigkeiten entstehen können, wird eine Reparationsroutine definiert, die durch paarweisen Tausch von Allelen innerhalb einzelner Zeilen Restriktionsverletzungen beseitigt. Der Mutationsoperator sieht innerhalb des betrachteten Chromosoms einen Tausch zwischen zwei zusammenhängenden, disjunkten Folgen der Länge *k* von Allelen innerhalb einer Zeile vor (mutation of order *k*), als Spezialfall das komplette Vertauschen zweier verschiedener Tage innerhalb einer Zeile (day mutation). Die Reproduktion erfolgt nach der oben beschriebenen Standardvorgehensweise.

Mit Hilfe ihres Verfahrens konnten Colorni, Dorigo und Maniezzo [1998] für mehrere Praxisfälle in jeweils achtstündigen Läufen auf einem 80486-33 MHz-PC zulässige Pläne erzeugen, die eine höhere Fitness als ein manuell bzw. mit Hilfe des kommerziellen Programms PC-Untis (vgl. Abschnitt 3.1.1.2) gesetzter Plan aufwiesen. Das Ergebnis konnte dabei durch die Einbeziehung eines Hillclimbing-Ansatzes, der durch Tausch-Operationen zwischen den Allelen der Gene zweier Stunden oder zweier Tage die Fitness eines Chromosomes zu erhöhen versucht, deutlich verbessert werden. Allerdings wurde der Genetische Algorithmus in einem Benchmark mit dem in Abschnitt 3.2.4.2 vorgestellten Tabu Search-Verfahren derselben Autoren hinsichtlich der erzielten Fitness deutlich geschlagen.

Ein weiterer Genetischer Algorithmus für die Setzung von Schulstundenplänen stammt von Drexl und Salewski [1997]. Er basiert auf einer indirekten Lösungsrepräsentation, d.h. die Gene eines Chromosoms enthalten nicht die konkreten Zeit- und ggf. Raumzuordnungen der Unterrichtseinheiten, sondern eine innerhalb einer definierten Vorschrift eindeutige

[41] Leider machen die Autoren in ihren Beiträgen von 1991 und 1992 nur recht spärliche Angaben über die von ihnen getesteten Beispielfälle und die für den Algorithmus verwendeten Parametereinstellungen. Der Aufsatz von 1998 enthält hierzu wesentlich detailliertere Informationen. Da in allen Fällen die getesteten Beispiele aus der Gegend von Mailand stammen, kann wohl davon ausgegangen werden, dass jeweils dieselben Rahmenbedingungen galten, so dass von der 1998er Darstellung auf die früheren Ansätze geschlossen werden kann.

Information darüber, wie bzw. wann eine Unterrichtseinheit zu setzen ist. Im Ansatz von Drexl und Salewski enthält jedes Gen eine zufällig erzeugte Zahl, die, eingesetzt in eine vordefinierte Prioritätsregel, festlegt, welche Sitzung welcher Unterrichtseinheit als Nächstes erzeugt wird. Für diese Sitzung wird die jeweils frühestmögliche Periode gewählt. Die Zufallszahlen werden durch Crossover rekombiniert oder durch Mutation, definiert als neues Ziehen der jeweiligen Zahl, verändert. Durch die indirekte Repräsentation werden direkt erkennbare, aufwendig zu reparierende Unzulässigkeiten zunächst verhindert. Es kann jedoch nicht ausgeschlossen werden, dass bei der jedem Generationswechsel nachfolgenden Dekodierung eines Chromosoms, also der Umsetzung der gespeicherten Informationen in einen konkreten Stundenplan, Unzulässigkeiten entstehen, die auf der Unfähigkeit der verwendeten Prioritätsregel beruhen, Zuordnungsengpässe frühzeitig zu erkennen und zu umgehen.

Einbezogen werden in das Modell von Drexl und Salewski die Restriktionstypen Vollständigkeit, Kollisionsfreiheit, Einhaltung von Ressourcenkapazitäten und die Beachtung von Mindest-Zeitabständen zwischen den Sitzungen bestimmter Unterrichtseinheitspaare. Eine Besonderheit besteht in der sonst unüblichen endogenen Festlegung der Anzahl und jeweiligen Dauer der zu jeder Unterrichtseinheit gehörenden Sitzungen über einen Unterrichtsmodus (mode) unter Beachtung der jeweils exogen vorgegebenen Gesamtstundenzahl. Die Fitnessfunktion repräsentiert die „Kosten“, die mit der Wahl eines Modus‘ für jede Unterrichtseinheit und mit ihrer konkreten Periodenzuordnung verbunden werden. Über die Einführung eines Dummy-Modus‘ mit hohen Kosten für die Nicht-Zuordnung von Unterrichtseinheiten wird die Vollständigkeitsanforderung relaxiert. Leider lässt sich den berichteten Ergebnissen nicht entnehmen, inwieweit es für die betrachteten Testfälle gelungen ist, zulässige Stundenpläne zu setzen. Aussagen zur Planqualität werden lediglich in Relation zur besten den Autoren für den jeweiligen Fall bekannten Lösung gemacht, die jedoch, zumindest für die größeren Fälle, ebenfalls heuristisch erzeugt wurde, sei es durch den Genetischen Algorithmus selbst oder durch die in Abschnitt 3.2.3.4 vorgestellte stochastische konstruktive Heuristik derselben Autoren. Auch wurden von Drexl und Salewski keine Praxisfälle, sondern lediglich zufallsgenerierte Beispiele getestet, so dass eine Aussage über die Praxistauglichkeit des Ansatzes kaum möglich ist.

Die Kombination eines Genetischen Algorithmus‘ mit einer konstruktiven Setzheuristik auf Basis der Logischen Programmierung (vgl. Abschnitt 3.2.3.3) schlägt Monfroglio [1996] für die Schulproblematik vor. Hierzu wird eine indirekte Repräsentation verwendet, in der die Zahl der Gene eines Chromosoms genau der Anzahl der zu setzenden Unterrichtseinheiten entspricht. Die Allele des Gens i eines Chromosoms gibt dabei an, welche Unterrichtseinheit in der i-ten Iteration der konstruktiven Setzheuristik in den Plan aufzunehmen ist. Vor Beginn des Genetischen Algorithmus‘ werden die Unterrichtseinheiten anhand eines speziellen Engpassindex‘ nach absteigender Schwierigkeit sortiert, wobei durch Permutation von Unterrichtseinheiten mit gleichem Indexwert mehrere verschiedene Reihenfolgen und damit unterschiedliche Chromosomen entstehen. Diese bilden die Startpopulation. Die nachfolgenden iterativen Crossover- und Mutationsoperationen werden so durchgeführt, dass stets jede zu setzende Unterrichtseinheit genau einmal in jedem Chromosom vertreten ist. Alternativ zu dieser Vorgehensweise definiert Monfroglio auch ein Verfahren der direkten Repräsentation, das er als zweite Lösungsphase an das Verfahren der indirekten Repräsentation anschließt. Leider macht der Autor keine genauen Angaben über die beiden betrachteten Testfälle, doch scheint es sich um künstlich erzeugte Probleme zu handeln, die mit 50 bzw. 100 Klassen recht groß sind. Monfroglio berichtet, dass unter Einsatz des hybriden Genetischen Algorithmus' eine bessere Lösungsqualität erzielt werden konnte als allein mit einer engpassorientierten konstruktiven Heuristik auf Basis der Logischen Programmierung.

Junginger [1995] präsentiert einen Genetischen Algorithmus für die Stundenplansetzung deutscher Oberstufen, ein Problem, das wegen des in diesen Jahrgangsstufen üblichen Kursprinzips eher mit der Veranstaltungssetzung an Universitäten verwandt ist als mit dem „klassischen" Setzungsproblem an Schulen (vgl. Abschnitt 2.5.1). Verwendet wird hier eine direkte Repräsentation, in der jedes Chromosom einer binären Zuordnungsmatrix entspricht. Darin bezeichnen die Zeilen die einzelnen Perioden, die Spalten die zu setzenden Kurse. Jedes Gen kann als Wert entweder 0 oder 1 annehmen. Hat das Gen in Zeile p und Spalte c eines Chromosoms den Wert 1, so bedeutet dies, dass Kurs c in Periode p stattfindet, der Wert 0 hingegen, dass er nicht in p stattfindet. Als Modellrestriktionen werden lediglich Vollständigkeit und Kollisionsfreiheit für die Schüler einbezogen, wobei die Vollständigkeit über alle genetischen Operationen hinweg gewahrt wird. Lehrer und Raumdimension bleiben vollständig ausgeklammert. Die Startpopulation wird unter Beachtung der Vollständigkeit zufällig erzeugt. Die Kreuzung zweier Chromosomen wird stets hinter einer zufällig gewählten Spalte vorgenommen, die Mutation durch Vertauschen der Allelen zweier zufällig bestimmter Gene, von denen eine 0 und die andere 1 ist, innerhalb einer zufällig ausgewählten Spalte. Die Auswahl der für Crossover und Mutation bestimmten Chromosomen erfolgt ebenfalls zufällig, wobei verschiedene Auswahlkonzepte getestet werden. Im Rahmen der Reproduktion wird die neue Generation nicht standardmäßig per Zufallsauswahl bestimmt, sondern es werden durch die r in der aktuellen Verfahrensiteration erzeugten Nachkommen die r unfittesten Populationsmitglieder ersetzt bzw., in einer zweiten Version des Verfahrens, alle N Chromosomen der vorigen Generation mit den r Nachkommen vermengt und aus diesen $N + r$ Chromosomen die N fittesten für die nächste Generation ausgewählt. Die zu minimierende Fitnessfunktion misst dabei die Summe der Kurskollisionen aller Schüler. Mit seinem Ansatz gelang es Junginger, zwei mit bis zu 22 Kursen, 60 Schülern und 19 Perioden eher kleine Beispielfälle in unter 2.000 Verfahrensiterationen zulässig zu lösen, allerdings wird die hierfür benötigte Zeit in seinem Bericht nicht genannt.

Weitreichende Anwendung haben Genetische Algorithmen auch in der Veranstaltungssetzung an Universitäten gefunden. Zu nennen sind hier die Arbeiten von Paechter et al. [1994], Paechter, Cumming und Luchian [1995], Paechter, Norman und Luchian [1996] sowie Paechter, Rankin und Cumming [1998], in denen in verschiedenen Varianten eine Hybridisierung mit lokalen Suchverfahren eingesetzt wurde. Rich [1996] verwendet ein Hillclimbing-Verfahren für die Reparation von Unzulässigkeiten, die durch genetische Operationen entstehen. Einen Ansatz, in dem mit Hilfe der Logischen Programmierung die

Zulässigkeit jeder erzeugten Lösung hinsichtlich Vollständigkeit, Kollisionsfreiheit, Raumbeschränkungen, Sperrungen und Fixierungen sichergestellt wird, präsentieren Erben [1995] sowie Erben und Keppler [1996]. Ansätze, die sowohl für die Veranstaltungssetzung an Universitäten wie die Examenssetzung entwickelt wurden, stammen von Burke, Elliman und Weare [1994] sowie Corne, Ross und Fang [1994].

Genetische Ansätze im Bereich der Examenssetzung haben Corne, Fang und Mellish [1993], Ross, Corne und Fang [1994], Ross und Corne [1995], Corne und Ross [1996], Ross, Hart und Corne [1998], Weare, Burke und Elliman [1995], Burke, Newall und Weare [1996], und Ergül [1996] vorgestellt. Eine Evaluation verschiedener genetischer Operatoren auf Basis einer direkten Repräsentation und einer Generierung ausschließlich zulässiger Lösungen geben Burke, Elliman und Weare [1995]. Dabei werden einige neue vorausschauende Operatoren definiert, die eine Verkürzung des Prüfungszeitraumes begünstigen. Auch aus diesem Bericht wird der Nutzen einer Hybridisierung mit einem Hillclimbing-Verfahren deutlich. Paechter [1994] hat ein Verfahren für die Setzung von Studierenden-Präsentationen entwickelt, die in ihrer Problemstruktur der Examenssetzung sehr ähnlich ist. Die Qualität der mit diesem Verfahren erzeugten Pläne lag über der einer Handsetzung. Angaben zur Laufzeit werden jedoch nicht gemacht.

3.2.4.5 Neuronale Netze

Wenige Ansätze der automatisierten Setzung beruhen auf einem Neuronalen Netzwerkmodell. Grundlage der Modellierung ist hier wie bei den Genetischen Algorithmen eine naturwissenschaftliche Analogie, die sich jedoch nicht auf die Genetik, sondern auf Struktur und Funktionsweise des Gehirns und des Nervensystems bezieht.[42]

Kernelemente eines Neuronalen Netzes sind Neuronen oder Nervenzellen, die über Nervenbahnen (Dendriten und Synapsen) netzwerkartig verknüpft sind. Sie stellen die Entscheidungsvariablen des Modells dar. Jedes Neuron verarbeitet Informationen über die Aktivitäten anderer Neuronen, die es über Nervenbahnen empfängt. Die Verarbeitung geschieht autonom, indem das Neuron die Informationen in eine Aktivierungsfunktion (activation function) einsetzt, dessen Ergebnis seine eigene Aktivität ist, die dem Wert der Entscheidungsvariablen entspricht. Dieser Wert wird als Information nach außen abgegeben und beeinflusst so wiederum die Festlegung der Aktivitäten anderer Neuronen. Die Berechnung der Aktivierungsfunktion erfolgt entweder direkt auf Basis einer Gewichtung der empfangenen Variablenwerte oder indirekt über den Einfluss der eigenen Aktivität auf den Energiewert des Netzwerkes, der das Resultat einer von den Aktivitäten aller Neuronen abhängenden Energiefunktion (energy function; = Bewertungsfunktion) ist. Die Energiefunktion beinhaltet die Relaxation sämtlicher Restriktionen sowie evtl. ein oder mehrere originäre Zielkriterien.

Eine grobe Charakterisierung Neuronaler Netzwerke innerhalb der iterativen Setzheuristiken gibt Tabelle 3.13. Sie unterstellt, dass die Aktivitätsberechnung der Neuronen sequenziell erfolgt. Es ist jedoch auch möglich, diese, analog zur Arbeitsweise des Gehirns, parallel auf verschiedenen Prozessoren durchzuführen. In beiden Fällen impliziert die autonome Informationsverarbeitung der einzelnen Neuronen das hohe Risiko einer Unzulässigkeit der Gesamtlösung, da die Konsequenzen der Aktivitätsänderung eines Neurons für die Aktivitäten benachbarter Neuronen erst in einer späteren Verfahrensiteration berücksichtigt werden. Das Verfahren führt daher nur dann zum Erfolg, wenn das Netzwerk gegen einen stabilen Zustand konvergiert, in dem die Aktivitäten aller Neuronen nach jeder Verfahrensiteration gleich bleiben und alle durch frühere Aktivitätsänderungen gesetzten Impulse von allen Neuronen verarbeitet sind.

Tabelle 3.13: Charakterisierung des iterativen Verfahrenstyps *Neuronales Netzwerk*

(zur Erläuterung der Entwurfsentscheidungen vgl. Abschnitt 3.2.1.2.2)

Entwurfsentscheidung	***Ausprägung***
|*OK*|	= 1
Nachbarschaftsdefinition	Modifikation der Aktivität eines einzelnen, zufällig ausgewählten Neurons in direkter Abhängigkeit von den Aktivitäten benachbarter Neuronen oder indirekt über die Energie-(= Bewertungs-)funktion
AnzGN	= 1
Bewertungsfunktion	offen; i.d.R. Maß für den Abstand zur Zulässigkeit
Akzeptanzregel	immer
Löschungsregel	eindeutig, da |*OK*| = 1
Abbruchkriterium	stabiler Netzwerkzustand bzw. max. Iterationszahl ohne Verbesserung von f_{best} bzw. generelle max. Iterationszahl erreicht

Einen Ansatz für die Stundenplansetzung an Schulen mit Hilfe eines Neuronalen Netzwerkmodells haben Gislén, Peterson und Söderberg [1992] entwickelt. Mit ihrem Verfahren konnten sie eine Lösung für eine große schwedische Oberstufe innerhalb von einer Stunde auf einer APOLLO DN 10000 Workstation erzeugen, wobei jedoch nur die Vollständigkeit und die Kollisionsfreiheit als Modellrestriktionen berücksichtigt waren. Weitere Anforderungen wie die Springstundenfreiheit für Klassen, die gleichmäßige Verteilung der Sitzungen eines Fachs in einer Klasse über die Woche und die Berücksichtigung einer Mittagspause wurden in Form von Zielsetzungen modelliert, die jedoch offenbar nach Erreichen der Zulässigkeit nicht weiterverfolgt wurden. Auch Kovacic [1993] präsentiert einen Ansatz für die Plansetzung an Schulen. Dieser Ansatz berücksichtigt lediglich Vollständigkeits- und Kollisionsfreiheitsrestriktionen sowie Sperrungen. Dem Autor ist es gelungen, in 51 von 100 Testläufen mit jeweils unterschiedlicher Initialisierung der Neuronenaktivitäten für ein hinsichtlich seiner Herkunft nicht spezifiziertes Problem mit 37 Lehrern, 18 Klassen, 40 Perioden, 33 Räumen und 202 Unterrichtsein-

[42] Eine grundlegende Einführung in die Theorie der Neuronalen Netze findet sich bei Hertz, Krogh und Palmer [1991], eine Darstellung von Anwendungsmöglichkeiten in Zusammenhang mit kombinatorischen Entscheidungsproblemen bei Hopfield und Tank [1985] sowie Burke und Ignizio [1992].

heiten mit einer Gesamtdauer von 588 Stunden einen zulässigen Plan zu erzeugen. Hierfür wurde eine Laufzeit von ca. 100 Minuten auf einem AT-PC benötigt.

Mit der Veranstaltungssetzung an Universitäten setzen sich Pellerin und Hérault [1994] auseinander, verwenden einen neuronalen Ansatz jedoch nur für die Zulässigkeitsprüfung von Teilplänen innerhalb eines interaktiven Systems, in dem alle Sitzungen manuell erzeugt werden. Dabei werden ausschließlich Kollisionsfreiheitsrestriktionen einbezogen. Ein weiterer Ansatz aus dem Universitätsbereich stammt von Elmohamed, Coddington und Fox [1998], die neben Vollständigkeit und Kollisionsfreiheit für Dozenten und Räume zahlreiche weitere Anforderungen, u.a. Sperrungen, Wegezeiten und minimale Kollisionen für Studierende, in die Setzung einbeziehen. Für die drei von ihnen betrachteten, mit bis zu 13.653 Studierenden, 1.200 Professoren, 509 Räumen und 3.839 Sitzungen sehr umfangreichen Beispielfälle ist es den Autoren jedoch nicht gelungen, einen vollständigen Plan zu erzeugen. In den Testläufen wurden trotz Laufzeiten von bis zu 20 Stunden auf einer Unix Workstation nur zwischen 61% und 95% der geforderten Sitzungen generiert.

Für die Setzung mündlicher Examina entwickelt wurde das von Mausser, Magazine und Moore [1996] sowie Mausser und Magazine [1996] beschriebene Verfahren, welches mehrperiodige Interviewdauern, Kollisionsfreiheit und Sperrungen berücksichtigt. Aufgrund der durch den Anwendungsfall vorgegebenen besonderen Problemstruktur – es müssen nicht alle Examina verplant werden, sondern die Gesamtlänge bzw. die Gesamtanzahl aller gesetzten Examina ist zu maximieren – spielen Unzulässigkeiten in dieser Anwendung eine untergeordnete Rolle. Die Lösungsqualität hängt jedoch nach Aussage der Autoren stark von der adäquaten Einstellung der Verfahrensparameter ab und erfordert einen aufwendigen Versuch-und-Irrtum-Prozess zu deren Identifikation.

3.2.4.6 Beurteilung

Obwohl keiner der bislang für die Lösung von Setzungsproblemen dokumentierten Ansätze alle für die Stundenplansetzung an deutschen allgemeinbildenden Schulen relevanten Restriktionen und Zielsetzungen berücksichtigt, gibt es keinen Grund zu bezweifeln, dass ein in dieser Hinsicht vollständiges Modell für die Lösung mit Hilfe einer beliebigen iterativen Heuristik formuliert werden kann (vgl. Tabelle 3.14). Fraglich ist allein, inwieweit die verfügbaren Verfahrenstypen genutzt werden können, um in einer akzeptablen Zeitspanne zulässige oder sogar hinsichtlich der Zielsetzungen gute Lösungen für ein solches Modell zu produzieren.

Alle bislang dargelegten iterativen Heuristiken für die Setzung von Schulstundenplänen beruhen auf der Relaxation mehrerer oder sogar aller problemrelevanter Restriktionen. Die Relaxation ist einerseits erforderlich, um schnell zu einer modellmäßig zulässigen Startlösung zu kommen, andererseits, um eine schnelle Durchführung der Nachbarschaftsoperationen zu garantieren, durch die die Suche vorangetrieben wird. Das dadurch entstehende hohe Risiko, sich am Ende eines langen Suchprozesses mit verbliebenen Unzulässigkeiten in der besten erzeugten Lösung abfinden zu müssen, ist naheliegend und wird durch die Vielzahl der Experimente, in denen dieser Fall eintrat, bestätigt.

Tabelle 3.14: Beurteilung *heuristischer iterativer Lösungsansätze*
(zur Erläuterung der Kriterien vgl. Tabelle 2.11, S.42)

Verfahrenstyp / *Kriterium*	Hillclimbing: bisherige Ansätze	Hillclimbing: Potenzialeinschätzung	Tabu Search: bisherige Ansätze	Tabu Search: Potenzialeinschätzung	Simulated Annealing: bisherige Ansätze	Simulated Annealing: Potenzialeinschätzung	Genetische Algorithmen: bisherige Ansätze	Genetische Algorithmen: Potenzialeinschätzung	Neuronale Netze: bisherige Ansätze	Neuronale Netze: Potenzialeinschätzung
Berücksichtigung der Restriktionstypen										
• Vollständigkeit (R-01):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.
• Kollisionsfreiheit (R-02):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.
• Sperrungen (R-03):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.
• Springstd.-verbot f. Klassen (R-04):	ja	mögl.	ja	mögl.	ja	mögl.	indir.	mögl.	ja[2]	mögl.
• Doppelunterrichtsverbot: (R-05):	ja	mögl.	ja	mögl.	ja	mögl.	ja	mögl.	indir.	mögl.
• Randstunden (R-06):	ja	mögl.	nein	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
• Fixierungen (R-07):	ja	mögl.	ja	mögl.	nein	mögl.	ja	mögl.	nein	mögl.
• Wegezeiten (R-09):	nein	mögl.	ja	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
• Freie Tage für Lehrer (R-12):	nein	mögl.	ja	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
Wahrscheinlichkeit einer zulässigen Lösung bzgl. aller o.g. Restriktionstypen:		gering		gering - mittel		gering - mittel		gering - mittel		gering
Berücksichtigung der Zielsetzungen										
• Gleichlange Klassentage (Z-01):	nein	mögl.	indir.	mögl.	indir.	mögl.	indir.	mögl.	nein	mögl.
• Früher Unterrichtsbeginn (Z-02):	nein	mögl.	indir.	mögl.	indir.	mögl.	indir.	mögl.	nein	mögl.
• Springstundenmin. f. Lehrer (Z-06):	nein	mögl.	ja	mögl.	ja	mögl.	ja	mögl.	nein	mögl.
• Vertretungsbereitschaft (Z-10):	nein	mögl.	nein	mögl.	nein	mögl.	nein	mögl.	nein	mögl.
• Fächerbeziehungen (Z-12):	nein	mögl.	ja	mögl.	nein	mögl.	ja	mögl.	nein	mögl.
Handhabung der Mehrfachzielsetzung:	Ziel-gew.[1]	Ziel-gew.	Ziel-gew.	Ziel-gew.	Ziel-gew.	Ziel-gew.	Ziel-gew.	Ziel-gew.	Ziel-gew.[1]	Ziel-gew.
Laufzeit:	kurz	kurz	mittel	mittel – lang	lang	lang	lang	lang	mittel	mittel - lang

Abkürzungen: mögl. = möglich, schw. = schwierig, indir. = indirekt, f. Doz. = für Dozenten, Zielgew. = Zielgewichtung
[1] unter Einbeziehung anderer als der hier genannten Zielsetzungen
[2] nur als Zielsetzung bei Gislén, Peterson und Söderberg [1992]; nach Erreichen der Zulässigkeit nicht weiterverfolgt

Allerdings sind die Erfolgsaussichten keineswegs für alle Verfahrenstypen gleich schlecht. Die größte Gefahr verbliebener Unzulässigkeiten entsteht bei isolierter Verwendung einer Hillclimbing-Heuristik, die vor allem deshalb schnell zu einem Ergebnis gelangt, weil sie abbricht, sobald sie durch Nachbarschaftsoperationen keine Verbesserung der einen betrachteten Lösung mehr erzielen kann. Besser steht es um solche iterativen Verfahrenstypen, die mit unterschiedlichen Mechanismen die Analyse weiter Bereiche des Entscheidungsbaumes sicherstellen, auch wenn dafür erheblich längere Laufzeiten von häufig mehreren Stunden in Kauf genommen werden müssen. Vor allem Tabu Search, Simulated Annealing und Genetische Algorithmen haben sich als erfolgreich erwiesen. Mit ihrer Hilfe gelang es mehrfach, Schulstundenpläne zu setzen, die von den jeweiligen Anwendern gegenüber manuell erzeugten Lösungen bevorzugt wurden. Allerdings waren für keinen der berichteten Fälle alle in Tabelle 3.14 genannten Restriktionen, sondern stets nur eine Auswahl relevant. Die Wahrscheinlichkeit, ein vollständiges Modell in akzeptabler Geschwindigkeit zulässig zu lösen, ist angesichts der bereits für weniger restringierte Modelle charakteristischen langen Laufzeiten eher als mäßig einzuschätzen. Schlecht dürften die Aussichten vor allem für ein Neuronales Netzwerk sein, denn die Wahrscheinlichkeit, dass dessen lokale Informationsverarbeitung gegen einen stabilen und dabei für das Problem zulässigen Gesamtzustand konvergiert, dürfte mit steigender Zahl einbezogener Restriktionen stark abnehmen. Allen iterativen Heuristiken für die Setzung von Schulstundenplänen ist das Erfordernis einer weitgehenden Problemrelaxation gemein, welches zwangsläufig eine Dominanz oder wenigstens Verwischung der originären Zielsetzungen durch die Relaxationsterme in der Bewertungsfunktion zur Folge hat.

Welcher der iterativen Verfahrenstypen für den Schulbereich am besten geeignet ist, lässt sich anhand bisher ausgeübter vergleichender Studien kaum feststellen. Die Tests, die Colorni, Dorigo und Maniezzo [1998] mit verschiedenen Genetischen Algorithmen, Tabu Search- und Simulated Annealing-Verfahren anhand mehrerer Testfälle aus dem Schulbereich durchgeführt haben, wiesen auf eine leistungsmäßige Dominanz von Genetischem Algorithmus und Tabu Search gegenüber dem Simulated Annealing hin. Ross und Corne [1995] hingegen stellen anhand mehrerer Examenssetzungsprobleme fest, dass mit Hilfe von stochastischem Hillclimbing und Simulated Annealing bessere Ergebnisse erzielt werden konnten als mit einem Genetischen Algorithmus. Diese Erfahrung wurde durch einen Vergleich derselben Verfahren anhand eines Problems der Personaleinsatzplanung für Priester durch Corne und Ogden [1998] bestätigt. Elmohamed, Coddington und Fox [1998] erzielten bei der Setzung universitärer Veranstaltungspläne mit Hilfe des Simulated Annealing deutlich bessere Ergebnisse als mit einem Neuronalen Netzwerk. Mausser und Magazine [1996] wiederum generieren mit einem Neuronalen Netzwerk Examenspläne von höherer Qualität als mit einem Hillclimbing-Ansatz und einem heuristischen (konstruktiven) Graphenfärbungsverfahren, stellen jedoch eine starke Parameterabhängigkeit in der Leistungsfähigkeit des Netzwerkes fest. Dowsland [1998] schließlich untersucht Tabu Search und Simulated Annealing anhand verschiedener Anwendungen in der Stundenplanung und beurteilt beide Verfahrenstypen positiv.

Letztlich dürfte der Schlüssel zum Erfolg jedoch weniger in der Beantwortung der Frage nach dem Sieger in der Konkurrenz, sondern vielmehr in einer geschickten Kombination verschiedener Verfahrenstypen liegen, wie die erfolgreiche wechselseitige Iteration von Tabu Search und Hillclimbing bei Schaerf [1996] zeigt, aber auch der Einsatz des Tabu Search in der Funktion des Mutationsoperators innerhalb eines Genetischen Algorithmus‘, den Costa [1995] für die Erstellung von Spielplänen für die nordamerikanische National Hockey League entworfen hat.

3.3 Zusammenfassung und kritische Würdigung

In den beiden vorangegangenen Unterkapiteln wurde ein Überblick über die bislang für Setzungsprobleme entwickelten Lösungsansätze gegeben und daraus das Potenzial abgeleitet, welches den jeweils zugrundeliegenden Verfahrenstypen hinsichtlich des erfolgreichen Einsatzes für die Stundenplansetzung an allgemeinbildenden Schulen zuzumessen ist. Das Ergebnis dieser Analyse ist in Tabelle 3.15 zusammengefasst.

Tabelle 3.15: Einschätzung von Verfahrenstypen hinsichtlich ihres Erfolgspotenzials für die Stundenplansetzung an allgemeinbildenden Schulen
(zur Erläuterung der Kriterien vgl. Tabelle 2.11, S.42)

		exakte konstruktive Verfahren					**heuristische iterative Verfahren**				
Verfahrenstyp / ***Kriterium***	**Handsetzung**	**exakte Graphenfärbung**	**(Gemischt) Ganzzahlige Programmierung**	**Nicht-lineare Programmierung**	**Implizite Enumeration mit Logischer Programmierung**	**Verfahren[1]**	**Hillclimbing heuristische konstruktive**	**Tabu Search**	**Simulated Annealing**	**Genetische Algorithmen**	**Neuronale Netze**
Berücksichtigung der Restriktionstypen											
• Vollständigkeit (R-01):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Kollisionsfreiheit (R-02):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Sperrungen (R-03):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Springstdverb. f. Klassen (R-04):	mögl.	schw.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Doppelunterrichtsverbot: (R-05):	mögl.	schw.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Randstunden (R-06):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Fixierungen (R-07):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Wegezeiten (R-09):	mögl.	schw.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Freie Tage für Lehrer (R-12):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
Wahrscheinlichkeit einer zulässigen Lösung bzgl. aller o.g. Restriktionstypen:	sehr hoch	gering	hoch	hoch	hoch	gering	gering	gering - mittel	gering - mittel	gering - mittel	gering
Berücksichtigung der Zielsetzungen											
• Gleichlange Klassentage (Z-01):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Früher Unterrichtsbeginn (Z-02):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Springstdmin. f. Lehrer (Z-06):	mögl.	schw.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Vertretungsbereitschaft (Z-10):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
• Fächerbeziehungen (Z-12):	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.	mögl.
Handhabung der Mehrfachzielsetzung:	real	indir.[2]	Zielgew.; ggf. Mindestniveaus d. entsprechende Restriktionen			Ziel-gew.	Zielgewichtung				
Laufzeit:	sehr lang	unakzeptabel	lang	unakzeptabel	lang	kurz - lang	kurz	mittel - lang	lang	lang	mittel - lang

Abkürzungen: mögl. = möglich, schw. = schwierig, Zielgew. = Zielgewichtung
[1] nur Verfahren, die in der Lage sind, den Lösungspfad zu wechseln (z.B. durch Tausch oder Backtracking)
[2] durch Variablen- und/oder Wertewahl

Obwohl seit ungefähr 40 Jahren an der Entwicklung automatisierter Setzverfahren gearbeitet wird, ist die Setzung in der schulischen Praxis bis heute weitgehend durch manuell gesteuerte Prozesse dominiert. Mit einem durchschnittlichen Arbeitsaufwand von etwa 40 Stunden stellt die Handsetzung dabei zwar das langwierigste, aber auch das sicherste Setzverfahren dar. Die Tatsache, dass ihre Durchführung keine explizite Modellbildung erfordert, erleichtert dem Stundenplaner die flexible und realitätskonforme Handhabung von Restriktionen und Zielsetzungen. Die intensive Nutzung seiner eigenen Intelligenz vermittelt ihm zudem einen weiteren Überblick über Planungszustand und eventuelle Engpässe, als dies einem Computerprogramm möglich ist. Durch den Erfolgsdruck führt die Suche des Stundenplaners letztlich immer zu einer Lösung, die vor dem Hintergrund der verfügbaren Ressourcen und der gestellten Anforderungen einen akzeptablen Kompromiss darstellt.

Lösungsansätze auf Basis automatisierter Setzverfahren lassen sich je nach verwendetem Verfahrenstyp grob in drei Kategorien einteilen: Ansätze auf Basis exakter konstruktiver Verfahren, Ansätze auf Basis heuristischer konstruktiver Verfahren und Ansätze auf Basis heuristischer iterativer Verfahren. Jeder Ansatz beruht auf einem in graphischer, logischer, mathematischer oder naturwissenschaftlicher Ausdrucksweise formulierten Modell, welches die jeweilige Problemstellung so abbildet, dass ein dem Verfahrenstyp entsprechender Algorithmus für ihre Lösung definiert werden kann.

Exakte konstruktive Verfahren weisen den großen Vorteil auf, dass sie garantiert gegen eine zulässige und optimale Lösung für das zugrundegelegte Modell konvergieren, sofern eine solche Lösung existiert. Dennoch ist ihre Verbreitung bis heute relativ begrenzt, da das Risiko, dass ein solches Verfahren nicht in akzeptabler Laufzeit zu einer Lösung gelangt, häufig als sehr hoch eingeschätzt wird. Bei differenzierterer Betrachtung einzelner Verfahrenstypen innerhalb dieser Kategorie kann dieses Pauschalurteil jedoch nicht aufrechterhalten werden. Skeptisch zu beurteilen sind vor allem Verfahren auf Basis graphischer Modelle, da diese keine hinreichend genaue Abbildung der Restriktionen und Zielsetzungen zulassen. Unpraktikabel sind wegen des Fehlens effizienter Algorithmen auch nicht-lineare mathematische Modellierungen. Hingegen eröffnen die (Gemischt-) Ganzzahlige und die Logische Programmierung durchaus die Möglichkeit, in einer zwar unter Umständen mehrstündigen, aber durchaus vertretbaren Zeitspanne zu einer Lösung zu gelangen, wie einige erfolgreiche Implementationen der jüngeren Zeit zeigen.

Heuristische konstruktive Verfahren schränken durch Anwendung frei definierbarer Konstruktionsregeln die Suche nach einer Modelllösung gegenüber den exakten Verfahren stark ein. Sie sichern so einen schnellen Verfahrensabschluss, nehmen dafür jedoch in Kauf, dass das Aufspüren einer existierenden zulässigen oder optimalen Lösung nicht garantiert werden kann. Trotz Anwendung hochgradig problemspezifischer Regeln für die engpassorientierte, zielgerichtete oder zufällige Auswahl der in jedem Konstruktionsschritt zu wählenden Variablen und des ihr zuzuweisenden Wertes ist es bislang nicht gelungen, mit Hilfe einer konstruktiven Setzheuristik für eine reale Schule einen vollständigen, zulässigen Stundenplan zu erzeugen. Da sich die für die verwandten Probleme der Veranstaltungssetzung an Universitäten und der Examenssetzung entwickelten Heuristiken nicht grundsätzlich von denen des Schulbereichs unterscheiden, ist mit einer erfolgreichen Adaption dieser Heuristiken für das Setzungsproblem allgemeinbildender Schulen nicht zu rechnen.

Iterative Lösungsverfahren unterscheiden sich von konstruktiven grundsätzlich dadurch, dass sie Lösungen nicht wie letztere aus den Basisdaten heraus konstruieren, sondern durch iterative Modifikation mit Hilfe sog. Nachbarschaftsoperatoren aus anderen, bereits generierten Lösungen ableiten. Sie setzen somit das Vorliegen mindestens einer mit einem konstruktiven Verfahren erzeugten Lösung voraus. Da iterative Verfahren, mit Ausnahme der unpraktikablen vollständigen Enumeration, das Auffinden einer existierenden hinsichtlich des Modells zulässigen oder optimalen Lösung nicht garantieren können, sind sie stets heuristischer Natur. Damit eine Startlösung leicht gewonnen werden kann und eine schnelle Bewegung des Verfahrens im Lösungsraum möglich ist, beruhen alle Modelle, die einer iterativen Heuristik für die Setzung von Schulstundenplänen zugrunde liegen, auf einer weitreichenden Relaxation der Problemrestriktionen. Ziel des Verfahrens ist es i.d.R., mit Hilfe zielgerichteter oder zufälliger Nachbarschaftsoperationen die gewichtete Summe der Unzulässigkeiten in der aktuell besten gefundenen Lösung schrittweise zu verringern, um so schließlich zu einer zulässigen Lösung zu gelangen. Die Einbindung originärer Zielsetzungen in die gewichtete Bewertungsfunktion ist ebenfalls möglich, tritt jedoch gegenüber der Relaxationsproblematik weitgehend in den Hintergrund.

Unter den verfügbaren iterativen Verfahrenstypen verspricht das Hillclimbing die geringsten Erfolgsaussichten, da es abbricht, sobald es die betrachtete Bewertungsfunktion mittels der vorgesehenen Operationen nicht mehr verbessern kann. Ebenfalls gering erscheint die Wahrscheinlichkeit, mit Hilfe eines Neuronalen Netzes zu einer zulässigen Lösung zu gelangen, denn es ist fraglich, ob die für diesen Verfahrenstyp charakteristische lokale Informationsverarbeitung angesichts der starken Restringiertheit des Problems und der dadurch bedingten hohen gegenseitigen Beeinflussung der Neuronen in vertretbarer Zeit gegen einen stabilen Zustand konvergieren kann. Gute Ergebnisse für Setzungsprobleme wurden mit Tabu Search, Simulated Annealing und Genetischen Algorithmen erzielt. Sie übertrafen teilweise diejenigen einer Handsetzung. Allerdings hat auch im Bereich dieser Verfahrenstypen keiner der berichteten Ansätze alle für die Stundenplansetzung an allgemeinbildenden Schulen relevanten Restriktionen und Zielsetzungen berücksichtigt. Es muss angesichts der bereits für weniger restringierte Modelle langen Laufzeiten von mehreren Stunden in Frage gestellt werden, ob ein Verfahren auf Basis eines dieser Typen in die Lage versetzt werden kann, zulässige Lösungen für ein vollständiges Setzungsmodell zu produzieren, ohne dabei die Lösungszeiten über das akzeptable Maß hinaus zu verlängern.

Insgesamt ist festzustellen, dass es bisher keinen automatisierten Ansatz gibt, der eine Gewähr dafür bietet, das in Kapitel 2 beschriebene Problem der Stundenplansetzung an allgemeinbildenden Schulen in angemessener Zeit zulässig und hinsichtlich der gesetzten Ziele akzeptabel zu lösen. Andererseits ist ebenso zu konstatieren, dass die Möglichkeiten, die die verfügbaren Verfahrenstypen für die Gestaltung von Lösungsansätzen bieten, noch bei weitem nicht voll ausgeschöpft sind. So setzen sich erst wenige Autoren mit möglichen

Hybridisierungen heuristischer Verfahren auseinander, etwa dem Einsatz von Hillclimbing-Verfahren für die Mutation innerhalb Genetischer Algorithmen oder der heuristischen Verwendung der Logischen Programmierung für die Generierung von Startlösungen iterativer Heuristiken. Aber auch und gerade die exakten Verfahrenstypen erwachen erst langsam aus ihrem „Dornröschenschlaf". Ihre Vorteile sind, abgesehen von der bereits erwähnten Konvergenzgarantie,

1. eine hohe Abbildungsgenauigkeit von Restriktionen und Zielsetzungen durch mathematische oder logische Modellierung,
2. der nicht oder im Vergleich zu iterativen Heuristiken in wesentlich geringerem Umfang gegebene Zwang zur Problemrelaxation und, daraus erwachsend,
3. die Möglichkeit, Zielsetzungen des Problems umfassend zu berücksichtigen, wobei neben der in allen Verfahren üblichen Zielgewichtung durch die Formulierung entsprechender Modellrestriktionen die Erreichung von Mindestniveaus der Ziele gesichert werden kann.

Die Chancen, diese Vorteile für die Gewinnung besserer Setzergebnisse nutzen zu können, sind heute nicht nur wegen der erheblich gestiegenen Hardware-Leistungsfähigkeit, sondern auch und gerade aufgrund der zwischenzeitlichen Entwicklung hoch-effizienter Standard-Optimierungssoftware wesentlich höher als etwa noch vor 20 Jahren. Handlungsbedarf ist trotz aller bisherigen Bemühungen in jedem Fall gegeben. Zwar sind PC-gestützte Stundenplanprogramme mit integrierter Setzautomatik heute an vielen Schulen im Einsatz, doch sind die damit generierten Pläne ohne mühsame manuelle Nachbearbeitung i.d.R. nicht verwendbar. Äußerungen Berliner Stundenplaner in der im Winter 1997/98 durchgeführten Umfrage deuten darauf hin, dass sich der hohe Arbeitsaufwand für die Setzung eines Stundenplanes durch die Verwendung eines PC-Programms nicht wesentlich verkürzt.

Vor diesem Hintergrund soll nun ein neuer Ansatz gewagt werden, die Leistungsfähigkeit der automatisierten Setzung zu verbessern, ein Ansatz auf Basis der Gemischt-ganzzahligen Programmierung.