



(10) **DE 10 2021 204 043 B3** 2022.02.03

(12)

Patentschrift

(21) Aktenzeichen: **10 2021 204 043.8**
(22) Anmeldetag: **22.04.2021**
(43) Offenlegungstag: –
(45) Veröffentlichungstag
der Patenterteilung: **03.02.2022**

(51) Int Cl.: **G06Q 10/06 (2012.01)**
G06Q 10/04 (2012.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
Robert Bosch Gesellschaft mit beschränkter Haftung, 70469 Stuttgart, DE

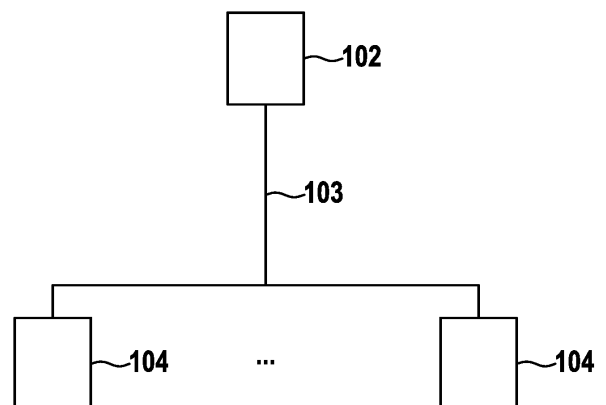
(72) Erfinder:
Skocovsky, Peter, Wien, AT; Geibinger, Tobias, Wien, AT; Stepanova, Daria, 71272 Renningen, DE; Eiter, Thomas, Wien, AT; Musliu, Nysret, Wien, AT; Oetsch, Johannes, Wien, AT

(56) Ermittelte Stand der Technik:

DE	10 2016 204 392	A1
DE	695 13 877	T2
US	9 798 947	B2
US	2020 / 0 320 456	A1

(54) Bezeichnung: **Scheduler und, insbesondere computerimplementiertes, Verfahren zur Maschinenzeitplanung zum Durchführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz**

(57) Zusammenfassung: Ein Scheduler (102) und ein Verfahren, insbesondere ein computerimplementiertes Verfahren, zur Maschinenzeitplanung zum Ausführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz (104), welches das Bestimmen, in einem ersten Durchlauf, eines ersten Zeitplans, der ein Optimierungsproblem löst, wobei das Optimierungsproblem durch eine Reihe von Regeln definiert wird, welche die Gruppe von Jobs zu dem Maschinensatz (104) zuweisen, wobei der erste Zeitplan jeden Job der Gruppe von Jobs einer Maschine des Maschinensatzes (104) zuordnet, die zum Verarbeiten dieses Jobs in der Lage ist, wobei eine Ausführung von Jobs, die der erste Zeitplan zu einer Maschine zuweist, durch diese Maschine zeitlich derart geplant wird, dass sie innerhalb einer Maschinenzeitspanne abgeschlossen wird, und, in einem zweiten Durchlauf, entweder das Festlegen einer Beschränkung für mindestens eine Maschinenzeitspanne und das Bestimmen eines zweiten Zeitplans, der das Optimierungsproblem für die Gruppe von Jobs und den Maschinensatz (104) unter der Beschränkung löst, oder das Bestimmen eines zweiten Zeitplans, der das Optimierungsproblem für eine Untergruppe der Gruppe von Jobs und eine Untergruppe des Maschinensatzes (104) löst, wobei die Untergruppe des Maschinensatzes (104) die Maschine ausschließt und die Untergruppe der Gruppe von Jobs Jobs ausschließt, die der Maschine gemäß dem ersten Zeitplan zugewiesen werden, umfasst.



Beschreibung

Allgemeiner Stand der Technik

[0001] Die Erfindung betrifft einen Scheduler und ein, insbesondere computerimplementiertes, Verfahren zur Maschinenzeitplanung zum Durchführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz.

[0002] Dimitrios Letsios, Miten Mistry, Ruth Misener „Exact lexicographic scheduling and approximate rescheduling“, European Journal of Operational Research Volume 290, Ausgabe 2, 16. April 2021, Seite 469-478 offenbart Aspekte eines verwandten Zeitplanungsansatzes.

[0003] Aus der US 2020 / 0 320 456 A1, der DE 695 13 877 T2, der DE 10 2016 204 392 A1 und der US 9 798 947 B2 sind weitere Verfahren zur Maschinenzeitplanung bekannt.

Offenbarung der Erfindung

[0004] Der Scheduler, das Computerprogramm und das, insbesondere computerimplementierte, Verfahren zur Maschinenzeitplanung gemäß den unabhängigen Ansprüchen lösen die Aufgabe, die Zeitplanung weiter zu verbessern und direkt für komplexere Zeitplanungsprobleme verwendbar zu sein, die Maschinenfestzuordnung, Abfolge-abhängige Einrichtezeiten und Freigabetermine beinhalten und die Spezifikation des Zeitplanungsproblems einfacher aufrechtzuerhalten ist, wenn sich Anforderungen ändern.

[0005] Das, insbesondere computerimplementierte, Verfahren zur Maschinenzeitplanung zum Ausführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz umfasst das Bestimmen, in einem ersten Durchlauf, eines ersten Zeitplans, der ein Optimierungsproblem löst, wobei das Optimierungsproblem durch eine Reihe von Regeln definiert wird, welche die Gruppe von Jobs zu dem Maschinensatz zuweisen, wobei der erste Zeitplan jeden Job der Gruppe von Jobs einer Maschine des Maschinensatzes zuordnet, die zum Verarbeiten dieses Jobs in der Lage ist, wobei eine Ausführung von Jobs, die der erste Zeitplan zu einer Maschine zuweist, durch diese Maschine derart zeitlich geplant wird, dass sie innerhalb einer Maschinenzeitspanne abgeschlossen wird, und, in einem zweiten Durchlauf, entweder das Festlegen einer Beschränkung für mindestens eine Maschinenzeitspanne und das Bestimmen eines zweiten Zeitplans, der das Optimierungsproblem für die Gruppe von Jobs und den Maschinensatz unter der Beschränkung löst, oder das Bestimmen eines zweiten Zeitplans, der das Optimierungsproblem für eine Untergruppe der Gruppe von Jobs und eine Untergruppe des Maschinensatzes löst, wobei die Untergruppe des Maschinensatzes die Maschine ausschließt und die Untergruppe der Gruppe von Jobs ausschließt, die der Maschine gemäß dem ersten Zeitplan zugewiesen werden. Dieses Verfahren kann direkt für komplexere Zeitplanungsprobleme verwendet werden, die Maschinenfestzuordnung, Abfolge-abhängige Einrichtezeiten und Freigabetermine beinhalten. Darüber hinaus ist ein weiterer Vorteil, dass die Spezifikation des Zeitplanungsproblems einfacher aufrechtzuerhalten ist, wenn sich Anforderungen ändern, da das Verfahren auf einen Lösungsansatz für Wissensrepräsentation und Wissensverarbeitung nach dem neuesten Stand der Technik mit einer entsprechenden Modellierungssprache anwendbar ist.

[0006] Das Verfahren kann das Auswählen einer Maschine aus dem Maschinensatz mit einer Maschinenzeitspanne, die bei einer Produktionsspanne zum Ausführen der Aufgabe endet, gemäß dem ersten Zeitplan und das Bilden der Untergruppe des Maschinensatzes ohne mindestens eine Maschine umfassen. Dies verringert die Komplexität in dem Optimierungsproblem in Bezug auf die Maschinen.

[0007] Das Verfahren kann das Bilden der Untergruppe der Gruppe von Jobs ohne mindestens einen Job, der zu der Maschine zugewiesen wird, umfassen. Dies verringert die Komplexität in dem Optimierungsproblem in Bezug auf die Jobs.

[0008] Das Verfahren kann das Festlegen mehrerer Maschinenzeitspannen für Maschinen des Maschinensatzes, die derart zeitlich geplant sind, dass mindestens ein Job gemäß dem ersten Zeitplan ausgeführt wird, umfassen, wobei eine Maschinenzeitspanne mit einer längeren Dauer als mindestens eine andere Maschinenzeitspanne der mehreren Maschinenzeitspannen als die Beschränkung aus den mehreren Maschinenzeitspannen ausgewählt wird. Dies beschränkt die Lösung des Optimierungsproblems durch die Maschinenzeitspanne, die eine längere Dauer als andere Maschinenzeitspannen aufweist. Somit wird die Lösung sehr effizient festgelegt.

[0009] Das Verfahren kann das Festlegen einer Abfolge, in welcher Jobs, die zu der Maschine zugewiesen werden, zeitlich für die Verarbeitung auf dieser Maschine geplant werden, und das Festlegen der Maschinenzeitspanne in Abhängigkeit von einem Fertigstellungszeitpunkt eines letzten Jobs in der Abfolge umfassen. Dies gestattet das Festlegen der Maschinenzeitspanne zum Erhöhen der Robustheit des Zeitplanungsverfahrens.

[0010] Das Verfahren kann das Festlegen des Fertigstellungszeitpunktes des letzten Jobs in Abhängigkeit von einer Summe einer Startzeit eines ersten Jobs in der Abfolge und einer Summe der Verarbeitungszeiten von mindestens einem Job in der Abfolge, der zeitlich derart geplant ist, dass er früher als der letzte Job startet, und einer Dauer des letzten Jobs umfassen. Somit wird der Fertigstellungszeitpunkt des letzten Jobs sehr präzise festgelegt. Dies erhöht die Robustheit des Zeitplanungsverfahrens weiter.

[0011] Das Verfahren kann das Bestimmen mehrerer Zeitpläne in mehreren Durchläufen umfassen, wodurch eine Anzahl von Durchläufen auf weniger als eine Anzahl von Maschinen in den mehreren Maschinen begrenzt wird. Dies macht die Zeitplanung viel schneller.

[0012] Das Verfahren kann die Zeitplanung von mindestens einer Maschine des Maschinensatzes zum Ausführen von mindestens einem Job der Gruppe von Jobs gemäß dem zweiten Zeitplan umfassen. Auf diese Weise werden die Maschinen zeitlich sehr effizient geplant.

[0013] Ein Scheduler zum Ausführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz innerhalb einer Produktionsspanne ist zum Ausführen der Schritte des Verfahrens konfiguriert.

[0014] Ein Computerprogramm umfasst computerlesbare Anweisungen, die, wenn sie durch einen Computer ausgeführt werden, den Computer zum Durchführen der Schritte in dem Verfahren veranlassen.

[0015] Weitere vorteilhafte Ausführungsformen können aus der folgenden Beschreibung und den Zeichnungen abgeleitet werden. Darin zeigt:

Fig. 1 schematisch mindestens einen Teil eines Schedulers für einen Maschinensatz,

Fig. 2 schematisch Schritte, die durch den Scheduler durchgeführt werden,

Fig. 3 schematisch Schritte in einem ersten Verfahren, und

Fig. 4 schematisch Schritte in einem zweiten Verfahren.

[0016] **Fig. 1** zeigt schematisch mindestens einen Teil eines Schedulers 102 für einen Maschinensatz K. Der Scheduler 102 kann mindestens einen Prozessor und mindestens einen Speicher umfassen, insbesondere einen Speicher zum Speichern von computerlesbaren Anweisungen, die, wenn sie durch mindestens einen Prozessor des Schedulers 102 ausgeführt werden, den Scheduler 102 zum Ausführen eines Verfahrens, insbesondere eines computerimplementierten Verfahrens, zur Zeitplanung, das unten beschrieben wird, veranlassen. Der Maschinensatz (im Folgenden mit K bezeichnet, nicht eingezeichnet) stellt mehrere Maschinen 104 dar. Der Scheduler 102 und die Maschinen der mehreren Maschinen 104 in dem Beispiel sind zumindest zeitweise über eine Datenverknüpfung 103 verbunden. Der Scheduler 102 in dem Beispiel ist zum Anweisen von mindestens einigen der Maschinen der mehreren Maschinen 104 zum Ausführen von Jobs einer Gruppe von Jobs (im Folgenden mit J bezeichnet, nicht eingezeichnet) konfiguriert.

Das Verfahren löst ein Zeitplanungsproblem:

[0017] In Anbetracht von m Maschinen k in dem Maschinensatz K und n Jobs j in der Gruppe von Jobs J muss jeder Job j durch eine einzelne Maschine k verarbeitet werden und jede Maschine k kann jeweils höchstens einen Job j verarbeiten; eine Bevorrechtigung ist nicht gestattet.

[0018] Einige Maschinen k können nur bestimmte Jobs j erledigen; $cap(j)$ ist ein Maschinensatz, der zum Verarbeiten von Job j in der Lage ist.

[0019] Ein Freigabetermin $r_{j;k}$ wird für jeden Job j und jede Maschine k als eine nichtnegative Ganzzahl spezifiziert. Freigabetermine sind maschinenabhängig, weil die Transportzeit für Jobs zu den Maschinen vom Transportsystem und ihrem Standort abhängt. Kein Job j kann vor seinem Freigabetermin $r_{j;k}$ starten.

[0020] Eine spezifizizierte Zeitmenge kann zum Wechseln von einem Job j zum nächsten erforderlich sein. Eine Zeit $s_{i;j;k}$ ist die Zeit, die zum Einrichten von Job j direkt nach Job i auf Maschine k benötigt wird.

[0021] Diese Zeiten werden als Abfolge-abhängige Einrichtezeiten bezeichnet. Jeder Job j weist eine Dauer $d_{j;k}$ auf, die von der Maschine k abhängt, zu welcher er zugewiesen wird.

[0022] Ein Zeitplan für eine Problem Instanz ist durch Folgendes definiert:

1. eine Zuweisung a , die jeden Job j einer Maschine $k \in \text{cap}(j)$, die zu dessen Verarbeitung in der Lage ist, zuordnet;
2. für jede Maschine k , ein Gesamtauftrag \preceq_k zur Gruppe J von Jobs j zugewiesen zu der Maschine k über die Zuweisung a .

[0023] Die Beziehung \preceq_k legt die Abfolge fest, in welcher die Jobs j in der Gruppe von Jobs J auf der Maschine k verarbeitet werden.

[0024] Falls jeder Job j durch irgendeine Maschine k verarbeitet werden kann, existiert mindestens ein Zeitplan für die Problem Instanz.

[0025] Es soll angenommen werden, dass j_1, \dots, j_l eine Verarbeitungsabfolge der Jobs j ist, die in einem gegebenen Zeitplan zu der Maschine k zugewiesen werden. Die Verarbeitungszeit p_j eines Jobs j_i ist seine Dauer plus die Einrichtezeit für seinen Vorgänger, falls einer vorhanden ist; d.h., $p_{j_1} = d_{j_1;k}$ und $p_{j_i} = s_{j_{i-1};j_i;k} + d_{j_i;k}$, bei $i > 1$.

[0026] Eine Startzeit st_{j_i} von Job j_i ist $r_{j_i;k}$, falls $i = 1$, und $\max(r_{j_{i-1};k}; st_{j_{i-1}} + p_{j_{i-1}})$ bei $i > 1$. Ein Fertigstellungszeitpunkt c_{j_i} von Job j_i ist $st_{j_i} + p_{j_i}$. Eine Maschinenzeitspanne der Maschine k , $\text{span}(k)$, ist der Fertigstellungszeitpunkt des letzten Jobs j_l auf der Maschine k .

[0027] Das Ziel ist das Maximieren eines Durchsatzes, welcher in dem Beispiel als eine Anzahl von Jobs j , die pro Zeiteinheit verarbeitet werden, definiert ist. Grundsätzlich wird dies durch das Minimieren einer Produktionsspanne, d.h. einer längsten Maschinenzeitspanne des Zeitplans, erreicht. Jedoch stellt sich häufig heraus, wenn Maschinen mit einer hohen Festzuordnung im Einsatz sind, dass viele Jobs nur durch ein paar wenige Maschinen verarbeitet werden können. Dies bedeutet, dass wenige Maschinen die Produktionsspanne festlegen, nahezu ungeachtet dessen, wie Jobs zeitlich auf den verbleibenden Maschinen geplant werden. Dies ist nicht ideal, wenn Jobs dynamisch umgeplant werden müssen, z.B. aufgrund eines plötzlichen Maschinenausfalls, was jederzeit passieren kann. Für derartige Fälle haben Fachexperten die Forderung zum Ausdruck gebracht, dass „alle Maschinen ihre Vorgänge so früh wie möglich abschließen sollten“. Dies ermöglicht dem Scheduler die Freiheit für Umordnungen und hilft beim Aufrechterhalten eines hohen Durchsatzes.

[0028] Zur Behandlung dieses Problems verwendet das Verfahren eine Zielfunktion, welche auf das Erfassen eines Wunsches abzielt, dass alle Maschinen ihre Vorgänge so früh wie möglich abschließen. Die Zielfunktion wird im Folgenden als eine lexikografische Produktionsspanne bezeichnet.

[0029] Die lexikografische Produktionsspanne eines Zeitplans ist als ein Tupel aller Maschinenzeitspannen in nicht-aufsteigender Reihenfolge definiert. Das Verfahren verwendet eine lexikografische Reihenfolge der Tupel unterschiedlicher Zeitpläne zum Vergleich und bevorzugt einen Zeitplan mit einer kürzeren lexikografischen Produktionsspanne gegenüber einem Zeitplan mit einer längeren. Ein resultierender Zeitplan ist der Zeitplan mit einer minimalen lexikografischen Produktionsspanne. Der resultierende Zeitplan weist daher auch eine minimale Produktionsspanne auf, jedoch werden Gleichstände unter Verwendung von Maschinen, die ihre Vorgänge früher abschließen, aufgelöst.

[0030] Zum Berechnen von Zeitplänen mit einer minimalen lexikografischen Produktionsspanne werden unten ein beispielhafter exakter Algorithmus und ein beispielhafter angenäherter Algorithmus beschrieben. Beide verwenden Antwortsatzprogrammierung (ASP - Answer-Set Programming), einen logikbasierten Ansatz zur deklarativen Problemlösung, der zum Beispiel in Folgenden beschrieben ist:

Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Communications of the ACM 54(12), 92-103 (2011). El-Kholany, M., Gebser, M.: Job shop scheduling with multi-shot asp. <http://www.kr.tuwien.ac.at/events/taasp20/accepted.html> (2020), Workshop on Trends and Applications of Answer Set Programming (TAASP 2020).

[0031] Eine Zielfunktion, welche als lexikografische Produktionsspanne bezeichnet wird, formalisiert die Forderung, dass alle Maschinen ihre Vorgänge so früh wie möglich abschließen sollten.

[0032] In Anbetracht eines Zeitplans S , der m Maschinen beinhaltet, ist die lexikografische Produktionsspanne von S ein Tupel $ms(S) = (c_1, \dots, c_m)$ sämtlicher Maschinenzeitspannen c_i von S in nicht-aufsteigender Reihenfolge.

[0033] In dieser Definition ist c_1 eine maximale Maschinenzeitspanne und entspricht daher der Produktionsspanne gemäß Zeitplan S . Bei den Zeitplänen S und S' , die jeweils m Maschinen beinhalten, weist S eine kürzere lexikografische Produktionsspanne als S' auf, wenn $ms(S)$ in der lexikografischen Reihenfolge kürzer als $ms(S')$ ist. Dies bedeutet, dass auf dem geringsten Index i $c_i < c'_i$, wobei $ms(S) = (c_1, \dots, c_m)$ und $ms(S') = (c'_1, \dots, c'_m)$ nicht übereinstimmen.

[0034] Bei einem Satz S von Zeitplänen ist $S \in S$ dann optimal, wenn $ms(S)$ über alle Zeitpläne in dem Satz S hinweg minimal ist.

[0035] Die Dynamik eines Zeitplans S wird als eine Anzahl von Maschinen $M(S; t)$ beschrieben, die ihre Vorgänge zu oder vor einem Zeitpunkt t abschließen.

[0036] Wenn S und S' zwei, insbesondere unterschiedliche, Zeitpläne für eine beliebige Problem Instanz sind, dann ist $ms(S) < ms(S')$, wenn der Zeitpunkt t derart ist, dass $M(S; t) > M(S'; t)$, und für jeden $t' > t$, $M(S; t') \geq M(S'; t')$.

[0037] Bei Problemen, die viele Maschinen beinhalten, kann ein hierarchisches Minimieren sämtlicher Maschinenzeitspannen unverhältnismäßig sein, wenn die Gesamtproduktionsspanne nur durch wenige Maschinen dominiert wird. Jedoch gestattet das Vergleichen lexikografischer Produktionsspannen eine Parametrisierung mit einer Ganzzahl l , die eine Anzahl von Komponenten definiert, die in dem Vergleich zu berücksichtigen sind.

[0038] Bei den Zeitplänen S und S' , die jeweils m Maschinen beinhalten, und mit der Ganzzahl l , $1 \leq l \leq m$, ist die Produktionsspanne gemäß Zeitplan S $ms(S) = (c_1, \dots, c_m)$ in einer parametrisierten lexikografischen Reihenfolge kürzer als die Produktionsspanne gemäß Zeitplan S' $ms(S') = (c'_1, \dots, c'_m)$. Bei einem Zeitplan mit m Maschinen wird die Produktionsspanne erhalten, wenn $l = 1$ und die volle lexikografische Produktionsspanne wird erhalten, wenn $l = m$.

[0039] In dem Beispiel wird die Problemstellung mit einem Modell modelliert, das Regeln umfasst, welche die Gruppe von Jobs J zu dem Maschinensatz K zuweisen. Ein Zeitplan ordnet jeden Job j der Gruppe von Jobs einer Maschine $k \in \text{Cap}(j)$ des Maschinensatzes K , die zum Verarbeiten dieses Jobs j in der Lage ist, zu. Eine Ausführung von Jobs j , die der Zeitplan zu einer Maschine k zuweist, durch diese Maschine k wird derart zeitlich geplant, dass sie innerhalb einer Maschinenzeitspanne c_i dieser Maschine k abgeschlossen wird.

[0040] Das Modell und die oben beschriebene Zielfunktion können mit einer Erweiterung von ASP mit Differenzlogik modelliert werden, wie zum Beispiel beschrieben in Janhunen, T., Kaminski, R., Ostrowski, M., Schellhorn, S., Wanko, P., Schaub, T.: Clingo goes linear constraints over reals and integers. Theory and Practice of Logic Programming 17(5-6), 872-888 (2017).

[0041] Zum Lösen eines Optimierungsproblems, welches durch das Modell und die Zielfunktion mit dem exakten Algorithmus definiert wird, kann Multi-Shot-Lösen zum Implementieren mehrerer Aufrufe an den ASP-Löser auf eine ressourcenfreundliche Art und Weise verwendet werden. Dies ist zum Beispiel beschrieben in Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Multi-shot asp solving with clingo. Theory and Practice of Logic Programming 19(1), 27-82 (2019).

[0042] Zum Lösen des Optimierungsproblems mit dem angenäherten Algorithmus können Teile einer Lösung nach Löser-Aufrufen vorliegen, was das Finden nahezu optimaler Lösungen in kurzer Zeit gestattet.

[0043] Der Scheduler und das Verfahren zur Zeitplanung können bei der Halbleiterherstellung Anwendung finden. In diesem Aspekt wird der Maschinensatz K zum Herstellen von Halbleitern verwendet. Der Scheduler und das Verfahren zur Zeitplanung können auch in verteilten Rechensystemen eingesetzt werden. Die Maschinen in diesem Aspekt können Prozessoren oder Rechenkerne von Prozessoren oder dedizierte Hardware zum Durchführen von Rechenoperationen sein. Die Jobs werden unter gewissen Beschränkungen zum

Berechnen machbarer Zeitpläne zu Maschinen zugewiesen. Eine der Anforderungen ist, dass die Zeitpläne derart erstellt werden sollten, dass alle Maschinen ihre Vorgänge so früh wie möglich abschließen. Die Optimierungsfunktion, lexikografische Produktionsspanne genannt, und die Algorithmen zu deren Minimierung, die unten beschrieben werden, können dazu verwendet werden, sich mit dieser Anforderung auseinanderzusetzen und die gewünschten Zeitpläne zu berechnen. Die Maschinen können gemäß dem Zeitplan angewiesen werden, der das Optimierungsproblem löst.

[0044] ASP ist ein kompakter relationaler, im Wesentlichen propositionaler, Formalismus, bei welchem in einem Vorverarbeitungsschritt, Fundierung genannt, Variable in der Eingabesprache durch konstante Symbole ersetzt werden. Ein ASP-Programm ist eine Reihe von Regeln der folgenden Form:

$$p_1 | \dots | p_k : -q_1, \dots, q_m, \text{ nicht } r_1, \dots, \text{ nicht } r_n.$$

wobei alle p_i , q_j und r_l Atome sind. Ein Kopf sind alle Atome vor dem Implikationssymbol $:-$ und ein Körper sind sämtliche Atome und negierten Atome danach. Die intuitive Bedeutung dieser Regel ist, dass, wenn alle Atome q_1, \dots, q_m abgeleitet werden können und es keine Belege für jegliche der Atome r_1, \dots, r_n gibt (d.h. die Regel greift), mindestens eines aus p_1, \dots, p_k wahr sein muss. Eine Interpretation I ist ein Satz von Atomen. Es handelt sich um einen Antwortsatz eines Programms, wenn alle seine Regeln auf minimale und konsistente Art und Weise erfüllt werden. I muss ein \subseteq -Minimalmodell aller Regeln, die greifen, sein. Aspekte davon sind zum Beispiel beschrieben in Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New generation computing* 9(3-4), 365-385 (1991).

[0045] Eine Regel mit einem leeren Körper wird ein Fakt genannt, wobei $:-$ üblicherweise weggelassen ist. Fakten werden zum Ausdrücken von Kenntnissen verwendet, die bedingungslos wahr sind. Eine Regel mit einem leeren Kopf ist eine Beschränkung. Der Körper einer Beschränkung kann durch keinen Antwortsatz erfüllt werden und wird zum Wegstreichen unerwünschter Lösungskandidaten verwendet.

[0046] Eine häufige syntaktische Erweiterung sind Auswahlregeln der folgenden Form:

$$i\{p_1, \dots, p_k\} j : -q_1, \dots, q_m, \text{ nicht } r_1, \dots, \text{ nicht } r_n.$$

[0047] Die Bedeutung ist, dass, wenn die Regel greift, auch irgendeine Teilmenge S von p_1, \dots, p_k mit $i \leq |S| \leq j$ wahr sein muss.

[0048] In einem Beispiel kann ein ASP-Löser, der Differenzlogik verwendet, zum Lösen des Optimierungsproblems verwendet werden. Ein Beispiel für den ASP-Löser ist clingo-dl (<https://github.com/potassco/clingo-dl>).

[0049] Die Differenzlogik kann zum Lösen des Optimierungsproblems unter einer Beschränkung, insbesondere einer Differenzbeschränkung, verwendet werden. Die Differenzbeschränkung in dem Beispiel ist eine Erweiterung der Form $u-v \leq d$, wobei u und v Ganzzahlvariable sind und d eine Ganzzahlkonstante ist.

[0050] Im Gegensatz zu uneingeschränkten Ganzzahlbeschränkungen können Systeme von Differenzbeschränkungen in Polynomialzeit gelöst werden. Letztgenannte können in clingo-dl unter Verwendung von Theorieatomen ausgedrückt werden. Theorieatome sind zum Beispiel beschrieben in Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with clingo 5. In: *Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2016).

[0051] Das Modell in dem Beispiel umfasst eine Problem Instanz, die durch ASP-Fakten unter Verwendung einiger feststehender Prädikatnamen beschrieben wird.

[0052] Ein Beispiel mit einer Maschine m_1 und zwei Jobs j_1, j_2 ist unten beschrieben. Die Maschine m_1 ist in der Lage, alle Jobs zu verarbeiten, und alle Freigabetermine sind 0. Eine Einrichtezeit in dem Beispiel ist 4, wenn von Job j_1 zu Job j_2 gewechselt wird, und 2, wenn von Job j_2 zu Job j_1 gewechselt wird. Beide Jobs haben eine Dauer 5. Die entsprechenden Fakten sind Folgende:

```

1   { asg(J,M) : cap(M,J) } 1 :- job(J).
2
3   before(J1,J2,M) | before(J2,J1,M) :- asg(J1,M), asg(J2,M), J1 < J2.
4
5   1 { first(J,M) : asg(J,M) } 1 :- asg(_,M).
6   1 { last(J,M) : asg(J,M) } 1 :- asg(_,M).
7   1 { next(J1,J2,M) : before(J1,J2,M) } 1 :- asg(J2,M), not first(J2,M).
8   1 { next(J2,J1,M) : before(J2,J1,M) } 1 :- asg(J2,M), not last(J2,M).
9   :- first(J1,M), before(J2,J1,M).
10  :- last(J1,M), before(J1,J2,M).
11
12  &diff{ 0 - c(J1) } <= -(T+D+S) :- asg(J1,M), next(J3,J1,M),
13  setup(J3,J1,M,S), duration(J1,M,D), release(J1,M,T).
14  &diff{ c(J2) - c(J1) } <= -(P+S) :- before(J2,J1,M), next(J3,J1,M),
15  setup(J3,J1,M,S), duration(J1,M,P).
16  &diff{ 0 - c(J1) } <= -(T+D) :- asg(J1,M), duration(J1,M,D), release(J1,M,T).
17  &diff{ c(J) - cmax } <= 0 :- job(J).
18
19  &diff{ c(J2) - c(J1) } <= -P :- before(J2,J1,M), duration(J1,M,P).
20
21  1 { span(M,T) : int(T) } 1 :- machine(M).
22  &diff{ c(J) - 0 } <= S :- asg(J,M), span(M,S).
23  #minimize{ T@T,M : span(M,T) }.

```

Eine ASP-Codierung zum Berechnen minimaler Zeitpläne ist unten unter Bezugnahme auf Zeilen des Programms, das oben angegeben ist, beschrieben. Für jede Problem Instanz I weisen die Zeitpläne von I mit einer minimalen lexikografischen Produktionsspanne eine Eins-zu-Eins-Entsprechung mit den Antwortsätzen der Regeln oben ergänzt durch die Faktendarstellung von I auf.

Die Codierung in dem Beispiel besteht aus drei Teilen:

Zeile 1-10 modellieren qualitativ machbare Abfolgen von Jobs auf Maschinen:

[0053] Zeile 1 drückt aus, dass jeder Job zu einer Maschine zugewiesen wird, die zu dessen Verarbeitung in der Lage ist. Die Bezeichnung $asg(J,M) : cap(M,J)$ bedeutet, dass im Fundierungsschritt für jeden Wert j der globalen Variable J , wie sie im Körper erscheint. Dies bedeutet, dass $asg(J,M)$ durch alle Atome $asg(j,m)$ ersetzt wird, für welche $cap(j,m)$ abgeleitet werden kann.

[0054] Zeile 3 drückt eine Regel aus, die es erfordert, dass die zu einer Maschine zugewiesenen Jobs vollkommen geordnet sind. D.h., für jegliche zwei verschiedene derartige Jobs j_1 und j_2 ist entweder $j_1 < j_2$ oder $j_2 < j_1$ wahr.

[0055] Zeile 5 definiert ein Prädikat $first(J,M)$, das einen ersten Job auf jeder Maschine darstellt.

Zeile 6 definiert ein Prädikat $last(J,M)$, das einen letzten Job auf jeder Maschine darstellt.

Zeile 7 umfasst ein Prädikat $next(J_1, J_2, M)$, das einen Nachfolger von jedem Job, außer dem letzten Job, und eine Beschränkung zu diesem Prädikat, die sicherstellt, dass eine Auswahl gemäß dem Prädikat $next(J_1, J_2, M)$ mit einer Reihenfolge kompatibel ist, die durch ein Prädikat $before(J_1, J_2, M)$ angegeben wird, darstellt.

Zeile 8 umfasst ein Prädikat $next(J_2, J_1, M)$, das einen Vorgänger von jedem Job, außer dem ersten Job, und

eine Beschränkung zu diesem Prädikat, die sicherstellt, dass eine Auswahl gemäß dem Prädikat next (J2,J1,M) mit einer Reihenfolge kompatibel ist, die durch ein Prädikat before(J2,J1,M) angegeben wird, darstellt.

[0056] Zeile 9 umfasst eine Beschränkung zu dem Prädikat first(J,M), die sicherstellt, dass eine Auswahl gemäß dem Prädikat first(J,M) mit einer Reihenfolge kompatibel ist, die durch ein Prädikat before(J2,J1 ,M) angegeben wird.

Zeile 10 umfasst eine Beschränkung zu dem Prädikat last(J,M), die sicherstellt, dass eine Auswahl gemäß dem Prädikat last(J,M) mit einer Reihenfolge kompatibel ist, die durch ein Prädikat before(J1 ,J2,M) angegeben wird.

Zeile 12-19 modellieren quantitativ Fertigstellungszeitpunkte mit Differenzlogik.

[0057] Zeile 12-16 modellieren das oben beschriebene Zeitplanungsproblem.

Zeile 17 definiert c_max als eine obere Grenze von jeglichem Fertigstellungszeitpunkt. In dem Beispiel instanziiert der ASP-Löser Ganzzahlvariable mit einem kleinsten möglichen Wert. Somit ist in jeglichem Antwortsatz, den der Löser bestimmt, c_max die tatsächliche Produktionsspanne. Zeile 19 definiert eine zusätzliche, redundante Regel. In dem Beispiel sucht der ASP-Löser in einem Suchraum nach Lösungen. Diese Regel hilft dem ASP-Löser beim weiteren Reduzieren des Suchraumes.

Zeile 21-23 definieren, wie das Lösen des Optimierungsproblems erreicht wird.

[0058] Zeile 21 definiert eine Spanne für jede Maschine. Hier wird angenommen, dass int(T) einen begrenzten Bereich von Ganzzahlen bereitstellt.

Zeile 22 setzt durch, dass Maschinen ihre Vorgänge nicht später als in der Spanne abschließen, die für sie definiert ist.

Zeile 23 definiert die Zielfunktion.

[0059] Insbesondere steuert jegliche Maschine ihre Spanne c zu einer Kostenfunktion auf einem Prioritätslevel c bei.

[0060] Die Kostenfunktion akkumuliert Beitragswerte und der Löser minimiert Antwortsätze durch einen lexikografischen Vergleich von Kostentupeln geordnet nach ihrer Priorität.

[0061] Die Differenzlogik wird in diesem Beispiel verwendet, um auszudrücken, dass Jobs in der Reihenfolge, die durch jegliches Prädikat „next“ definiert wird, auf die Maschinen gebracht werden. Die Verwendung von Differenzlogik hat den Vorteil, dass Ganzzahlarithmetik in booleschen ASP-Beschränkungen vermieden wird, welche die Größe der Fundierung erhöhen würde.

[0062] Der exakte Algorithmus wird als Optimierung der lexikografischen Produktionsspanne bezeichnet und wird unten unter Bezugnahme auf **Fig. 2** beschrieben. **Fig. 2** zeigt schematisch beispielhafte Schritte, die durch den Scheduler 102 durchgeführt werden.

[0063] Ein Zeitplan wird in dem Beispiel als eine Lösung des Optimierungsproblems durch das iterative Lösen des Optimierungsproblems bestimmt. Das Modell M wird als die Eingabe 202 an einen ASP-Löser 204 bereitgestellt. Der Zeitplan ist eine Ausgabe 206 eines ASP-Fakt-Generators 208.

[0064] Der ASP-Löser 204 ist zum Suchen nach einer Lösung, solve(M), für das Optimierungsproblem und zum Ausgeben entweder eines nicht-leeren Satzes R, der das Optimierungsproblem für M löst, oder von $R=\phi$, wenn innerhalb feststehender Ressourcengrenzen keiner gefunden wird, konfiguriert. Eine Konfiguration 210 ist zum Bereitstellen eines Parameters I in einem Bereich $1 \leq I \leq m$ für einen Iterator 212 konfiguriert. Der Iterator 214 stellt einen Zähler i an den ASP-Fakt-Generator 208 zur Verfügung, der immer dann in einem Bereich inkrementiert wird, wenn der ASP-Löser 204 einen leeren Satz ϕ ausgibt. Der Iterator 214 ist zum Starten des Zählers i mit $i=1$ in einem ersten Durchlauf und zum Inkrementieren des Zählers i um Eins, immer wenn der ASP-Löser einen leeren Satz $R=0$ ausgibt, konfiguriert. Der ASP-Fakt-Generator 208 ist zum Festlegen einer Beschränkung 216 in Abhängigkeit von jeglichem nicht-leeren Satz R konfiguriert. Die Beschränkung 216 eines ersten Durchlaufs i ist eine Eingabe für den nächsten Durchlauf $i+1$. Der ASP-Löser 204 ist zum Lösen des Optimierungsproblems für M in dem nächsten Durchlauf $i+1$ unter dieser Beschränkung 216 konfiguriert. In dem Beispiel werden die Beschränkungen von vorhergehenden Durchläufen zu

dem Modell M hinzugefügt. Der ASP-Fakt-Generator 208 ist in dem Beispiel zum Ausgeben des Zeitplans nach einem letzten Durchlauf $i \leq I$ konfiguriert.

[0065] Der exakte Algorithmus zum Festlegen der Optimierung der lexikografischen Produktionsspanne ist unten beschrieben.

[0066] Eingabe: Modell M mit m Maschinen und Parameter I mit $1 \leq I \leq m$

Ausgabe: Zeitplan R für M mit parametrisierter lexikografischer

Produktionsspanne (c_1, \dots, c_m)

$\text{solve}(M)$ ergibt eine nicht-leere Lösung R für M oder $R = \emptyset$, wenn innerhalb feststehender Ressourcengrenzen keine gefunden wird.

$\text{bound}(i \leq b)$, $\leq \in \{\leq\}$ Beschränkungen, die durchsetzen, dass $c_i \leq b$ für die lexikografische Produktionsspanne (c_1, \dots, c_m)

```

1   (c_1, ..., c_l) ← (0, ..., 0)
2   R ← solve(M)
3   für i ← 1 bis I do
4       S ← R
5       während S ≠ ∅ do
6           R ← S
7           c_i ← i-tes Element der lexikografischen Produktionsspanne von S
8           S ← solve(M ∪ bound(i ≤ c_i))
9       M ← M [ bound(i ≤ c_i)
10  ergibt R mit der lexikografischen Produktionsspanne (c_1, ..., c_l)

```

[0067] Mit diesem exakten Algorithmus können Zeitpläne mit einer minimalen lexikografischen Produktionsspanne unter Verwendung von multiplen Löser-Aufrufen und dem inkrementellen Hinzufügen von Beschränkungen berechnet werden.

[0068] Dies stellt eine Mehrzieloptimierung dar, die einen hierarchischen Abstieg mit der höchsten Priorität als erstes umfasst. Aufgrund von Symmetrien ist hierbei ein Fehlen von Lösungen üblicherweise aufwändiger für dieses Problem als eine zu finden; dies macht alternative Strategien mit weniger erwarteten Löser-Aufrufen, wie Binärsuche oder exponentiell ansteigende Suchschritte, weniger attraktiv.

[0069] In dem Beispiel werden clingo-dl und die oben beschriebene Codierung ohne die Optimierungsaussage in Zeile 21-23 zum Implementieren von $\text{solve}(M)$ verwendet. Unter Verwendung von clingo-dl, welche Multi-Shot-Lösen unterstützt, wobei Teile des Löserzustands über mehrere Läufe beibehalten werden, werden Rechenressourcen eingespart. Insbesondere ist $\text{solve}(M)$ in dem exakten Algorithmus nicht auf die Verwendung eines ASP-Lösers beschränkt. Stattdessen kann jegliches exakte Verfahren, das zum Erzeugen von Lösungen für ein Modell M in der Eingabesprache des entsprechenden Systems in der Lage ist, zum Einsatz kommen.

Die Beschränkungen für $\text{bound}(i \leq b)$ sind zum Beispiel in ASP ausgedrückt:

[0070] Die i-te Komponente der lexikografischen Produktionsspanne, die kleiner als oder gleich b ist, ist äquivalent zur Durchsetzung, dass mindestens $m-i+1$ Maschinen eine Spanne von höchstens b aufweisen. In dem Beispiel wird letzteres durch das nicht-deterministische Auswählen von $m-i+1$ Maschinen und das Durchsetzen, dass sie ihre Abläufe nicht später als b abschließen, codiert:

```

(m-i+1) { sel(M) : machine(M) }.
&diff { span(M) - 0 } <= b :- sel(M) .

```

[0071] Der exakte Algorithmus ergibt garantiert einen Zeitplan mit einer minimalen lexikografischen Produktionsspanne, wenn die Ressourcen für $\text{solve}(M)$ nicht begrenzt sind.

[0072] Ein beispielhaftes erstes Verfahren zur Zeitplanung ist unter Bezugnahme auf **Fig. 3** beschrieben. Das erste Verfahren umfasst einen Schritt 302. Bei Schritt 302 wird ein Zeitplan bestimmt, der das Optimierungsproblem für den Maschinensatz K und die Gruppe von Jobs J löst.

[0073] Das Optimierungsproblem wird durch eine Reihe von Regeln definiert, welche die Gruppe von Jobs zu dem Maschinensatz zuweisen. Der Zeitplan ordnet jeden Job der Gruppe von Jobs einer Maschine des Maschinensatzes, die zum Verarbeiten dieses Jobs in der Lage ist, zu.

[0074] Eine Ausführung von Jobs, die der Zeitplan zu einer Maschine zuweist, durch diese Maschine wird derart zeitlich geplant, dass sie innerhalb einer Maschinentzeitpanne abgeschlossen wird.

[0075] Danach wird ein Schritt 304 ausgeführt. Bei dem Schritt 304 wird eine Beschränkung für mindestens eine Maschinentzeitpanne in Abhängigkeit von der Maschinentzeitpanne festgelegt.

[0076] Danach wird ein Schritt 306 ausgeführt. Bei dem Schritt 306 wird ein Zeitplan bestimmt, der das Optimierungsproblem für die Gruppe von Jobs und den Maschinensatz unter der Beschränkung löst. Eine Ausführung von Jobs, die der Zeitplan zu einer Maschine zuweist, durch diese Maschine wird derart zeitlich geplant, dass sie innerhalb einer Maschinentzeitpanne abgeschlossen wird.

[0077] Danach wird ein Schritt 308 ausgeführt. Bei Schritt 308 wird bestimmt, ob ein Zähler von Durchläufen i einen Parameter l überschreitet oder nicht. Wenn der Zähler von Durchläufen i den Parameter l überschreitet, wird ein Schritt 310 ausgeführt. Ansonsten wird der Zähler erhöht und Schritt 304 wird ausgeführt.

[0078] Bei Schritt 310 wird der Zeitplan als eine Lösung des Optimierungsproblems ausgegeben.

[0079] In der Praxis kann eine Zeit, die mit dem Suchen nach einer Lösung für das Optimierungsproblem verbracht wird, durch eine geeignete Zeitbegrenzung begrenzt sein.

[0080] Für den angenäherten Algorithmus wird ein Optimierer, insbesondere ein exakter Optimierer verwendet, der zum Finden von Zeitplänen mit kurzen Produktionsspannen konfiguriert ist. Dieser Optimierer kann jegliche Vorgehensweise verwenden, welche gut beim Finden von Zeitplänen mit kurzen Produktionsspannen ist. Dieser Optimierer wird in dem Beispiel zum Berechnen von Zeitplänen mit kurzen lexikografischen Produktionsspannen eingesetzt, wie unten beschrieben:

Eingabe: Modell M mit m Maschinen und Parameter l mit $1 \leq l \leq m$

Ausgabe: Zeitplan S für M mit parametrisierter lexikografischer

Produktionsspanne (c_1, \dots, c_l)

$\text{opt}(M)$ ergibt die beste Lösung für M , die innerhalb feststehender Ressourcengrenzen gefunden wird

1 $(c_1, \dots, c_l) \leftarrow (0, \dots, 0)$

2 $S \leftarrow \emptyset;$

3 für $i \leftarrow 1$ bis l do

4 $S \leftarrow \text{opt}(M)$

5 $c_i \leftarrow$ Produktionsspanne von S

6 Entfernen einer Maschine k , die ihre Vorgänge bei c_i abschließt,
und aller Jobs, die k aus M zugewiesen werden

7 ergibt S

[0081] Dieser Annäherungsalgorithmus verwendet den Optimierer zum wiederholten Berechnen und Verbessern von Teilen einer Lösung, die durch den Optimierer bereitgestellt wird. Dies erfordert mehrere Löser-Aufrufe an den Optimierer. Eine Anzahl der Löser-Aufrufe ist durch die Anzahl von Maschinen begrenzt. Nach jedem Löser-Aufruf wird die Problem Instanz signifikant vereinfacht. Daher ist die Problem Instanz einfacher zu lösen.

[0082] Der Optimierer kann durch Verwendung von Multi-Shot-Lösen unter Wiederverwendung heuristischer Werte und gelernter Bedingungen aus vorhergehenden Löserläufen implementiert werden.

[0083] In diesem Beispiel kann der ASP-Fakt-Generator 208, in einem ersten Durchlauf, zum Entfernen jeglicher Maschine k , die ihre Vorgänge bei c_i abschließt, und aller Jobs, die aus M zu k zugewiesen werden, aus dem Modell M konfiguriert sein. Der ASP-Löser 204 ist in diesem Beispiel zum Aufrufen des Optimierers zum Lösen des derart geänderten Modells M konfiguriert.

[0084] Ein beispielhaftes zweites Verfahren zur Zeitplanung ist unten unter Bezugnahme auf **Fig. 4** beschrieben. Das zweite Verfahren umfasst einen Schritt 402. Bei Schritt 402 wird ein Zeitplan bestimmt, der das Optimierungsproblem für den Maschinensatz K und die Gruppe von Jobs J löst.

[0085] Danach wird ein Schritt 404 ausgeführt. Bei Schritt 404, mindestens eine Maschine k mit einer Maschinenzeitspanne, die derart zeitlich geplant ist, dass ihre Vorgänge innerhalb der Produktionsspanne des Zeitplans abgeschlossen werden.

[0086] Danach wird ein Schritt 406 ausgeführt. Bei Schritt 406 wird eine Untergruppe des Maschinensatzes K ohne die Maschine k gebildet.

[0087] Danach wird ein Schritt 408 ausgeführt. Bei Schritt 408 wird eine Untergruppe der Gruppe von Jobs J ohne die Jobs j , die der Maschine k gemäß dem Zeitplan zugewiesen werden, gebildet.

[0088] Danach wird ein Schritt 410 ausgeführt. Bei Schritt 410 wird der Zeitplan, der das Optimierungsproblem für die Untergruppe der Gruppe von Jobs J und die Untergruppe des Maschinensatzes K löst, bestimmt.

[0089] Danach wird ein Schritt 412 ausgeführt. Bei Schritt 412 wird bestimmt, ob ein Zähler von Durchläufen i einen Parameter I überschreitet oder nicht. Wenn der Zähler von Durchläufen i den Parameter I überschreitet, wird ein Schritt 414 ausgeführt. Ansonsten wird der Zähler erhöht und Schritt 404 wird ausgeführt.

[0090] Bei Schritt 414 wird der Zeitplan als eine Lösung des Optimierungsproblems ausgegeben.

Patentansprüche

1. Verfahren, insbesondere ein computerimplementiertes Verfahren, zur Maschinenzeitplanung zum Ausführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz (104), **gekennzeichnet durch** das Bestimmen (302, 402), in einem ersten Durchlauf, eines ersten Zeitplans, der ein Optimierungsproblem löst, wobei das Optimierungsproblem durch eine Reihe von Regeln definiert wird, welche die Gruppe von Jobs zu dem Maschinensatz (104) zuweisen, wobei der erste Zeitplan jeden Job der Gruppe von Jobs einer Maschine des Maschinensatzes zuordnet, die zum Verarbeiten dieses Jobs in der Lage ist, wobei eine Ausführung von Jobs, die der erste Zeitplan einer Maschine zuweist, durch diese Maschine zeitlich derart geplant wird, dass sie innerhalb einer Maschinenzeitspanne abgeschlossen wird, und, in einem zweiten Durchlauf, entweder das Festlegen (304) einer Beschränkung für mindestens eine Maschinenzeitspanne und das Bestimmen (306) eines zweiten Zeitplans, der das Optimierungsproblem für die Gruppe von Jobs und den Maschinensatz (104) unter der Beschränkung löst, oder das Bestimmen (410) eines zweiten Zeitplans, der das Optimierungsproblem für eine Untergruppe der Gruppe von Jobs und eine Untergruppe des Maschinensatzes (104) löst, wobei die Untergruppe des Maschinensatzes (104) die Maschine ausschließt und die Untergruppe der Gruppe von Jobs ausschließt, die der Maschine gemäß dem ersten Zeitplan zugewiesen werden.

2. Verfahren nach Anspruch 1, **gekennzeichnet durch** das Auswählen einer Maschine aus dem Maschinensatz (104) mit einer Maschinenzeitspanne, die bei einer Produktionsspanne zum Ausführen der Aufgabe endet, gemäß dem ersten Zeitplan und das Bilden (406) der Untergruppe des Maschinensatzes (104) ohne mindestens eine Maschine.

3. Verfahren nach Anspruch 2, **gekennzeichnet durch** das Bilden (408) der Untergruppe der Gruppe von Jobs ohne mindestens einen Job, welcher der Maschine zugewiesen wird.

4. Verfahren nach Anspruch 1 oder 2, **gekennzeichnet durch** das Festlegen mehrerer Maschinenzeitspannen für Maschinen des Maschinensatzes (104), die zeitlich derart geplant sind, dass sie mindestens einen Job gemäß dem ersten Zeitplan ausführen, wobei eine Maschinenzeitspanne mit einer längeren

Dauer als mindestens eine andere Maschinenzeitspanne der mehreren Maschinenzeitspannen als die Beschränkung aus den mehreren Maschinenzeitspannen ausgewählt wird.

5. Verfahren nach einem der vorhergehenden Ansprüche, **gekennzeichnet durch** das Festlegen einer Abfolge, in welcher Jobs, die der Maschine zugewiesen werden, zeitlich für die Verarbeitung auf der Maschine geplant werden, und das Festlegen der Maschinenzeitspanne in Abhängigkeit von einem Fertigstellungszeitpunkt eines letzten Jobs in der Abfolge.

6. Verfahren nach Anspruch 5, **gekennzeichnet durch** das Festlegen des Fertigstellungszeitpunktes des letzten Jobs in Abhängigkeit von einer Summe einer Startzeit eines ersten Jobs in der Abfolge und einer Summe der Verarbeitungszeiten von mindestens einem Job in der Abfolge, dessen Start zeitlich früher geplant ist als der letzte Job, und einer Dauer des letzten Jobs.

7. Verfahren nach einem der vorhergehenden Ansprüche, **gekennzeichnet durch** das Bestimmen mehrerer Zeitpläne in mehreren Durchläufen, unter Begrenzung einer Anzahl von Durchläufen auf weniger als eine Anzahl von Maschinen in den mehreren Maschinen.

8. Verfahren nach einem der vorhergehenden Ansprüche, **gekennzeichnet durch** das zeitliche Planen von mindestens einer Maschine des Maschinensatzes (104) zum Ausführen des mindestens einen Jobs der Gruppe von Jobs gemäß dem zweiten Zeitplan.

9. Scheduler (102) zum Ausführen einer Gruppe von Jobs einer Aufgabe mit einem Maschinensatz (104) während einer Produktionsspanne, **dadurch gekennzeichnet**, dass der Scheduler (102) zum Ausführen der Schritte des Verfahrens nach einem der Ansprüche 1 bis 8 konfiguriert ist.

10. Computerprogramm, **dadurch gekennzeichnet**, dass das Computerprogramm computerlesbare Anweisungen umfasst, die, wenn sie durch einen Computer ausgeführt werden, den Computer zum Durchführen der Schritte in dem Verfahren gemäß einem der Ansprüche 1 bis 8 veranlassen.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

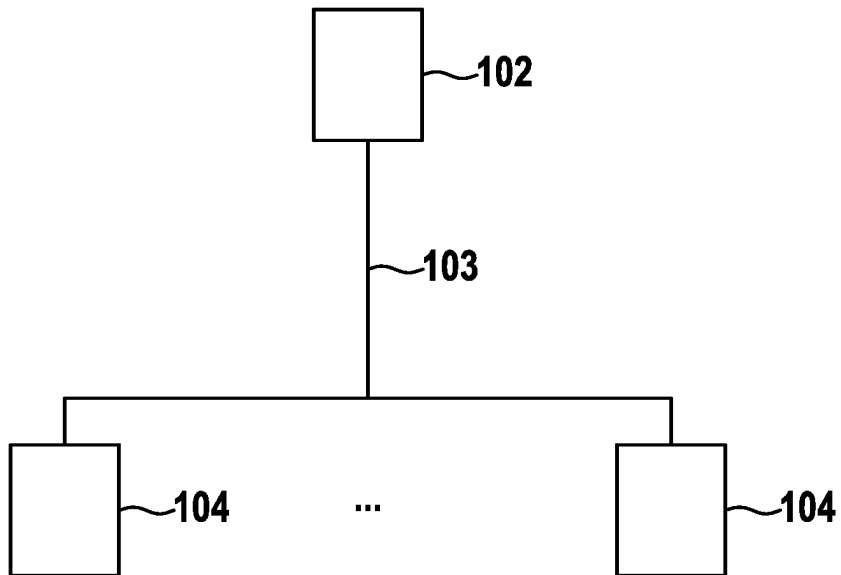


Fig. 1

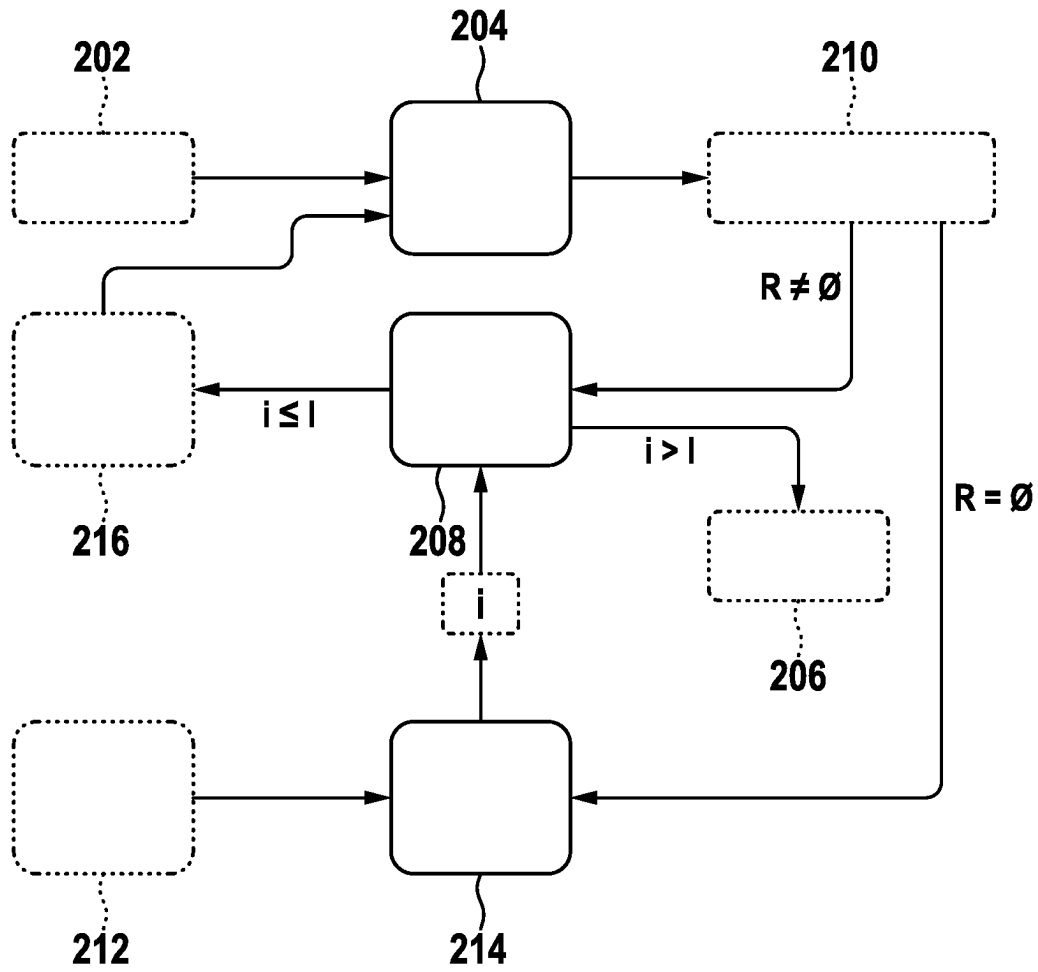


Fig. 2

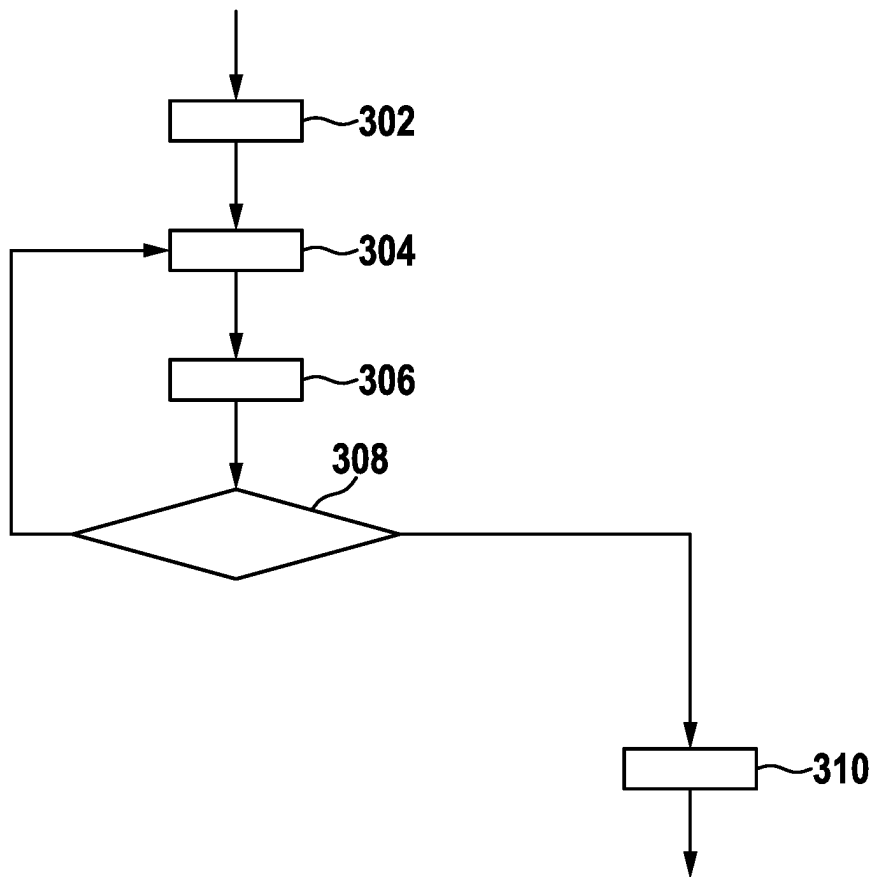


Fig. 3

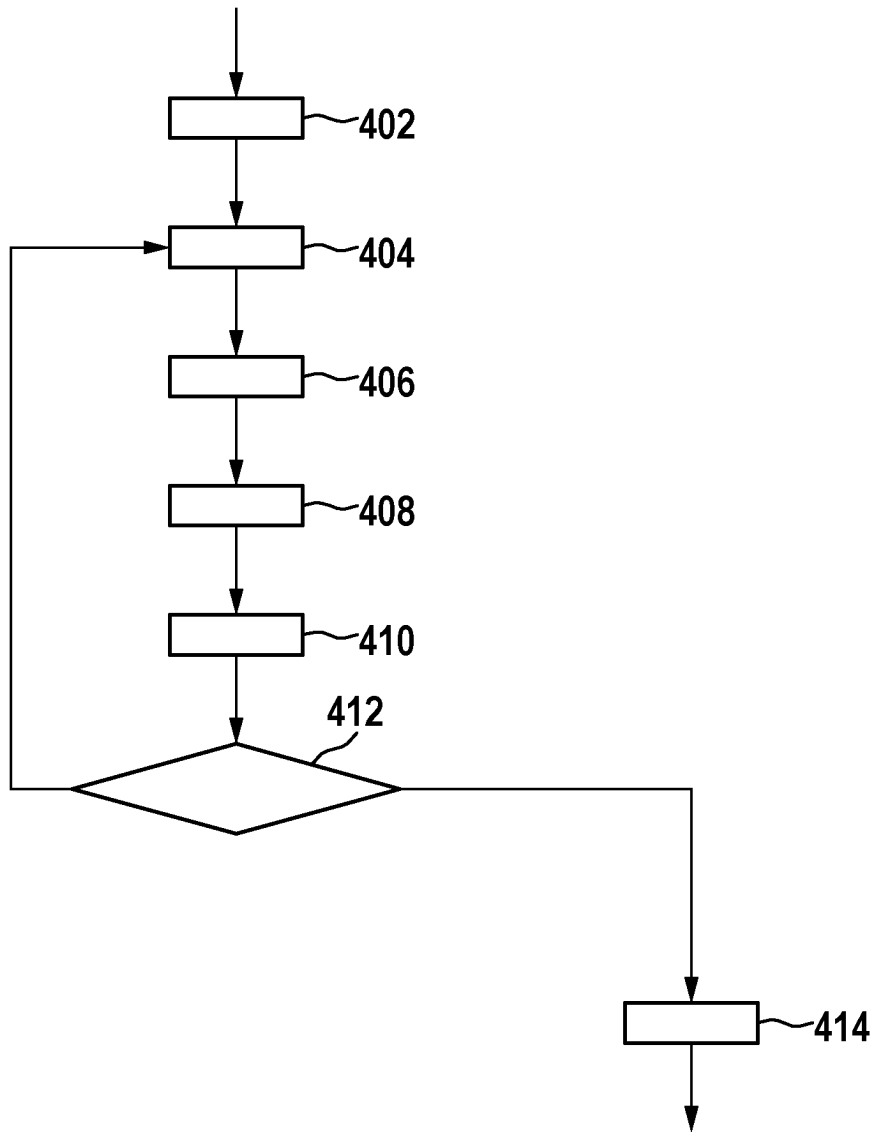


Fig. 4