

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号

特開2022-167845

(P2022-167845A)

(43)公開日 令和4年11月4日(2022.11.4)

(51)国際特許分類

F I

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/50 1 5 0 A

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/48 3 0 0 F

審査請求 未請求 請求項の数 10 O L 外国語出願 (全18頁)

(21)出願番号	特願2022-70067(P2022-70067)	(71)出願人	390023711
(22)出願日	令和4年4月21日(2022.4.21)		ローベルト ボツシュ ゲゼルシャフト
(31)優先権主張番号	10 2021 204 043.8		ミット ベシユレンクテル ハフツング
(32)優先日	令和3年4月22日(2021.4.22)		ROBERT BOSCH GMBH
(33)優先権主張国・地域又は機関	ドイツ(DE)		ドイツ連邦共和国 シュツツトガルト (番地なし)
			Stuttgart, Germany
		(74)代理人	100114890
			弁理士 アインゼル・フェリックス＝ラ
			インハルト
		(74)代理人	100098501
			弁理士 森田 拓
		(74)代理人	100116403
			弁理士 前川 純一
		(74)代理人	100134315
最終頁に続く			

(54)【発明の名称】 機械の集合を用いてタスクのジョブの集合を実行するためのスケジューラ、及び、特にコンピュータ実装された、機械スケジューリングの方法

(57)【要約】 (修正有)

【課題】タスクのジョブの集合を実行する機械スケジューリングの方法を提供する。

【解決手段】方法は、第1の反復において、最適化問題を解決する第1のスケジュールを決定する。最適化問題は、ジョブの集合を機械の集合に割り当てるルール集合によって定義される。第1のスケジュールは、ジョブの集合のうちのそれぞれのジョブを、機械の集合のうちの、当該ジョブを処理することができる1つの機械にマッピングする。1つの機械に割り当てたジョブの、当該機械による実行は、1つの機械スパン内に終了するようにスケジューリングされる。方法はまた、第2の反復において、少なくとも1つの機械スパンに関する制約を決定して、制約の下でジョブの集合及び機械の集合に関する最適化問題を解決する第2のスケジュールを決定するか又はジョブの集合のうちの部分集合及び機械の集合のうちの部分集合に関する最適化問題を解決する第2のスケジュールを決定する。

【選択図】なし

【特許請求の範囲】

【請求項 1】

機械の集合（１０４）を用いてタスクのジョブの集合を実行するための、特にコンピュータ実装された、機械スケジューリングの方法であって、

第１の反復において、

最適化問題を解決する第１のスケジュールを決定（３０２，４０２）し、

前記最適化問題は、前記ジョブの集合を前記機械の集合（１０４）に割り当てるルールの集合によって定義され、

前記第１のスケジュールは、前記ジョブの集合のうちのそれぞれのジョブを、前記機械の集合のうちの、当該ジョブを処理することができる１つの機械にマッピングし、

前記第１のスケジュールが或る１つの機械に割り当てたジョブの、当該機械による実行は、１つの機械スパン内に終了するようにスケジューリングされ、

第２の反復において、

少なくとも１つの機械スパンに関する制約を決定（３０４）して、

前記制約の下で前記ジョブの集合及び前記機械の集合（１０４）に関する最適化問題を解決する第２のスケジュールを決定（３０６）し、又は、

前記ジョブの集合のうちの部分集合及び前記機械の集合（１０４）のうちの部分集合に関する最適化問題を解決する第２のスケジュールを決定（４１０）し、

前記機械の集合（１０４）のうちの前記部分集合は、前記機械を除外し、前記ジョブの集合のうちの前記部分集合は、前記第１のスケジュールに従って前記機械に割り当てられたジョブを除外することを特徴とする方法。

【請求項 2】

前記機械の集合（１０４）から、前記第１のスケジュールに従って前記タスクを実行するためのメイクスパンで終了する機械スパンを有する機械を選択し、

前記機械の集合（１０４）のうちの、少なくとも１つの機械を含まない部分集合を形成（４０６）する、

請求項 1 に記載の方法。

【請求項 3】

前記ジョブの集合のうちの、前記機械に割り当てられた少なくとも１つのジョブを含まない部分集合を形成（４０８）する、

請求項 2 に記載の方法。

【請求項 4】

前記機械の集合（１０４）のうちの、前記第１のスケジュールに従って少なくとも１つのジョブを実行するようにスケジューリングされた複数の機械のための複数の機械スパンを決定し、

前記複数の機械スパンのうちの少なくとも１つの他の機械スパンよりも長い持続時間を有する機械スパンが、前記複数の機械スパンから制約として選択される、

請求項 1 又は 2 に記載の方法。

【請求項 5】

前記機械に割り当てられたジョブが当該機械上において処理されるようにスケジューリングされるシーケンスを決定し、

前記シーケンス内の最後のジョブの完了時間に依存して、前記機械スパンを決定する、請求項 1 乃至 4 のいずれか一項に記載の方法。

【請求項 6】

前記最後のジョブの完了時間を、前記シーケンス内の最初のジョブの開始時間の合計と、前記最後のジョブよりも早期に開始するようにスケジューリングされている前記シーケンス内の少なくとも１つのジョブの処理時間の合計と、前記最後のジョブの持続時間とに依存して決定する、

請求項 5 に記載の方法。

10

20

30

40

50

【請求項 7】

反復の回数を、前記複数の機械における機械の数よりも少なくなるように制限しながら、複数の反復において複数のスケジュールを決定する、
請求項 1 乃至 6 のいずれか一項に記載の方法。

【請求項 8】

前記第 2 のスケジュールに従って前記ジョブの集合のうちの少なくとも 1 つのジョブを実行するように、前記機械の集合（104）のうちの少なくとも 1 つの機械をスケジューリングする、
請求項 1 乃至 7 のいずれか一項に記載の方法。

【請求項 9】

機械の集合（104）を用いてタスクのジョブの集合をメイクスパン内において実行するためのスケジューラ（102）であって、

請求項 1 乃至 8 のいずれか一項に記載の方法の各ステップを実施するように構成されていることを特徴とするスケジューラ（102）。

【請求項 10】

コンピュータによって実行された場合に前記コンピュータに請求項 1 乃至 8 のいずれか一項に記載の方法の各ステップを実施させるためのコンピュータ可読命令を含むことを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

背景

本発明は、機械の集合を用いてタスクのジョブの集合を実行するためのスケジューラと、特にコンピュータ実装された、機械スケジューリングの方法とに関する。

【背景技術】

【0002】

Dimitrios Letsios, Miten Mistry, Ruth Misener 著 “Exact lexicographic scheduling and approximate rescheduling”, European Journal of Operational Research Volume 290, Issue 2, 2021 年 4 月 16 日、第 469 – 478 頁は、関連するスケジューリングアプローチの態様を開示している。

【0003】

米国特許出願公開第 2020 / 0320456 号明細書、独国特許発明第 69513877 号明細書翻訳、独国特許出願公開第 102016204392 号明細書及び米国特許第 9798947 号明細書から、機械スケジューリングのためのさらなる方法が公知である。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】米国特許出願公開第 2020 / 0320456 号明細書

【特許文献 2】独国特許発明第 69513877 号明細書翻訳

【特許文献 3】独国特許出願公開第 102016204392 号明細書

【特許文献 4】米国特許第 9798947 号明細書

【非特許文献】

【0005】

【非特許文献 1】Dimitrios Letsios, Miten Mistry, Ruth Misener 著 “Exact lexicographic scheduling and approximate rescheduling”, European Journal of Operational Research Volume 290, Issue 2, 2021 年 4 月 16 日、第 469 – 478 頁

【発明の概要】

【発明が解決しようとする課題】

【0006】

発明の開示

独立請求項に記載のスケジューラと、コンピュータプログラムと、特にコンピュータ実装された、機械スケジューリングの方法とは、スケジューリングをさらに改善して、機械の割り当てと、シーケンス依存のセットアップ時間と、リリース日とが関与している比較的複雑なスケジューリング問題のために直接的に使用可能にし、要件が変化した場合にスケジューリング問題の仕様を維持することをより容易にするという課題を解決する。

【課題を解決するための手段】

【0007】

機械の集合を用いてタスクのジョブの集合を実行するための、特にコンピュータ実装された、機械スケジューリングの方法は、第1の反復において、最適化問題を解決する第1のスケジュールを決定することを含み、最適化問題は、ジョブの集合を機械の集合に割り当てるルール集合によって定義され、第1のスケジュールは、ジョブの集合のうちのそれぞれのジョブを、機械の集合のうちの、当該ジョブを処理することができる1つの機械にマッピングし、第1のスケジュールが或る1つの機械に割り当てたジョブの、当該機械による実行は、1つの機械スパン内に終了するようにスケジューリングされ、当該方法は、第2の反復において、少なくとも1つの機械スパンに関する制約を決定して、制約の下でジョブの集合及び機械の集合に関する最適化問題を解決する第2のスケジュールを決定し、又は、ジョブの集合のうちの部分集合及び機械の集合のうちの部分集合に関する最適化問題を解決する第2のスケジュールを決定することを含み、機械の集合のうちの部分集合は、上記の機械を除外し、ジョブの集合のうちの部分集合は、第1のスケジュールに従って機械に割り当てられたジョブを除外する。本方法は、機械の割り当てと、シーケンス依存のセットアップ時間と、リリース日とが関与している比較的複雑なスケジューリング問題のために直接的に使用可能である。さらに、他の利点は、それぞれのモデル化言語を用いた従来技術による知識表現及び推論ソルバに本方法を適用することができるので、要件が変化した場合にスケジューリング問題の仕様を維持することがより容易であることである。

【0008】

当該方法は、機械の集合から、第1のスケジュールに従ってタスクを実行するためのメイクスパンで終了する機械スパンを有する機械を選択することと、機械の集合のうちの、少なくとも1つの機械を含まない部分集合を形成することとを含み得る。これにより、機械に関する最適化問題の複雑さが軽減される。

【0009】

当該方法は、ジョブの集合のうちの、機械に割り当てられた少なくとも1つのジョブを含まない部分集合を形成することを含み得る。これにより、ジョブに関する最適化問題の複雑さが軽減される。

【0010】

当該方法は、機械の集合のうちの、第1のスケジュールに従って少なくとも1つのジョブを実行するようにスケジューリングされた複数の機械のための複数の機械スパンを決定することを含み得る。そこでは、複数の機械スパンのうちの少なくとも1つの他の機械スパンよりも長い持続時間を有する機械スパンが、複数の機械スパンから制約として選択される。これにより、他の機械スパンよりも長い持続時間を有する機械スパンによって、最適化問題の解が制約される。これにより、解が非常に効率的に決定される。

【0011】

当該方法は、機械に割り当てられたジョブが当該機械上において処理されるようにスケジューリングされるシーケンスを決定することと、シーケンス内の最後のジョブの完了時間に依存して、機械スパンを決定することとを含み得る。これにより、スケジューリング方法のロバスト性を向上させるための機械スパンを決定することが可能となる。

【0012】

当該方法は、最後のジョブの完了時間を、シーケンス内の最初のジョブの開始時間の合

10

20

30

40

50

計と、最後のジョブよりも早期に開始するようにスケジューリングされているシーケンス内の少なくとも1つのジョブの処理時間の合計と、最後のジョブの持続時間とに依存して決定することを含み得る。これにより、最後のジョブの完了時刻が非常に精確に決定される。これにより、スケジューリング方法のロバスト性がさらに向上する。

【0013】

当該方法は、反復の回数を、複数の機械における機械の数よりも少なくなるように制限しながら、複数回の反復において複数のスケジュールを決定することを含み得る。これにより、スケジューリングが格段に高速になる。

【0014】

当該方法は、第2のスケジュールに従ってジョブの集合のうちの少なくとも1つのジョブを実行するように、機械の集合のうちの少なくとも1つの機械をスケジューリングすることを含み得る。このようにすると、機械が非常に効率的にスケジューリングされる。

10

【0015】

機械の集合を用いてタスクのジョブの集合をメイクスパン内において実行するためのスケジューラは、本方法の各ステップを実施するように構成されている。

【0016】

コンピュータプログラムは、コンピュータによって実行された場合にコンピュータに本方法の各ステップを実施させるためのコンピュータ可読命令を含む。

【0017】

さらなる有利な実施形態は、以下の説明及び図面から導出可能である。

20

【図面の簡単な説明】

【0018】

【図1】機械の集合のためのスケジューラの少なくとも一部を概略的に示す図である。

【図2】スケジューラによって実行されるステップを概略的に示す図である。

【図3】第1の方法におけるステップを概略的に示す図である。

【図4】第2の方法におけるステップを概略的に示す図である。

【発明を実施するための形態】

【0019】

図1は、機械の集合Kのためのスケジューラ102の少なくとも一部を概略的に示している。スケジューラ102は、少なくとも1つのプロセッサと、少なくとも1つのメモリ、特に、スケジューラ102の少なくとも1つのプロセッサによって実行された場合に、スケジューラ102に、以下において説明される、特にコンピュータ実装されたスケジューリング方法を実行させるコンピュータ可読命令を格納するためのメモリとを含み得る。機械の集合（以下においては参照符号Kが付されており、図示しない）は、複数の機械104を表している。本例におけるスケジューラ102と、複数の機械104のうちの各機械とは、少なくとも一時的にデータリンク103を介して接続されている。本例におけるスケジューラ102は、複数の機械104のうちの少なくともいくつかの機械に、ジョブの集合（以下においては参照符号Jが付されており、図示しない）のうちの各ジョブを実行するように命令するために構成されている。

30

【0020】

本方法は、スケジューリング問題を解決する。

40

【0021】

機械の集合K内のm個の機械kと、ジョブの集合J内のn個のジョブjとが与えられている場合には、いずれのジョブjも、単一の機械kによって処理される必要があり、いずれの機械kも、一度に最大で1つのジョブjを処理することができるものとする。即ち、プリエンブションは、許可されていない。

【0022】

機械kの中には、特定のジョブjしか扱うことができないものもある。即ち、cap(j)は、ジョブjを処理することができる機械の集合である。

【0023】

50

それぞれのジョブ j 及び機械 k ごとにリリース日 $r_{j;k}$ が、非負の整数として指定される。リリース日は、機械に依存している。なぜなら、機械へのジョブの輸送時間は、輸送システムと、輸送システムの場所とに依存しているからである。いずれのジョブ j も、ジョブのリリース日 $r_{j;k}$ より前に開始することはできない。

【0024】

ある1つのジョブ j から次のジョブ j に変更するためには、指定された長さの時間が必要とされる場合がある。時間 $s_{i;j;k}$ は、機械 k 上においてジョブ i の直後にジョブ j をセットアップするために必要とされる時間である。

【0025】

これらの時間は、シーケンス依存のセットアップ時間と称される。それぞれのジョブ j は、そのジョブ j が割り当てられた機械 k に依存している持続時間 $d_{j;k}$ を有する。

【0026】

問題インスタンスのためのスケジュールは、

1. それぞれのジョブ j を、そのジョブ j を処理することができる機械 $k \in \text{cap}(j)$ にマッピングする割り当て a 、
2. それぞれの機械 k ごとの、割り当て a によってその機械 k に割り当てられたジョブ j の集合 J に対する全順序

【数1】

$\preceq_k$

によって定義される。

【0027】

関係性

【数2】

$\preceq_k$

は、ジョブの集合 J 内のジョブ j が機械 k 上において処理されるシーケンスを決定する。

【0028】

それぞれのジョブ j を何らかの機械 k によって処理することができる場合には、問題インスタンスに対する少なくとも1つのスケジュールが存在する。

【0029】

$j_1, \dots, j_l$ が、所与のスケジュールにおいて機械 k に割り当てられたジョブ j の処理シーケンスであると仮定する。ジョブ j_i の処理時間 p_{j_i} は、このジョブ j_i の持続時間に、このジョブ j_i の先行ジョブ（存在する場合）のためのセットアップ時間を足したものであり、即ち、 $i > 1$ である場合、 $p_{j_1} = d_{j_1;k}$ 及び $p_{j_i} = s_{j_{i-1};j_i;k} + d_{j_i;k}$ である。ジョブ j_i の開始時刻 st_{j_i} は、 $i = 1$ である場合、 $r_{j_1;k}$ であり、 $i > 1$ である場合、 $\max(r_{j_i;k}, st_{j_{i-1}} + p_{j_{i-1}})$ である。ジョブ j_i の完了時刻 c_{j_i} は、 $st_{j_i} + p_{j_i}$ である。機械 k の機械スパン $span(k)$ は、機械 k 上における最後のジョブ j_l の完了時間である。

【0030】

目標は、本例においては単位時間あたりに処理されるジョブ j の数として定義されているスループットを最大化することである。原則として、メイクスパン、即ち、スケジュールの最大の機械スパンを最小化することにより、このことが達成される。しかしながら、高い割り当て率を有する機械に対処する場合には、ほんの少数の機械のみによって多数のジョブが処理され得るということがよく見受けられる。このことはつまり、残余の機械上においてジョブがどのようにスケジューリングされるかにほとんど関係なく、少数の機械によってメイクスパンが決定されてしまうということを意味する。このことは、例えば、いつでも起こり得る突然の機械故障のせいでジョブを動的に再スケジュールする必要があ

10

20

30

40

50

る場合には、理想的ではない。そのような場合、ドメイン専門家は、「全ての機械は、可能な限り早期に完了すべきである」という要件を示した。このことは、スケジューラに再配置のための自由を与え、高いスループットを維持することを支援する。

【0031】

この問題に対処するために、本方法は、全ての機械が可能な限り早期に完了するという希望を捉えることを目指した目的関数を使用する。目的関数は、以下においては辞書式メイクスパン (lexicographical makespan) と称される。

【0032】

スケジュールの辞書式メイクスパンは、非昇順での全ての機械スパンのタプルとして定義されている。本方法は、比較のために種々異なるスケジュールのタプルの辞書式順序を使用し、より大きい辞書式メイクスパンを有するスケジュールよりも、より小さい辞書式メイクスパンを有するスケジュールの方を優先する。結果として生じるスケジュールは、最小の辞書式メイクスパンを有するスケジュールである。従って、結果として生じるスケジュールは、最小のメイクスパンも有するが、より早期に完了する機械を使用することによって同順位が解消される。

【0033】

最小の辞書式メイクスパンを有するスケジュールを計算するために、例示的な厳密アルゴリズムと、例示的な近似アルゴリズムとを以下に説明する。両方とも解集合プログラミング (ASP) を使用しており、ASPは、例えば、以下の文献に記載されている宣言的な問題解決のためのロジックベースのアプローチである。

【0034】

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

【0035】

辞書式メイクスパンと称される目的関数は、全ての機械が可能な限り早期に完了すべきであるという要件を定式化したものである。

【0036】

m 個の機械が含まれているスケジュール S が与えられている場合には、 S の辞書式メイクスパンは、非昇順での S の全ての機械スパン c_i のタプル $ms(S) = (c_1, \dots, c_m)$ である。

【0037】

この定義においては、 c_1 は、最大の機械スパンであり、従って、スケジュール S に即したメイクスパンに相当する。それぞれ m 個の機械が含まれているスケジュール S 及び S' では、 $ms(S)$ が辞書式順序下で $ms(S')$ より小さい場合、 S は、 S' より小さい辞書式メイクスパンを有する。このことはつまり、最小の添え字 i に関して $c_i < c'_i$ であることを意味しており、ここで、 $ms(S) = (c_1, \dots, c_m)$ と、 $ms(S') = (c'_1, \dots, c'_m)$ とは、一致しない。

【0038】

スケジュールの集合 S に関して、集合 S 内の全てのスケジュールにわたって $ms(S)$ が最小である場合、 $S \in S$ が最適となる。

【0039】

スケジュール S のダイナミクスは、時点 t に又は時点 t よりも前に完了する機械 $M(S; t)$ の数として記述される。

【0040】

S 及び S' が、いくつかの問題インスタンスのための2つの、特にそれぞれ異なるスケジュールである場合には、 $M(S; t) > M(S'; t)$ であって、かつ、全ての $t' > t$

10

20

30

40

50

に対して $M(S; t') \geq M(S'; t')$ であるような時点 t が存在する場合、 $m_s(S) < m_s(S')$ である。

【0041】

多数の機械が関与している問題においては、メイクスパン全体が少数の機械のみによって支配されている場合、全ての機械スパンを階層的に最小化することは過剰である可能性がある。しかしながら、辞書式メイクスパン同士を比較することにより、比較において考慮すべきコンポーネントの数を定義する整数 l を用いたパラメータ化が可能となる。

【0042】

それぞれ m 個の機械が含まれていて、かつ、 $1 \leq l \leq m$ である整数 l を有するスケジュール S 及び S' の場合、スケジュール S に即したメイクスパン $m_s(S) = (c_1, \dots, c_m)$ は、パラメータ化された辞書式順序下でのスケジュール S' に即したメイクスパン $m_s(S') = (c'_1, \dots, c'_m)$ よりも小さい。 m 個の機械を有するスケジュールの場合、 $l=1$ の場合には、メイクスパンが得られ、 $l=m$ の場合には、完全な辞書式メイクスパンが得られる。

【0043】

本例においては、ジョブの集合 J を機械の集合 K に割り当てるルールを含むモデルを用いて、問題ステートメントがモデル化されている。スケジュールは、ジョブの集合のうちのそれぞれのジョブ j を、機械の集合 K のうちの、そのジョブ j を処理することができる1つの機械 $k \in \text{cap}(j)$ にマッピングする。スケジュールが1つの機械 k に割り当てたジョブ j の、この機械 k による実行は、この機械 k の機械スパン c_i 内に終了するようにスケジューリングされる。

【0044】

上述したモデル及び目的関数は、例えば、Janhunen, T., Kaminski, R., Ostrowski, M., Schellhorn, S., Wanko, P., Schaub, T. 著 “Clingo goes linear constraints over reals and integers”、Theory and Practice of Logic Programming 17(5-6)、第 872–888 頁 (2017 年) に記載されているような、差分ロジックを用いた ASP の拡張によってモデル化可能である。

【0045】

モデル及び目的関数によって定義された最適化問題を厳密アルゴリズムによって解決するために、マルチショット解決を使用して、ASP ソルバに対する複数の呼び出しをリソースフレンドリに実施することができる。このことは、例えば、Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T. 著 “Multi-shot asp solving with clingo”、Theory and Practice of Logic Programming 19(1)、第 27–82 頁 (2019 年) に記載されている。

【0046】

最適化問題を近似アルゴリズムによって解決するために、解の一部をソルバ呼び出し後に固定することができ、これにより、近最適解を短時間で見つけることが可能となる。

【0047】

スケジューラ及びスケジューリング方法は、半導体製造に適用可能である。この態様においては、機械の集合 K は、半導体を製造するために使用される。スケジューラ及びスケジューリング方法は、分散コンピューティングシステムにも同様に適用可能である。この態様における機械は、プロセッサ、又は、プロセッサのコンピューティングコア、又は、計算動作を実行するための専用ハードウェアであるものとしてよい。ジョブは、実行可能なスケジュールを計算するための一定の制約の下で機械に割り当てられる。要件のうちの1つは、全ての機械が可能な限り早期に完了するようにスケジュールが構築されるべきであるということである。この要件に対処して、所望のスケジュールを計算するために、辞書式メイクスパンと称される最適化関数と、この最適化関数を最小化するための以下において説明されるアルゴリズムとを使用することができる。最適化問題を解決するスケジュールに従って、機械に命令することができる。

【0048】

10

20

30

40

50

A S P は、本質的に命題的なコンパクトな関係形式であり、ここでは、入力言語における変数が、基礎化と称される前処理ステップにおいて一定の記号に置き換えられる。A S P プログラムは、以下の形態

$p_1 \mid \dots \mid p_k : -q_1, \dots, q_m, \text{not } r_1, \dots, \text{not } r_n$

のルール集合であり、ここで、全ての p_i , q_j , 及び r_l は、アトムである。ヘッドは、含意記号 $:$ の前の全てのアトムであり、ボディは、その後の全てのアトム及び否定アトムである。このルールが直感的に意味することは、全てのアトム $q_1, \dots, q_m$ を導出することができ、かつ、アトム $r_1, \dots, r_n$ のいずれについてもエビデンスがない場合（即ち、ルールが発火している）には、 $p_1, \dots, p_k$ のうちの少なくとも1つが真でなければならないということである。解釈 I は、アトムの集合である。この解釈 I は、プログラムの全てのルールが最小限かつ無矛盾に満たされている場合には、プログラムの解集合である。 I は、発火している全てのルールの $\subseteq$ 最小モデルでなくてはならない。この態様は、例えば、Gelfond, M., Lifschitz, V. 著 “Classical negation in logic programs and disjunctive databases”, New generation computing 9(3-4), 第365–385頁（1991年）に記載されている。

【0049】

空のボディを有するルールは、ファクトと称され、通常、 $:$ は省略される。ファクトは、無条件に真である知識を表現するために使用される。空のヘッドを有するルールは、制約である。制約のボディは、いかなる解集合によっても満たすことができず、不所望な解候補をブルーニングするために使用される。

【0050】

共通の構文拡張は、以下の形態

$i \{p_1, \dots, p_k\} j : -q_1, \dots, q_m, \text{not } r_1, \dots, \text{not } r_n$

の選択ルールである。

【0051】

この意味は、ルールが発火している場合、 $i \leq |S| \leq j$ である $p_1, \dots, p_k$ の何らかの部分集合 S も真でなければならないということである。

【0052】

一例においては、最適化問題を解決するために、差分ロジックを使用する A S P ソルバを使用することができる。A S P ソルバについての一例は、`clingo-dl` (<https://github.com/potassco/clingo-dl>) である。

【0053】

差分ロジックは、制約の下で、特に差分制約の下で最適化問題を解決するために使用可能である。本例における差分制約は、 $u - v \leq d$ の形態の式であり、ここで、 u 及び v は、整数変数であり、 d は、整数定数である。

【0054】

制限されていない整数制約とは対照的に、差分制約のシステムは、多項式時間において解決可能である。後者は、理論アトムを用いて `clingo-dl` において表現可能である。理論アトムは、例えば、Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P. 著 “Theory solving made easy with clingo 5”, In: Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016年) に記載されている。

【0055】

本例におけるモデルは、いくつかの固定の述語名を使用して A S P ファクトによって記述される問題インスタンスを含む。

【0056】

以下においては、1つの機械 m_1 と、2つのジョブ j_1, j_2 とを用いる例について説明する。機械 m_1 は、全てのジョブを処理することが可能であり、全てのリリース日は、

0 である。本例におけるセットアップ時間は、ジョブ j_1 からジョブ j_2 に変更する場合には 4 であり、ジョブ j_2 からジョブ j_1 に変更する場合には 2 である。どちらのジョブも、持続時間 5 を有する。対応するファクトは、以下の通りである。

【表 1】

```

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) }.
```

【0057】

以下においては、最小のスケジュールを計算するための ASP エンコーディングについて、上記のプログラムの各行を参照しながら説明する。全ての問題インスタンス I に対して、最小の辞書式メイクスパンを有する I のスケジュールは、 I のファクト表現によって増補される上記のルール of 解集合に一対一で対応する。

【0058】

本例におけるエンコーディングは、以下の 3 つの部分から成る。

【0059】

第 1 行～第 10 行は、機械上におけるジョブの実行可能なシーケンスを定性的にモデル化する。

【0060】

第 1 行は、それぞれのジョブが、そのジョブを処理することができる機械に割り当てられることを表す。表記 $asg(J, M) : cap(M, J)$ は、ボディにおいて発生するような、大域変数 J のそれぞれの値 j についての基礎化ステップを意味する。このことはつまり、 $asg(J, M)$ が、 $cap(j, m)$ を導出することができる全てのアトム $asg(j, m)$ によって置き換えられることを意味する。

【 0 0 6 1 】

第 3 行は、1 つの機械に割り当てられたジョブが完全に順序付けられることを要求するルールを表す。即ち、任意の 2 つの別個のこのようなジョブ j_1 及び j_2 について、

【 数 3 】

$$j_1 < j_2 \text{ または } j_2 < j_1$$

が当てはまる。

【 0 0 6 2 】

第 5 行は、それぞれの機械上における最初のジョブを表す述語 $f i r s t (J, M)$ を定義する。 10

第 6 行は、それぞれの機械上における最後のジョブを表す述語 $l a s t (J, M)$ を定義する。

第 7 行は、最後のジョブを除くそれぞれのジョブの後続ジョブを表す述語 $n e x t (J_1, J_2, M)$ と、述語 $n e x t (J_1, J_2, M)$ による選択が述語 $b e f o r e (J_1, J_2, M)$ によって与えられた順序に適合することを保証する、当該述語 $n e x t (J_1, J_2, M)$ に対する制約とを含む。

第 8 行は、最初のジョブを除くそれぞれのジョブの先行ジョブを表す述語 $n e x t (J_2, J_1, M)$ と、述語 $n e x t (J_2, J_1, M)$ による選択が述語 $b e f o r e (J_2, J_1, M)$ によって与えられた順序に適合することを保証する、当該述語 $n e x t (J_2, J_1, M)$ に対する制約とを含む。 20

第 9 行は、述語 $f i r s t (J, M)$ による選択が述語 $b e f o r e (J_2, J_1, M)$ によって与えられた順序に適合することを保証する、当該述語 $f i r s t (J, M)$ に対する制約を含む。

第 10 行は、述語 $l a s t (J, M)$ による選択が述語 $b e f o r e (J_1, J_2, M)$ によって与えられた順序に適合することを保証する、当該述語 $l a s t (J, M)$ に対する制約を含む。

【 0 0 6 3 】

第 12 行～第 19 行は、差分ロジックによる定量的なモデル完了時間である。

【 0 0 6 4 】

第 12 行～第 16 行は、上述のスケジューリング問題をモデル化する。 30

第 17 行は、 $c_m a x$ を任意の完了時間の上限として定義する。本例においては、A S P ソルバは、可能な限り最小の値を有する整数変数をインスタンス化する。従って、ソルバが決定するあらゆる解集合において、 $c_m a x$ は、実際のメイクスパンである。

第 19 行は、付加的な冗長的なルールを定義する。本例においては、A S P ソルバは、検索スペース内の解を検索する。このルールは、A S P ソルバが検索スペースをさらにブルーニングするために役立つ。

【 0 0 6 5 】

第 21 行～第 23 行は、最適化問題をどのようにして解決するかを定義する。

【 0 0 6 6 】

第 21 行は、それぞれの機械のためのスパンを定義する。ここでは、 $i n t (T)$ は、整数の有界範囲を提供すると仮定される。 40

第 22 行は、機械が、その機械に対して定義されたスパンより遅くならず完了することを強制する。

第 23 行は、目的関数を定義する。

【 0 0 6 7 】

特に、いずれの機械も、各自のスパン c を、優先レベル c においてコスト関数に付与する。

【 0 0 6 8 】

コスト関数は、寄与する値を累積し、ソルバは、優先順位によって順序付けられたコス 50

トタブルを辞書式に比較することによって解集合を最小化する。

【 0 0 6 9 】

この例においては、任意の述語 `next` によって定義された順序でジョブが機械上に配置されることを表すために、差分ロジックが使用される。差分ロジックの使用は、基礎化のサイズを増大させる可能性のある、ブール `ASP` 制約における整数演算が回避されるという利点を有する。

【 0 0 7 0 】

厳密アルゴリズムは、辞書式メイクスパンの最適化と称され、以下においては、図 2 を参照しながら説明される。図 2 は、スケジューラ 1 0 2 によって実行される例示的なステップを概略的に示している。

【 0 0 7 1 】

スケジュールは、本例においては、最適化問題を反復的に解決することによって最適化問題の解として決定される。モデル `M` は、入力 2 0 2 として `ASP` ソルバ 2 0 4 に供給される。スケジュールは、`ASP` ファクト生成器 2 0 8 の出力 2 0 6 である。

【 0 0 7 2 】

`ASP` ソルバ 2 0 4 は、最適化問題に対する解 `solve (M)` を検索して、`M` に関する最適化問題を解決する空でない集合 `R` を出力し、又は、固定のリソース制限内において何も見つからなかった場合に、

【数 4】

$$R = \emptyset$$

を出力するように構成されている。コンフィギュレーション 2 1 0 は、 $1 \leq l \leq m$ の範囲内のパラメータ `l` をイテレータ 2 1 2 に供給するように構成されている。イテレータ 2 1 4 は、`ASP` ソルバ 2 0 4 が空集合

【数 5】

$$\emptyset$$

を出力するたびに、所定の範囲内において増分されるカウンタ `i` を `ASP` ファクト生成器 2 0 8 に供給する。イテレータ 2 1 4 は、第 1 の反復において $i = 1$ でカウンタ `i` を開始し、`ASP` ソルバが空集合

【数 6】

$$R = \emptyset$$

を出力するたびに、カウンタ `i` を 1 だけ増分するように構成されている。`ASP` ファクト生成器 2 0 8 は、任意の空でない集合 `R` に依存して制約 2 1 6 を決定するように構成されている。第 1 の反復 `i` の制約 2 1 6 は、次の反復 $i + 1$ のための入力である。`ASP` ソルバ 2 0 4 は、この制約 2 1 6 の下での次の反復 $i + 1$ において、`M` に関する最適化問題を解決するように構成されている。本例においては、前回の反復の制約がモデル `M` に追加される。`ASP` ファクト生成器 2 0 8 は、本例においては、最後の反復 $i \leq l$ の後にスケジュールを出力するように構成されている。

【 0 0 7 3 】

以下においては、辞書式メイクスパン最適化を決定するための厳密アルゴリズムについて説明する。

【 0 0 7 4 】

入力： $1 \leq l \leq m$ である `m` 個の機械と、パラメータ `l` とが含まれているモデル `M`

出力：パラメータ化された辞書式メイクスパンを有する `M` に対するスケジュール `R` (`c` _ 1 , . . . , `c` _ `m`)

【 0 0 7 5 】

`solve` (M) は、固定のリソース制限内において何も見つからなかった場合に、M に対する空でない解 R、又は、

【 数 7 】

$$R = \emptyset$$

を返す。

【 0 0 7 6 】

`bound` (i ° b) , ° ∈ {< ≤} 制約は、辞書式メイクスパン (c _ 1 , . . . , c _ m) のための c _ i ° b が、以下の通りであることを強制する。 10

【 表 2 】

```

1      (c_1, ..., c_l) ← (0, ..., 0)
2      R ← solve(M)
3      for i ← 1 to l do
4          S ← R
5          while S ≠ ∅ do
6              R ← S
7              c_i ← ith element of the lexicographical makespan of S
8              S ← solve(M ∪ bound(i < c_i))
9          M ← M [ bound(i ≤ c_i)
10     return R with lexicographical makespan (c_1, ..., c_l)
```

20

【 0 0 7 7 】

この厳密アルゴリズムにより、複数のソルバ呼び出しと、制約の増分的な追加とを使用して、最小の辞書式メイクスパンを有するスケジュールを計算することができる。

【 0 0 7 8 】

これは、最高の優先度の最初の階層的な降下を含む、多目的最小化を表す。対称性に起因して、解がないことを示すことは、通常、この問題のために解を見つけることよりもコストがかかる。即ち、このことにより、二分検索又は指数関数的に増加する検索ステップのように予期されるソルバ呼び出しが比較的少ない代替戦略は、さほど魅力的なものではなくなる。

30

【 0 0 7 9 】

本例においては、`clingo-dl` 及び上記のコーディングは、`solve` (M) を実施するための第 21 行～第 23 行の最適化ステートメントなしで使用される。複数回の実行にわたってソルバ状態の一部が維持されるマルチショット解決を支援する `clingo-dl` を使用することにより、計算リソースが節約される。特に、厳密アルゴリズムにおける `solve` (M) は、ASP ソルバを使用することに限定されるものではない。これに代えて、それぞれのシステムの入力言語においてモデル M に対する解を生成することができる任意の厳密法を代わりに使用することができる。

40

【 0 0 8 0 】

`bound` (i ≤ b) に関する制約は、例えば、ASP において表現される。

【 0 0 8 1 】

b 以下である辞書式メイクスパンの i 番目の成分は、少なくとも m - i + 1 個の機械が最大で b のスパンを有することを強制することと同等である。本例においては、後者は、m - i + 1 個の機械を非決定論的に選択して、これらの機械が b より遅くならず完了することを強制することにより、

50

【表 3】

$(m-i+1) \{ \text{sel}(M) : \text{machine}(M) \}.$
 $\&\text{diff} \{ \text{span}(M) - 0 \} \leq b :- \text{sel}(M).$

のようにエンコーディングされる。

【0082】

厳密アルゴリズムは、 $\text{solve}(M)$ のためのリソースが制限されていない場合に、最小の辞書式メイクスパンを有するスケジュールを返すことが保証されている。

【0083】

スケジューリングの例示的な第1の方法について、図3を参照しながら説明する。第1の方法は、ステップ302を含む。ステップ302においては、機械の集合K及びジョブの集合Jに関する最適化問題を解決するスケジュールが決定される。

【0084】

最適化問題は、ジョブの集合を機械の集合に割り当てるルール集合によって定義される。スケジュールは、ジョブの集合のうちのそれぞれのジョブを、機械の集合のうちの、そのジョブを処理することができる1つの機械にマッピングする。

【0085】

スケジュールが或る1つの機械に割り当てたジョブの、この機械による実行は、1つの機械スパン内に終了するようにスケジューリングされる。

【0086】

その後、ステップ304が実行される。ステップ304においては、少なくとも1つの機械スパンに関する制約が、その機械スパンに依存して決定される。

【0087】

その後、ステップ306が実行される。ステップ306においては、制約の下でジョブの集合及び機械の集合に関する最適化問題を解決するスケジュールが決定される。スケジュールが或る1つの機械に割り当てたジョブの、この機械による実行は、1つの機械スパン内に終了するようにスケジューリングされる。

【0088】

その後、ステップ308が実行される。ステップ308においては、反復のカウンタ i がパラメータ l を超えているかどうか判定される。反復のカウンタ i がパラメータ l を超えている場合には、ステップ310が実行される。そうでない場合には、カウンタが増加させられて、ステップ304が実行される。

【0089】

ステップ310においては、スケジュールが、最適化問題の解として出力される。

【0090】

実際には、最適化問題に対する解を探索するために費やされる時間を、適当な制限時間によって制限することができる。

【0091】

近似アルゴリズムにおいては、小さいメイクスパンを有するスケジュールを見つけるように構成されたオブティマイザ、特に厳密オブティマイザが使用される。このオブティマイザは、小さいメイクスパンを有するスケジュールを見つけることに優れている任意のプロシージャを使用することができる。このオブティマイザは、本例においては、以下において説明されるように、小さい辞書式メイクスパンを有するスケジュールを計算するために採用される。

【0092】

入力： $1 \leq l \leq m$ である m 個の機械と、パラメータ l とが含まれているモデル M

出力：パラメータ化された辞書式メイクスパンを有する M に対するスケジュール $S(c_1, \dots, c_l)$

$\text{opt}(M)$ は、固定のリソース制限内において見つかった M に対する最適な解を返す

。

【表 4】

```

1      (c_1, ..., c_l) ← (0, ..., 0)
2      S ← ∅;
3      for i ← 1 to l do
4          S ← opt(M)
5          c_i ← makespan of S
6          remove some machine k that completes at c_i
           and all jobs assigned to k from M
7      return S

```

10

【0093】

この近似アルゴリズムは、オブティマイザを使用して、オブティマイザによって提供された解の一部を反復的に計算及び改善する。このために、オブティマイザに対する複数のソルバ呼び出しが必要とされる。ソルバ呼び出しの数は、機械の数によって制限される。それぞれのソルバ呼び出しの後、問題インスタンスは、顕著に単純化される。従って、問題インスタンスをより簡単に解決することができる。

【0094】

20

オブティマイザは、前回のソルバ実行からのヒューリスティック値及び学習された節を再利用したマルチショット解決を使用することによって実装可能である。

【0095】

本例においては、ASPファクト生成器208は、第1の反復において、c_iで完了する全ての機械kをモデルMから除去し、かつ、kに割り当てられた全てのジョブをMから除去するように構成可能である。ASPソルバ204は、本例においては、そのように変更されたモデルMを解決するためにオブティマイザを呼び出すように構成されている。

【0096】

以下においては、スケジューリングの例示的な第2の方法について、図4を参照しながら説明する。第2の方法は、ステップ402を含む。ステップ402においては、機械の集合K及びジョブの集合Jに関する最適化問題を解決するスケジュールが決定される。

30

【0097】

その後、ステップ404が実行される。ステップ404においては、スケジュールのメイクスパンで終了するようにスケジューリングされた機械スパンを有する少なくとも1つの機械kが選択される。

【0098】

その後、ステップ406が実行される。ステップ406においては、機械の集合Kのうちの、機械kを含まない部分集合が形成される。

【0099】

その後、ステップ408が実行される。ステップ408においては、ジョブの集合Jのうちの、スケジュールに従って機械kに割り当てられたジョブjを含まない部分集合が形成される。

40

【0100】

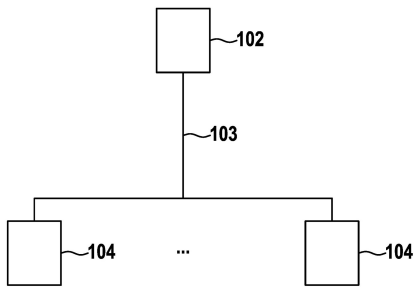
その後、ステップ410が実行される。ステップ410においては、ジョブの集合Jのうちの部分集合及び機械の集合Kのうちの部分集合に関する最適化問題を解決するスケジュールが決定される。

【0101】

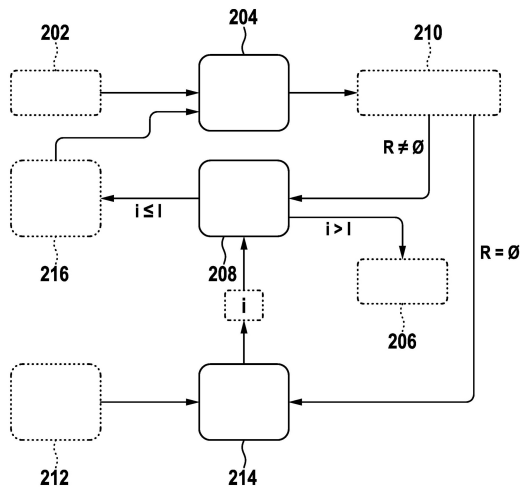
その後、ステップ412が実行される。ステップ412においては、反復のカウンタiがパラメータしを超えているかどうか判定される。反復のカウンタiがパラメータしを超えている場合には、ステップ414が実行される。そうでない場合には、カウンタが増

50

加させられて、ステップ 4 0 4 が実行される。
【 0 1 0 2 】
ステップ 4 1 4 においては、スケジュールが、最適化問題の解として出力される。
【 図 面 】
【 図 1 】



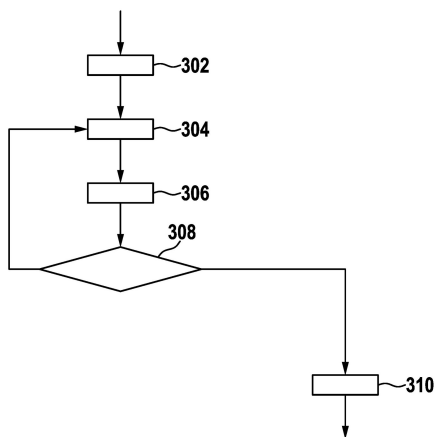
【 図 2 】



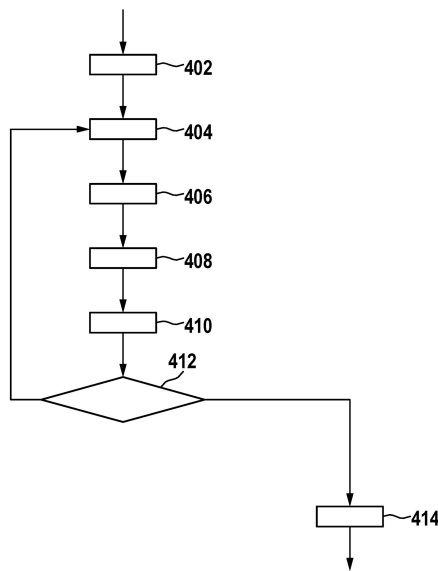
10

20

【 図 3 】



【 図 4 】



30

40

50

【外国語明細書】
2022167845000016.pdf

10

20

30

40

50

フロントページの続き

- 弁理士 永島 秀郎
(74)代理人 100162880
弁理士 上島 類
(72)発明者 ヨハネス エッチュ
オーストリア国 ウィーン ゾンマーガッセ 3 トップ 33
(72)発明者 ニスレット ムスリウ
オーストリア国 ウィーン マッツィンガーシュトラッセ 4アー／8／12
(72)発明者 ペーター スココヴスキー
オーストリア国 ウィーン ディートリヒガッセ 22／37
(72)発明者 トーマス アイター
オーストリア国 ウィーン グローセ ノイガッセ 5 トップ 8
(72)発明者 トビアス ガイビンガー
オーストリア国 ウィーン フォルシュングスベライヒ フュア ダーテンバンケン ウント アーテ
ィフィシャル インスティテュート フュア ロジック アンド コンピューテイション 192-02
(72)発明者 ダリア ステパノワ
ドイツ連邦共和国 レオンベルク ヒンデンブルクシュトラッセ 68