

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5759333号
(P5759333)

(45) 発行日 平成27年8月5日(2015.8.5)

(24) 登録日 平成27年6月12日(2015.6.12)

(51) Int.Cl.

G05B 19/418 (2006.01)

F 1

G 05 B 19/418

Z

請求項の数 5 (全 23 頁)

(21) 出願番号 特願2011-227451 (P2011-227451)
 (22) 出願日 平成23年10月14日 (2011.10.14)
 (65) 公開番号 特開2012-94136 (P2012-94136A)
 (43) 公開日 平成24年5月17日 (2012.5.17)
 審査請求日 平成26年10月9日 (2014.10.9)
 (31) 優先権主張番号 12/911,766
 (32) 優先日 平成22年10月26日 (2010.10.26)
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 596170170
 ゼロックス コーポレイション
 XEROX CORPORATION
 アメリカ合衆国、コネチカット州 068
 56、ノーウォーク、ピーオーボックス
 4505、グローバー・アヴェニュー 4
 5
 (73) 特許権者 502096543
 パロ・アルト・リサーチ・センター・イン
 コーポレーテッド
 Palo Alto Research
 Center Incorporated
 アメリカ合衆国、カリフォルニア州 94
 304、パロ・アルト、コヨーテ・ヒル・
 ロード 3333

最終頁に続く

(54) 【発明の名称】セル生産方式における適応型ロットサイズ決定のためのシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

複数のジョブおよび複数のセルのリストを作成することを含み、前記複数のジョブは前記複数のセルの少なくとも一部により生産される予定であり、前記セルのそれぞれは少なくとも 1 つの生産装置を含み、

前記ジョブを自動分割することを含み、

前記自動分割することが、

前記ジョブをより小さいサイズのロットに分割することと、

前記セルの種々のセルへの前記より小さいサイズのロットを一度に一つのロットずつ割り振ることであって、前記割り振ることは、最早完了経路アルゴリズムを用いて前記セルに前記ジョブの前記より小さいサイズのロットを割り当てるなどを含み、前記最早完了経路アルゴリズムは、最少のセル間跳躍回数を有する前記複数のジョブの各特定のジョブのための実行可能な経路のセットを列挙し、対応する経路によって識別されるそれらのセルを用いることだけによってベストな割り振りを見つけ、製造処理量と待ち時間との少なくとも 1 つを考慮して前記特定のジョブを終了させる前記経路及び対応割り振りを選択することを含む、前記割り振ることと、

すべての前記より小さいサイズのロットが割り振られると、より大きなロットを形成するために、前記より小さいサイズのロットのうちのどれを、割り振られた前記より小さいサイズのロットの他のロットと組み合わせができるかを判断することと、

同じジョブの連続するロットとして組み合わせ可能と判断された前記より小さいサイズ

のロットをいっしょに統合することと、を含む

セル生産方式環境内の複数のセル全体にわたって作業負荷のバランスをとる方法であつて、

前記方法は、少なくとも電子プロセッサの利用により実行される
方法。

【請求項 2】

前記自動分割することが、最適ロットサイズを決定するための固定した式を用いずに機能する、請求項 1 に記載の方法。

【請求項 3】

先行して割り振られたロットおよび新しく割り振られたロットが連続しかつ結合されるときには、前記新しく割り振られたロットに対する準備時間を準備時間から差し引く、請求項 1 に記載の方法。 10

【請求項 4】

前記ロットの 2 つが連続して割り振られており、かつ同じジョブに属している場合、およびその場合のみ、2 つのロットは連続している、請求項 3 に記載の方法。

【請求項 5】

前記 2 つの連続したロットを統合するとき、

前記統合されたロットのサイズは統合中の前記 2 つのロットのサイズの合計であり、

前記統合されたロットの開始時刻は、より早く開始するように割り振られている前記ロットの開始時刻であり、 20

前記統合されたロットの終了時刻は、より遅く終了するように割り振られているロットの終了時刻である、請求項 4 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、セル生産方式概念を使用する改良型リーン文書生産（LDP）の新規で改善されたシステムおよび方法に関する。特に、セル生産方式環境内の複数のセル全体にわたって作業負荷のバランスをとるシステムおよび方法に関する。

【背景技術】

【0002】

従来の印刷所は、印刷ジョブ、印刷ジョブミックス、およびシステムを通じる印刷ジョブの総量から機能的に独立した形で構成されている。 30

【0003】

したがって、通常、印刷所は、すべてのプリンタを「プリンタ専用」領域内に集めて一団にする、またはすべてのインサーダを「インサーダ専用」領域内に集めて一団にするなどして印刷所の設備を配置するであろう。これは、プリンタの近くにあるのは他のプリンタだけであり、インサーダの近くにあるのは他のインサーダだけであり、単一の生産ジョブを仕上げるために密接に関与する必要のある異なる種類の機械間の分離を事実上引き起こすことを意味する。結果として、従来の印刷所の仕掛品（WIP）が非常に高額である可能性があると同時に、平均ジョブ納期が長くなる。LDP 解決方法は文書生産プロセス内のこの「摩擦」を認識しており、これらの効率の悪さの大部分を取り除く「セル」と呼ばれるプロセスに適した小型店舗を作るように印刷所レイアウトを再編する。 40

【発明の概要】

【発明が解決しようとする課題】

【0004】

LDP 解決方法は文書生産を「リーン」プロセスにする数々の革新を含んでいるが、この技術の中心には依然としてセル生産方式の概念があり、これは多くのやりがいのある技術的課題を作り出しており、これらの技術的課題のうち最も注目に値するものは、セルの概念の周囲に構成されている生産環境においてどのようにして効率的にジョブを割り振るかということである。新しいスケジューリングエンティティとしてのセルの出現以外にも 50

、 L D P システム内で使用されるスケジューラに対して付加的な課題を提起する他の多くの問題がある。

【 0 0 0 5 】

1つの問題は大きくテールを引くジョブサイズ分布であり、この分布は印刷ジョブ（特に、大きな印刷所で見られる印刷ジョブ）のサイズが大幅にばらつくという事実を示しており、有限分散を有するいかなる「教科書的な」分布（正規分布または指數分布など）によっても、もはやそれらの印刷ジョブ分布を十分に特徴付けることはできない。多くのスケジューリングアルゴリズムおよびシステムがまったく正反対を前提としている（すなわち、大きくテールを引かない分布を取り扱うだけである）ことを考えれば、この課題に対処するには新しいスケジューラが必要である。

10

【 0 0 0 6 】

既存のスケジューラは店舗レベルのスケジューラによりジョブをセル（または必要に応じてセルのシーケンス）に最初に割り当てる 2 層構造のスケジューリング方法の形を取り、各セルはそれ自身のジョブキューを有しているため、ひとたびあるセルにジョブが割り当てられると、セルの中のジョブステップが終わるまで、そのジョブは恒久的にそのセルに結び付けられる。この戦略は、店舗レベルのスケジューラが各セルの作業負荷をいつでも正確に予測でき、かつ各セルの利用レベルをできるだけ高く（または均等に分布した状態に）保持するようにジョブからセルへの写像を計算できる場合にはうまくいく。しかしながら、現在のスケジューラを用いる場合、この戦略を実現するのは困難な可能性がある状況がある。1つの理由は、店舗レベルのスケジューラが、その決定を行う間に各セルの中の詳細なスケジューリング制約条件を考慮しないことである。ジョブの具体的な到着日および納入日、シーケンス依存の設定コスト、および単一セルに割り振られた類似の納入日のジョブの個数などの要因は、店舗レベルのスケジューラには扱われない。扱われないこれらすべての要因は、常にセルの作業負荷に寄与する可能性がある（および、通常、寄与する）ことは明らかである。したがって、ジョブからセルへのこのような任意の固定した写像は、いくつかのセルを過負荷にさせる一方で、他のセルはときどきジョブのない状態のままにしておく傾向がある。

20

【 0 0 0 7 】

既存の L D P システムを改善することが有利であろう他の領域は「バッチ分割」の領域にあり、これは L D P システムにおいて大きなジョブを取り扱うためのスループット改善戦略である。バッチ分割は、長いジョブを「バッチ」と呼ばれる多数の、より小さいユニットに分ける。また、ときには、バッチをサブジョブと呼んでもよい。狙いは、長いジョブの一部分がシステムの中を通って流れる準備ができるとすぐにダウンストリーム待ち時間を無くすことである。

30

【 0 0 0 8 】

既存の L D P 解決方法では、ワークフローの種類（すなわち、シリアルまたはアセンブリ）に応じて 2 つの異なる方法でロットサイズを計算する。したがって、これらの状況でのバッチ分割の効率を改善することは有利であろう。本願は、これらおよび他の問題に対処する。

【課題を解決するための手段】

40

【 0 0 0 9 】

本願は、セル生産方式概念を使用する改良型リーン文書生産（ L D P ）の新規で改善されたシステムおよび方法を提示する。 L D P プロセスは、作業工程表を生成するために自動分割アルゴリズムおよび／または最早完了経路アルゴリズムを内蔵している。

【図面の簡単な説明】

【 0 0 1 0 】

【図 1】セルおよびセルに対応する装置を組み合わせた印刷ワークフローシステムを示すブロック図である。

【図 2】図 1 の印刷ワークフローシステム内で使用されるソフトウェアコンポーネントを含むブロック図である。

50

【図3】図1の印刷ワークフローシステムを使用するサーバのブロック図である。

【図4】文書処理ジョブのサブジョブへの分割に適合するために印刷ワークフローシステムをどのようにして使用するかを示すブロック図である。

【図5】対応する一連のキューを経由してジョブをセルにルーティングするための方法を示すブロック図である。

【図6】n個の作業を含むシリアルワークフローを示す図である。

【図7】6個の作業を含むアセンブリワークフローを示す図である。

【図8】ロットサイズに対するクリティカルパス変化を示すグラフを示す図である。

【図9】ロットを統合するためのフローチャートである。

【図10】統合が行われると何が起こるかについてさらに詳述する図である。 10

【図11】最早完了スケジューリングシステムおよび方法である。

【図12】経路列挙フローチャートである。

【発明を実施するための形態】

【0011】

開示される実施形態は、文書工場（印刷所などの）内の文書処理ジョブの生産を調整するためのスケジューラとして働くリーン生産プロセスサーバ（LPPS）または適切なソフトウェアを実行する他の電子プロセッサの使用を想定している。サーバは文書処理ジョブを制御するためにリーン生産技術を活用し、UNIXおよびWindows（「UNIX」はオープン・ソース・グループの登録商標であり、他方、「Windows」はマイクロソフト社の登録商標である）に基づくプラットフォームを含む多くの異なるプラットフォーム上で実行できるが、これらのUNIXおよびWindowsに基づくプラットフォームに限らない。サーバはワークフロー優先順位を決定し、それに応じてワークフローを管理する。間もなく開示される実施形態は、また、他のさまざまなオペレーティングシステムを実行するプラットフォームとともに実施してもよいことを当業者は理解するであろう。さらに、サーバは専用コンピュータシステム上で実行される必要はなく、むしろプリンタ、コピー機などの他のさまざまな電子装置上で実行されてもよい。文書処理ジョブのワークフロー優先順位は、さまざまなジョブ処理ユニットを観察することにより決定できる。 20

【0012】

本明細書に開示される少なくとも1つの説明に役立つ実施形態は、文書工場が自立したセルに区分されていると見なす。各セルは、少なくとも1種類の文書処理ジョブを完了するのに十分な文書工場内の資源の論理的分類である。したがって、第1のセルはプリンタおよびバインダを含んでいてもよく、他方、第2のセルはコピー機および照合機を含んでいてもよい。LPPSは、このようなセルの中で文書処理ジョブを効果的に分配するに関与する。 30

【0013】

図1を参照すると、印刷所（すなわち、文書工場）内の印刷ワークフローシステム100を示している。ソフトウェアを作動させるように構成された電子プロセッサなどの少なくとも1つのプロセッサを含む印刷ワークフローシステム100は、多数のセル102～106を制御するとともに、通信リンク108を介してセル102～106へ情報を送信し、セル102～106から情報を受信する。セル102～106は、所与の生産物タイプの文書処理ジョブを完了するのを支援するための少なくとも1つの装置を含んでいる。例えば、印刷装置110が600dpi白黒プリンタである可能性があり、他方、印刷装置112が1200dpiカラープリンタである可能性がある。もちろん、これらはあくまで例に過ぎず、他の多くの処理装置がセルの中にあってもよい。 40

【0014】

図2を参照すると、印刷ワークフローシステム100で使用してもよいモジュールの種類のうちのいくつかを示している。これらのモジュールは、選択された文書処理ジョブに対するワークフローを決定するワークフロー写像モジュール114を含んでいる。特に、ワークフローモジュールは、文書処理ジョブを完了するのに必要な運用ステップと、その 50

中でこれらの運用ステップを実行すべきシーケンスと、を特定する。文書処理ジョブをバッチ（本明細書ではロットとも呼ばれる）またはサブジョブに分割するために、およびバッチをセルに送って完了するためにジョブ分解モジュール 116 を含んでいる。セル内の装置により処理される予定の少なくとも 1 つのバッチを受け取るために所与のセルに生産物セルコントローラ（PCC）118 を提供してもよい。最後に、セルにより処理される予定のバッチを割り当てるためにセル割当モジュール 120 を提供する。

【0015】

一般に、印刷ジョブを受け取り、ワークフロー写像モジュール 114 により印刷ジョブに対するワークフローを開発する。ジョブ分解モジュール 116 はジョブをバッチに分割してもよく、その後、セル割当モジュール 120 により完了するために、バッチまたはジョブをセル（例えば、102～106）に割り当てる。割り当てられたセルの生産物セルコントローラ 118 にバッチを送ってもよく、この生産物セルコントローラ 118 で各バッチをさらに細分してもよい。10

【0016】

図 3 を参照すると、印刷ワークフローシステム 100 での使用に適したサーバ 300（すなわち、LPPS）の一実施例を示している。サーバ 300 は、プロセッサ 302、記憶装置 304、通信リンク 306、および入力モジュール 308 を含んでいる。通信リンク 306 を介してさまざまな装置から入力を受信するために入力モジュール 308 を使用する。入力モジュール 308 はサーバ 300 を代表して文書処理ジョブを受信する。プロセッサ 302 はサーバ 300 上でプログラム命令を実行して、文書処理ジョブを管理する。20 サーバ 300 は記憶装置 304 内に命令を保存する。例えば、モジュール 114、116、および 118（図 2 の）のデータ、ならびに他のデータを記憶装置 304 内に保存してもよい。サーバ 300 上で、またはセルに特有の別のサーバ上でモジュール 120 のデータを実行してもよい。

【0017】

図 4 を参照すると、生産システムのセル（例えば、図 1 の 104 および 106 など）と情報をやりとりする印刷ワークフローシステム（図 1 の印刷ワークフローシステム 100 のような）のサーバ 300 を含む配置を示している。セルに対する生産物セルコントローラ 118 は、セル 104、106 によりさらに処理される予定の、サーバ 300 からのバッチ 400、402 を受け取る。サーバ 300 は、異なる生産物タイプを生産するために印刷所内の各セルの最大生産量および機能を記憶装置 304 内に保存している。例えば、ネットワーク内のセル 104 は 3 つの異なるタイプの文書を生産してもよく、セル 106 は 2 つのタイプの文書を生産してもよい。2 つの異なるセルが類似の文書タイプ、または同じ文書タイプでさえも生産できることが十分起こり得ることに言及する（すなわち、文書タイプは文書を完全に仕上げるための処理ステップのシーケンスにより一意的に特徴付けられる）。サーバ 300 はこの情報を保存して、どのセルが文書ジョブを処理する機能を有しているかを判断する。また、各セルの最大生産量を保存して、セルが生産できる特定の生産物タイプの生産量を判断する。30

【0018】

上述のように、さまざまな自立したセルに送って処理するためにジョブ分解モジュール 116 は文書処理ジョブをバッチに分割してもよい。ネットワーク内のセルが自立している限りは、セルはジョブを完全に処理できる。図 4 に示す実施例では、文書処理ジョブはバッチ 400 および 402 に分割され、それぞれセル 104 および 106 に送られる。生産物セルコントローラ 118 および 118' は、バッチ 400、402 をそれぞれ装置集合 404A～404C および装置集合 406A～406C に送って処理する。40

【0019】

図 5 を参照すると、本実施形態とともに使用するためのジョブルーティングシステム 500 の簡略版を示している。ひとそろいの保存されたジョブ 504 と通信するルータ 502 が、セル割当モジュール 120（図 2）および LPPS 300（図 3）に関連するジョブ管理機能の少なくとも一部を含んでいてもよい。ルータは一連の印刷処理関連キー 5050

06(506-1、506-2、...506-nとして指定される)とさらに通信し、これらのキューはセル102~106にそれぞれ動作可能に関連している。

【0020】

上述のように、この領域での特定のスループット改善戦略はロットサイズ決定（通称バッチ分割）であり、このロットサイズ決定は、大きいジョブを「ロット」と呼ばれる（またある時には、本明細書でバッチまたはサブジョブと呼ばれる）多数の、より小さいユニットに分ける。狙いは、長いジョブの一部分がさらに処理する準備ができるとすぐにダウンストリーム待ち時間を無くすことである。既存のLDP解決方法では、ワークフローの種類（シリアルまたはアセンブリ）に応じて2つの異なる方法でロットサイズを計算する。

10

【0021】

ワークフローの概念に注意して、図6は、対応するステーション（ステーション1～ステーションn）で連続して実行されなければならないn個の作業（すなわち、作業1～作業n）を含むシリアルワークフロー600を示しており、各ステーションはジョブまたはジョブの一部分を完了するのに必要な上述の資源（例えば、プリンタ、バインダ、コピー機、照合機など）を含んでいる。各作業は、1生産品目あたり r_i タイムユニットの処理時間有している。処理の間に1つのロットから次のロットへ切り換えるためには、フロ-600は1生産品目あたり s_i ユニットの準備時間を必要とする。

【0022】

ジョブの全体のサイズはN項目と定義され、ロットサイズはb項目である。次の式で全体のマイクスパンを与える。

20

【数1】

$$T(b) = s_1 + (r_1 + r_2 + \dots + r_n) b + (N/b - 1) \max \{ s_1 + r_1 b, s_2 + r_2 b, \dots, s_n + r_n b \} \quad (1)$$

Tがロットサイズbの関数であるため、最適ロットサイズはTを最小にするロットサイズである。この最小値は次のように見つけることができる。

1. Nを正確に分割する整数bの集合を計算する。
2. この集合内のすべてのbに対するT(b)を評価し、これらの値をベクトルに保存する。
3. このベクトルの最小成分を選択する。この成分に対応するbが最適ロットサイズである。

30

【0023】

アセンブリワークフローでは、最適ロットサイズの計算はさらに複雑である。図7は、6個の作業（作業1～作業6）を含むアセンブリワークフロー700により処理されているジョブの実施例を示しており、これらの6個の作業の一部は並行して同時に実行され（例えば、作業1および2など）、これらの6個の作業の一部は連続的に実行される（例えば、作業5および6など）。6個の作業を示しているが、この6個という個数はあくまで例として提供したに過ぎず、本概念は他の個数のステーションおよび作業を有するワークフローにも同様に適用できることを理解すべきである。

40

【0024】

シリアルケースとは異なり、全マイクスパンが、ロットサイズごとに変化する可能性があるクリティカルパス（すなわち、最大生産時間を有する経路）に依存するため、アセンブリワークフローに対する最適ロットサイズを計算するための閉形式の式がない。このことを理解するために、準備時間 s_1 、 s_2 、およびユニット処理時間 r_1 、 r_2 をそれぞれ有する2つのステーションがあると仮定する。サイズbを有するロットに対するそれらの処理時間は、それぞれ $s_1 + b r_1$ および $s_2 + b r_2$ である。図8の線図800に示すように、 $b \leq b^*$ のとき、 $s_1 + b r_1 \leq s_2 + b r_2$ であり、これはステーション2がクリティカルパス上にあることを意味しており、 $b > b^*$ のとき、 $s_1 + b r_1 > s_2 + b r_2$ であり、これはステーション2ではなく、ステーション1がクリティカルパ

50

ス上にあることを意味している。

【0025】

現在の LDP スケジューラは、クリティカルパスを動的に決定するために下記の戦略を使用している。仮に、 $t_i = s_i + b r_i$ 、 $i \in \{1, 2, \dots, m\}$ がステーション i 上のサイズ b を有するロットの全処理時間を示すとする。さらに、

$$\bar{t}_i = t_i + m \max_{j \in P_i} t_j$$

10

であるとし、ここで、 P_i はステーション i の上流のステーションの集合である。そのとき、

$$\bar{t}_i, i \in \{1, 2, \dots, m\}$$

は、ステーション i を通じてロットを処理するための経過時間である。したがって、ライン内の最後のステーションを通じてロット（またはバッチ）を処理する経過時間である

$$\bar{t}_m$$

20

は、クリティカルパスにより決定されるサイズ b を有するロット（またはバッチ）の全納期である。さらに、各ステーションは、これまでのところ最大処理時間を与える上流のプランチの経過を追い、この上流のプランチがクリティカルパスを決定する。その後、 b のあらゆる可能な値に対して、既存の LDP スケジューラは、上述のプロセスを通して特定されるクリティカルパス内のステーションを用いて上述の式(1)により決定される $T(b)$ を計算する。そのとき、最適ロットサイズは $T(b)$ を最小にするロットサイズである。 $T(b)$ はロットサイズ b の変化がクリティカルパスの変化を引き起こすときに再評価される必要があるだけであるため、 b が変化するたびに $T(b)$ を計算する必要はないことに留意されたい。

【0026】

30

上述のように、それを用いて既存の LDP スケジューラ内の最適ロットサイズを決定するためのプロセスは、特に、最も一般的なタイプであるアセンブリワークフローに対して、かなり複雑である。しかしながら、さらに重要なことは、実際の最適ロットサイズは上述のプロセスで計算される値から大きく異なる可能性があることである。これは、第 1 のロットを処理するまでは、これまでに使用されたすべてのステーションは使用されていない状態でなければならない、という暗黙の前提に起因している。残念ながら、これは、店舗がほとんど使用されていないときに提示されるほんのわずかなジョブに対して正しいだけである。忙しい店舗では、後で提示されたジョブは、まだ依然として処理中の先に提示されたジョブと競争しなければならない。最悪の場合、すべての機械が既に完全に占有されているときには、ロットサイズ決定がマイクスパンの低減に少しもつながらない可能性がある。それどころか、複数の準備時間が同じジョブに属する異なるロットに対して必要となるため、ロットサイズ決定がマイクスパンを増加させる可能性がある。

【0027】

40

現在のロットサイズ決定式の他の制限は、单一ジョブステップを処理するのに複数の機械が利用できるケースを取り扱うのに、処理速度の寄せ集めに依存していることである。LDP では、所望のスループットを実現するために複数のプリンタおよび / またはインサータを備えた単一のセルを有することが一般的である。これは、各プロセス段階 i は単一の準備時間 s_i および処理速度 r_i だけを有することができる、という式(1)の前提に違反している。この問題を解決するために、既存の LDP スケジューラは、同じ機能（または機能シーケンス）を実行できる同じセル内のすべての機械の処理速度を合計する。こ

50

これは複数の並列機械を有するセルに対して同じ式が引き続き適用できることを確保するが、また同様に、これは、複数の、より遅い機械は多くの状況で単一の高速度の機械と同じではないという事実を無視している。

【0028】

任意のこのような静的に計算されたロットサイズは現実の印刷所ほど動的なスケジューリング環境には適合できないため、上述の制限がない、後述のシステムおよび方法は、「最適」ロットサイズを決定するのに、いかなる固定した式も使用しない。その代わり、図9のフローチャート900に示すように、本システムおよび方法は大きなジョブを多数の小ロット（バッチまたはサブジョブ）に分割し、一度に一つずつ、それらを割り振る。すべてのロットが割り振られると、統合段階を使用して、より大きなロットを形成するために、どのロットを他のロットと組み合わせができるかを判断する。したがって、最終的なスケジュールは、あまりに多くの小さいロットを有することに苦しむことはない。実際に、ジョブを単一のロットよりも多いロットにさらに分割すべきではない（まれな）事象では、本システムおよび方法は最適ロットサイズを見つけるであろうということを示すことができる。図9は上記を達成するためのステップ（さらに詳述されるであろう）を規定する高レベルフロー900である。したがって、プロセスフロー900は下記の通りである。

1. ジョブを十分多くのロット（またはシステムで事前に定義されてもよい最大許容ロット数）に分割する（902）。

2. すべてのロットに、まだ割り振られていないとの印を付ける（904）。

3. まだ割り振られていないロット数 > 0 の間はループに入り（906）、ループプロセスは下記の通りである。

a. 次のロットを割り振る（908）。

b. 新しい／現在のロットに、割り振り済みとの印を付ける（910）。

c. 同じ機械上の先行するロットおよび新しい（または次の）ロットが同じジョブに属しており、かつ現在のロットと連続して割り振られているかどうかを判断する（「いいえ」のとき、プロセスはステップ906に戻る）。「はい」のとき、プロセスは続く（912）。

d. 新しく割り振られたロットの準備時間を差し引く（914）。

4. ループが終わると（すなわち、ステップ906が「いいえ」であるとき）、同じ機械上に割り振られた連続したロットを統合する（916）。

【0029】

少なくとも1つの実施形態では、「十分大きな」の使用は、自然に課された制約により決定されたロットサイズを意味すると理解されることが分かる。また、2つのロットが連続して割り振られており、かつ同じジョブに属している場合、およびその場合のみ、2つのロットは連続している。

【0030】

図10のフローチャート1000で示すように、2つの連続したロットを統合するとき、統合されたロットを定義するために下記のステップを実行する。

1. 統合されたロットのサイズは統合中の2つのロットのサイズの合計である（1002）。

2. 統合されたロットの開始時刻は、より早く開始するよう割り振られているロットの開始時刻である（1004）。

3. 統合されたロットの終了時刻は、より遅く終了するよう割り振られているロットの終了時刻である（1006）。

【0031】

もちろん、統合されたロットは、他の統合済みロット、またはまだ統合されていないロットと、さらに統合できる。すべての*i* {1, 2, ..., m} に対して、機械*i*上に割り振られている連続したロットを見つけることができない場合、およびその場合のみ、スケジュールを極限まで統合する。このアルゴリズムは極限まで統合されたスケジュールだ

10

20

30

40

50

けを戻すため、このことは、極限まで統合されたスケジュールはジョブを完了するために必要な最少のロットしか含んでいないことから、アルゴリズムに必要以上に多くのロットを作らせない。例えば、1台を除き、すべての機械が稼働中の場合には、アルゴリズムは稼働していない同じ機械上にすべてのロットを連続して割り振ることを統合段階において発見するため、それらのロットのすべてを單一ロットに統合することで、複数の準備の代わりに1回の準備だけが必要になる。このようなケースは特に一般的というわけではないが、ときどき起こる。單一ロットスケジュールは極限まで統合されたスケジュールのうちの、すべてのロットを統合できる单なる特別なケースであるため、新しいアルゴリズムはその特別なケースを難なく取り扱う。記載のアルゴリズムは、セルの変化する作業負荷に合わせてロットサイズを自動的に変化させるため、自動分割アルゴリズムと呼んでもよい。

10

【0032】

複数のセルを関与させる必要があるワークフローを有するジョブに対しては、上述の自動分割アルゴリズムは、セル境界を超えたスケジューリングができるスケジューラ内で使用される。したがって、本明細書ではLDPにおいてセル間のスケジューリングを行うための計算効率の良い方法についても同様に説明している。また、しかしながら、本教示はセル生産方式の他の実施形態にも適用できることを理解すべきである。

【0033】

効率的な複数セルスケジューリングの際立った態様は、ジョブを仕上げるために協働しなければならない異なるセルを接続するための「最短経路」を特定することである。このような経路を見つけることは困難だがやりがいがある可能性があり、その理由は、関与するセルの個数を最少にすることだけが目的ではなく、これらのセルで費やされる時間を最小にすることの方がさらに重要であるためである。この目的を達成するために、本開示は、最も効率的な経路を特定するための最早完了スケジューリング戦略1100(図11)を開発した。アルゴリズムは下記のように働く。

20

- 最初に、アルゴリズムは最少回数のセル間跳躍を有するすべての経路を列挙する(1102)。

- その後、この集合内の各経路に対して、アルゴリズムは、経路により特定されるそれらのセルを使用することだけにより最良スケジュールを見つけようと試みる(1104)。

30

- その後、アルゴリズムは経路を選択し、ジョブを最も早く終える対応するスケジュールを選択する(1106)。

【0034】

上記について詳述すると、経路列挙アルゴリズムについて最初に説明するが、このアルゴリズムは最少回数のセル間跳躍を要求するジョブを完了するための経路の集合を戻すため、本明細書ではこのアルゴリズムを最少跳躍経路列挙アルゴリズムと呼ぶ。この議論では、「steps」はジョブのステップの総数であり、「h」はジョブが要求するセル間跳躍回数である。hはワークフローに含まれるセル数から1を引いた数値以上であるが、しかし、单一セルを2回以上訪れ、これらの訪問の間に、途中でさらに他のセルを訪れる(例えは、最初にセルAを訪れ、その後セルBを訪れ、その後再びセルAを訪れる)場合、hは、さらに大きくなる可能性があることに注目すること。訪問されたセルの集合の基數ではなく、セル間跳躍回数を使用する理由は、セル境界を超えた仕掛品(WIP)転送のようなセル間ワークフロー連携のコストを、より正確に明らかにするためである。図11の計数器ステップ1102および1104に対する最少跳躍経路列挙アルゴリズム1200のステップを図12に示している。この配置では、システムは下記のように働くであろう。

40

- h_{max} (セル間跳躍の仮想上限)を0に設定する(1202)。

- $h_{max} < steps$ の間はループに入る(1206)。

- $R_{min-hop}$ が、ジョブを完了できる h_{max} 以下のセル間跳躍を有するすべての経路の集合であるとする(1206)。

50

- b . $R_{min-hop} < >$ のとき、ループを出る(1208)。
- c . $R_{min-hop} < >$ ではないとき、 $h_{max} = h_{max} + 1$ (1210)。
- 3 . $R_{min-hop} < >$ のとき、最少跳躍経路の集合として $R_{min-hop}$ を戻す(1212)。
- 4 . $R_{min-hop} < >$ ではないとき、エラーを報告する(すなわち、現在のジョブは店舗内には無い1つ以上の機能を要求する)(1214)。

【0035】

上述の列挙アルゴリズム1200は、反復拡大戦略を使用して経路列挙基準を徐々に緩和する。セル間跳躍の最大回数は $Steps$ (ジョブステップの総数)から1を引いた数値に上方から境界されているため、ジョブが要求するすべての機能が店舗内のどこか(必ずしも単一セル内とは限らない)に存在しているとき、アルゴリズムは少なくとも1つの経路を見つけるように設計されている。一方、アルゴリズムは、より少ないセル間跳躍を有する経路を最初に試してみるため、少なくとも1つの実行可能な経路が存在しているとき、アルゴリズムはまた、 $R_{min-hop}$ 、すなわち、最少跳躍経路の集合を見つけるように設計されている。10

【0036】

最少跳躍経路の集合を見つけると、セル間スケジューリングアルゴリズムは、それぞれの $r_{R_{min-hop}}$ に対する可能な最良のスケジュールを見つけようと試みて、ジョブを最も早く完了する経路 r^* を選択する(すなわち、図11のステップ1106)。したがって、このアルゴリズムを最早完了経路スケジューリングと呼んでもよい。20

【0037】

既存の、式に基づく静的なアルゴリズムと比較して、この実施形態の自動分割アルゴリズムは完全に動的であり、リアルタイムのセル作業負荷に適応できる。特に、自動分割アルゴリズムは、

- 1 . 静的な式を使用せず、
- 2 . シリアルワークフローとアセンブリワークフローの両方に適用でき(すなわち、異なる種類のワークフローに対して別々に取り扱う必要がなく)、これによりコードの簡略化をもたらし、
- 3 . 最初は使用されていない機械についての前提を要求せず、
- 4 . 不十分なスケジューリング品質につながる可能性がある擬似的な処理速度を生成する速度の寄せ集め(個々の機械の処理速度の合計として)の必要性がない。30

【0038】

実験では、店舗A(中型店)、店舗B(中型店)、および店舗C(大型店)を含む多くの異なる大きさの印刷所について、記載の自動分割アルゴリズムの性能を評価した。店舗A(中型店)では、納期遅れのジョブ数(637から560に減少)、平均遅れ(773から738に減少)、および平均納期(934から899に短縮)を含むすべての主要なスケジューリング評価指標から見て、自動分割は静的方法よりも良いことが分かった。店舗B(中型店)では、自動分割は納期遅れのジョブ数を1から0に、平均遅れを-137から-150に(小さいほど良い)、および平均納期を20.46から7.62に減少させた。大型店舗である店舗C(大型店)では、両方の方法とともに、納期遅れのないスケジュールを見つけることができたが、自動分割は平均納期を1.79から0.84に、および平均処理時間を1.07から0.61に短縮する。40

【0039】

ここで際立った革新は動的ロットサイズ決定であるため、異なる作業負荷と、納入日などのさまざまなスケジューリング制約条件との下でロットサイズが変化するかどうかを観察することにより、静的ロットサイズ決定との違いを見つけることができる。静的方法では、このような変化はロットサイズにいかなる実質的变化も引き起こさないであろうが、印刷所の生産性はさまざまな条件の下で変動するであろう。しかしながら、自動分割を用いる動的ロットサイズ決定では、これらのすべての要因がロットサイズに影響を与える可能性があり、通常、影響を与えることになり、このことはスケジューラのソースコード50

を必要とせずに観察できる。

【0040】

最早完了経路スケジューリングでは、類似の方法を使用して、動的径路選択アルゴリズムが使用されているかどうかを判断でき、その理由は、最早完了経路スケジューリングでは、どの経路が「最短なのか」が、すべてのセルの作業負荷と、他のスケジューリング制約条件とに依存するためであるが、他方、先行する静的方法は、セルがどんなに忙しくても、常に同じ経路を選択する。この場合も先と同様に、このスケジューリング挙動を見つけるのにソースコードにアクセスする必要はない。

【0041】

記載の経路列挙アルゴリズムの変形は、すべての最少跳躍経路を見つけた後も、実行可能な経路を列挙し続けることであり、その理由は、ジョブを完了するための全体の処理時間から見ると、最少のセル間跳躍回数を有する経路が必ずしも「最短」であるとは限らないためである。さらに、経路列挙アルゴリズムは、本明細書に示すように、セルの近所については考慮しない（例えば、セルAはセルBとは近いが、セルCとは離れている）。経路列挙アルゴリズムの他の実施形態では、特定の経路が同じワークフロー内の2つの離れた（または隣接する）セルを接続する場合には、その経路に対して不利益（または恩恵）を付加する。

10

【0042】

本願は、少量の計算の複雑性を有するが、それにもかかわらず、他の品質評価指標の中で納期遅れのジョブ数と、平均納期の両方を減少させるのに有効な、既存のLDPスケジューリングアルゴリズムの多くの改善について説明している。使用されるスケジューリングフレームワークの一般性を考えると、本概念は、特に、自動車製造、供給チェーン管理を含む他の分野に適用できる。

20

【0043】

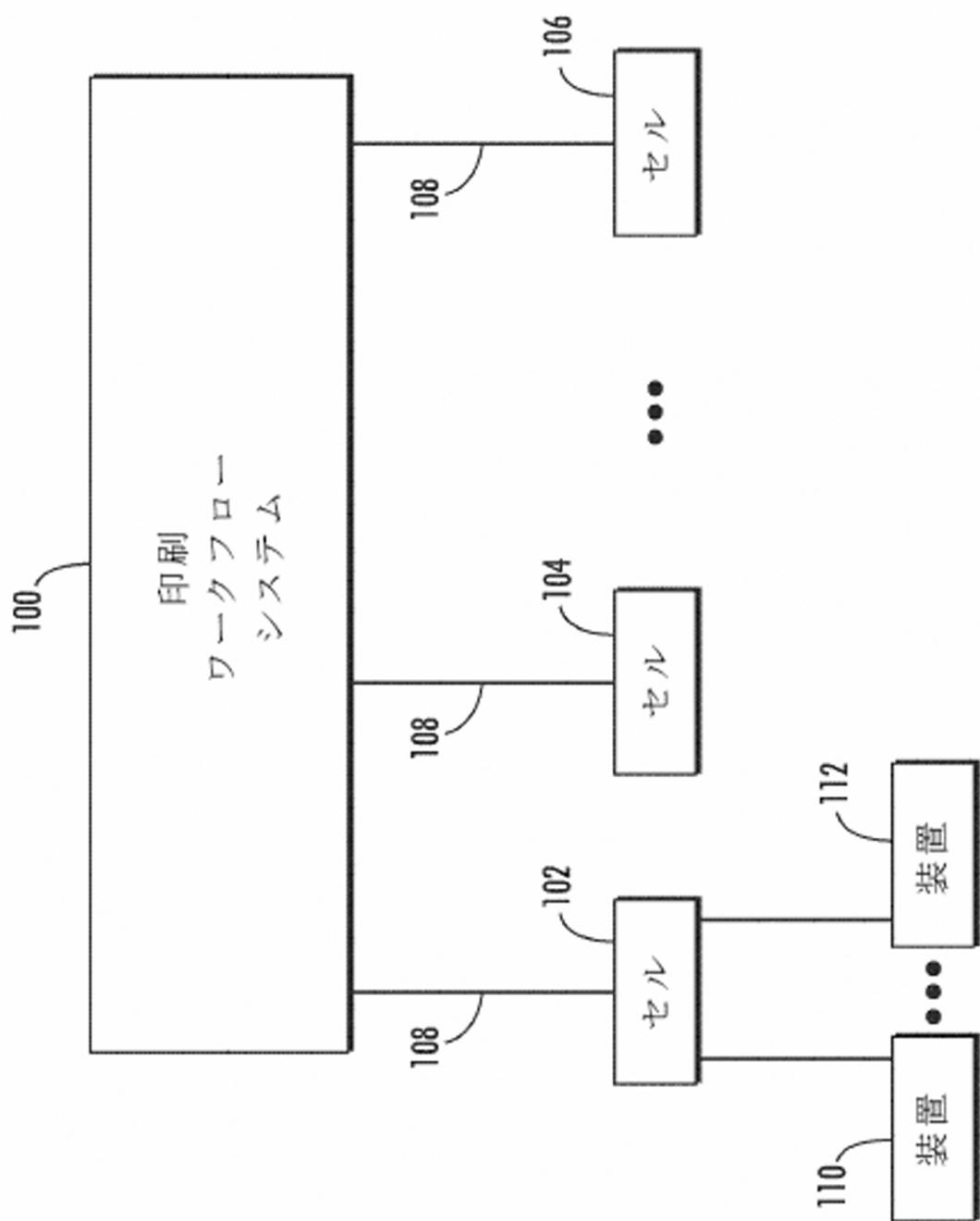
また、本願は、上述の方法を利用して、バッチ分割を備えたセル生産方式のための改良されたスケジューリング性能を実現するシステムを含んでいる。また、システムは、最早完了経路アルゴリズムおよび／または動的ロット／バッチ（改良されたシステム性能のための自動分割スケジューリングアルゴリズム）の並列実現を含んでいてもよい。並列であることにより、最適なスケジュールを戻すために自動分割作業および最早完了経路作業は2つのコアプロセッサにより並列処理してもよいことが分かる。

30

【0044】

上述の、および他の、特徴および機能、またはそれらの代替の変形が、他の多くの異なるシステムまたは応用において組み合わされてもよいことを理解されたい。これらの特徴および機能、またはそれらの代替の、現在のところ予見できない、または思いがけないさまざまな代替手段、変更、変形、または改良が、その後に当業者により行われる可能性があるが、これらの代替手段、変更、変形、または改良もまた下記クレームにより含まれるものとする。

【図1】



【図2】

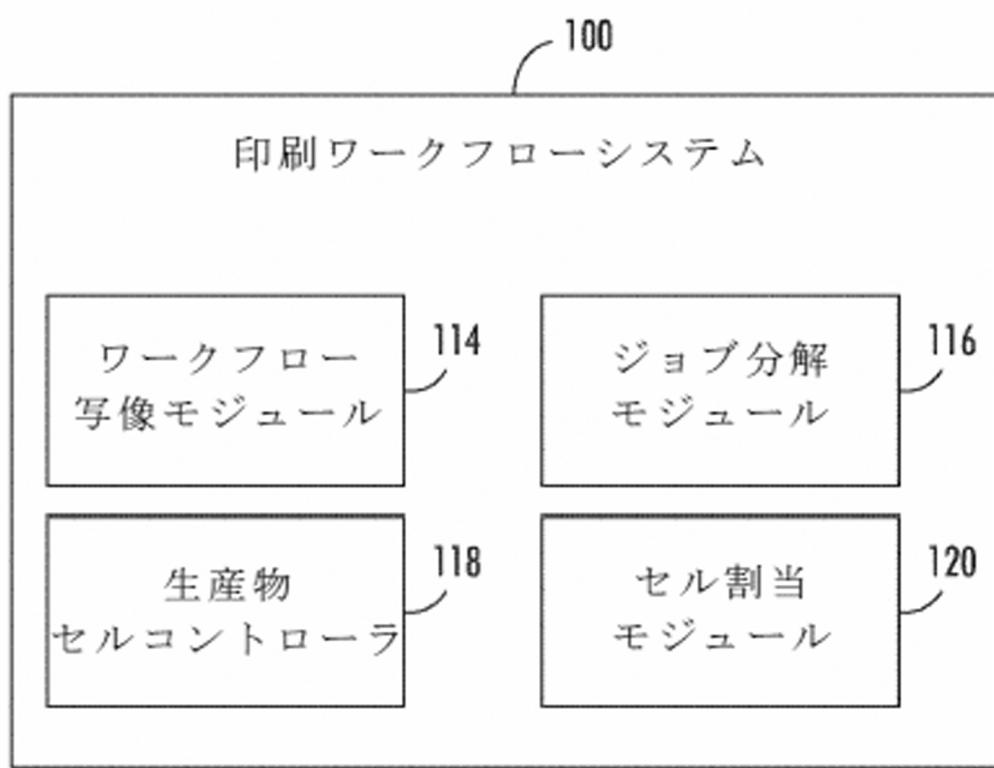


図2

【図3】

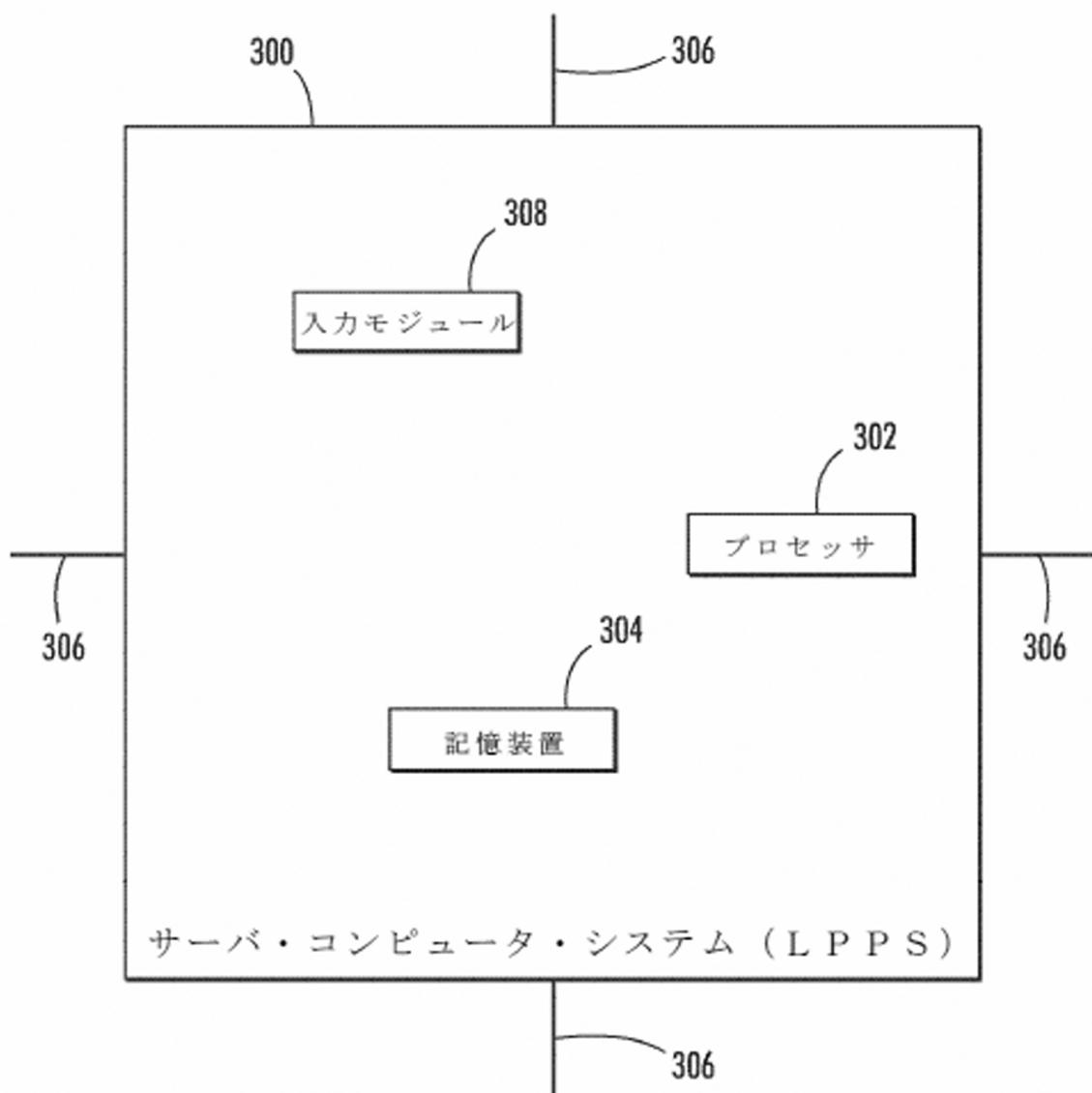
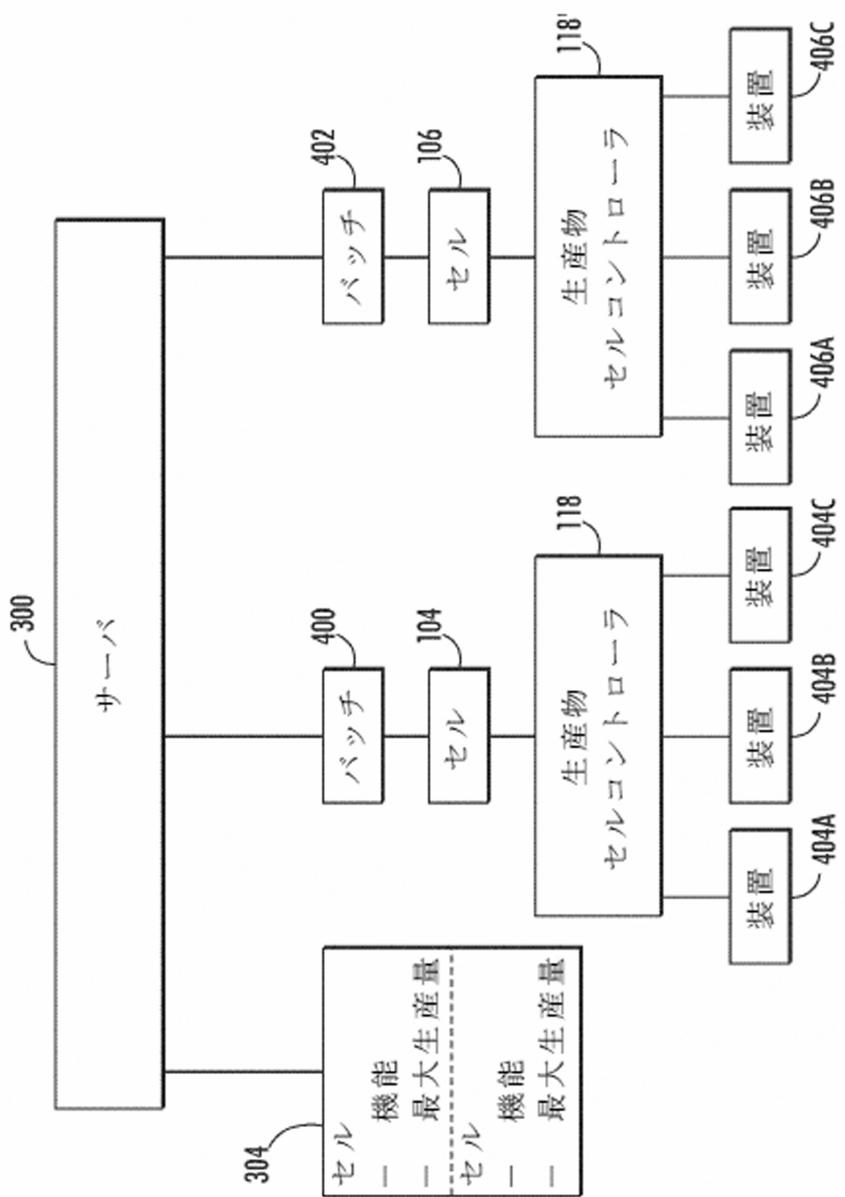


図3

【 図 4 】



4

【図5】

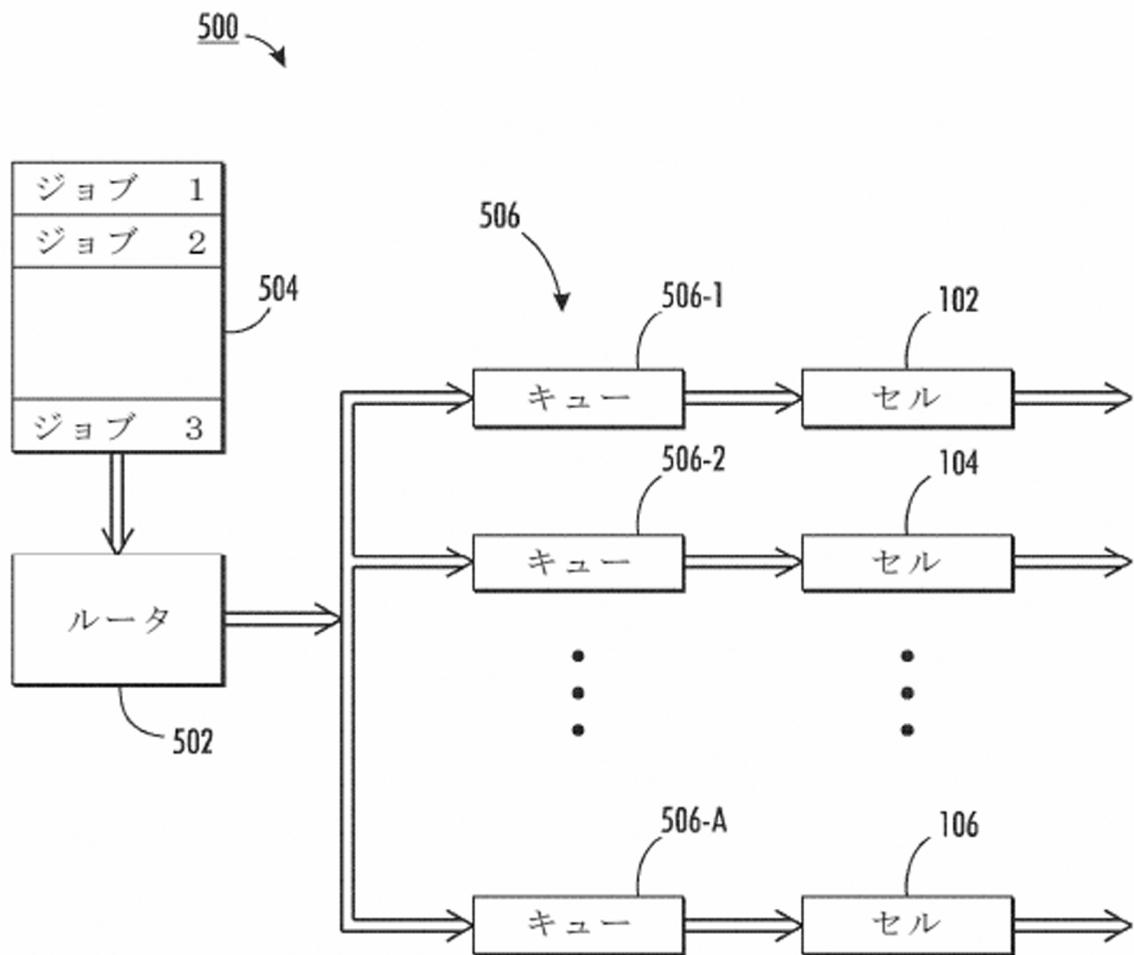


図5

【図6】

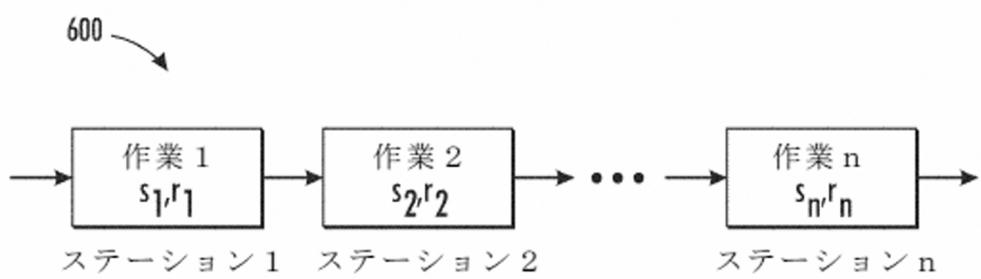


図6

【図7】

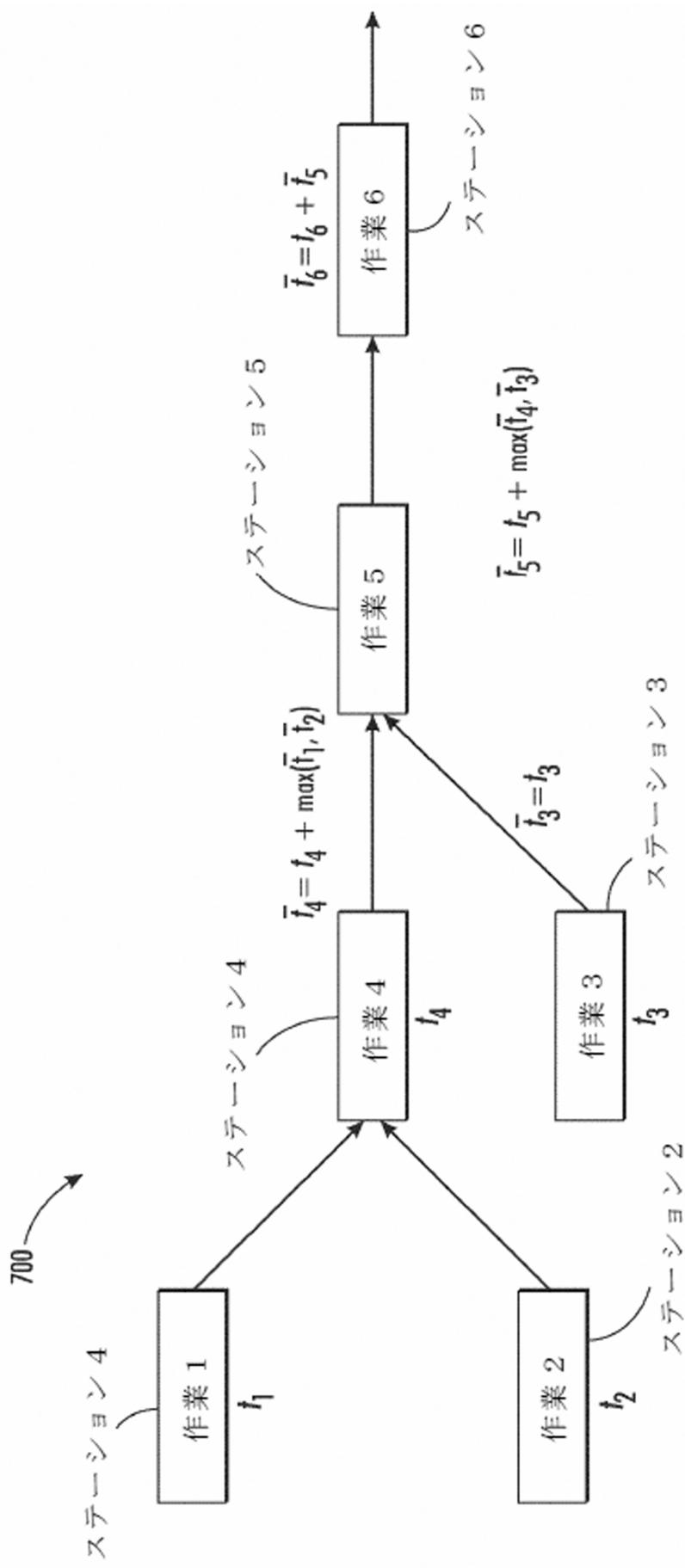


図7

【図 8】

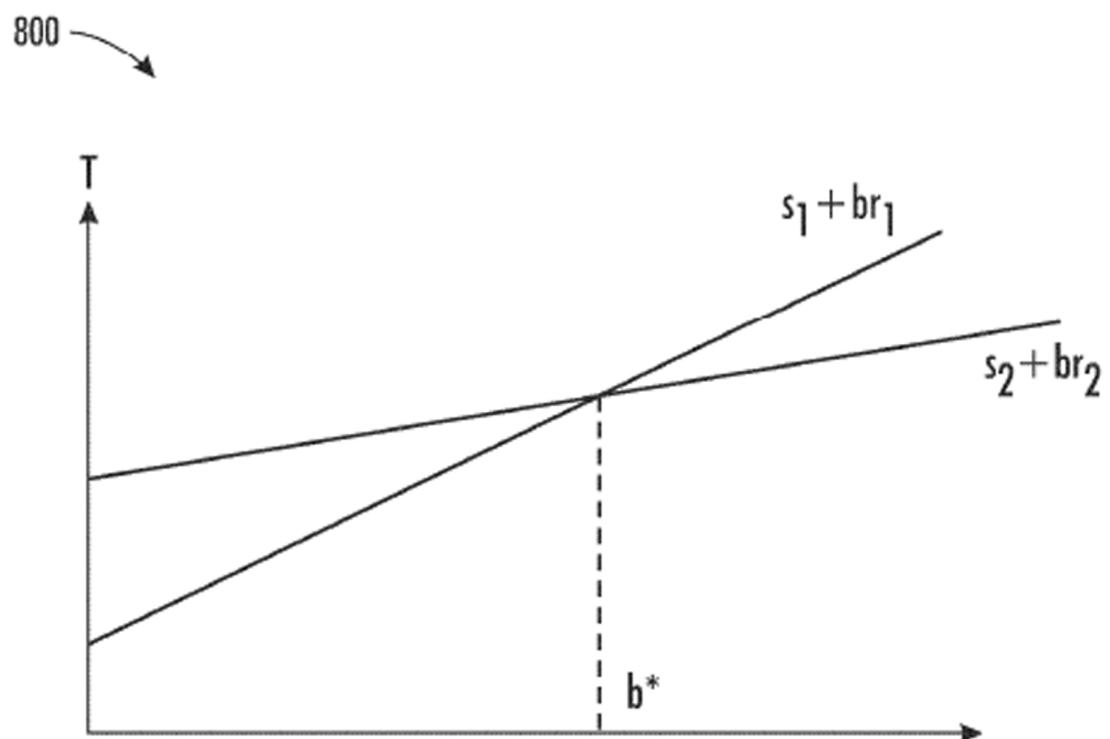


図 8

【図9】

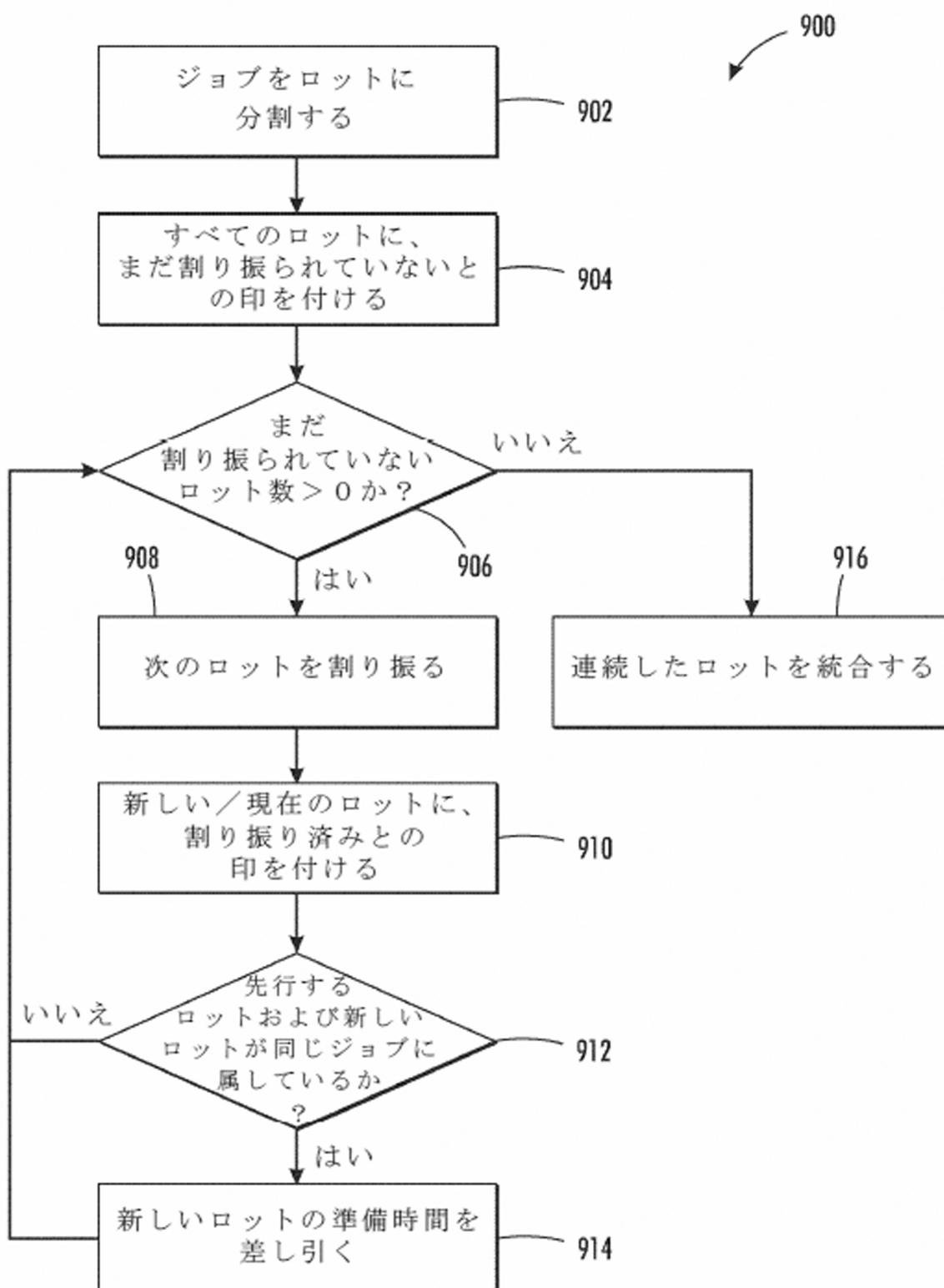


図9

【図10】

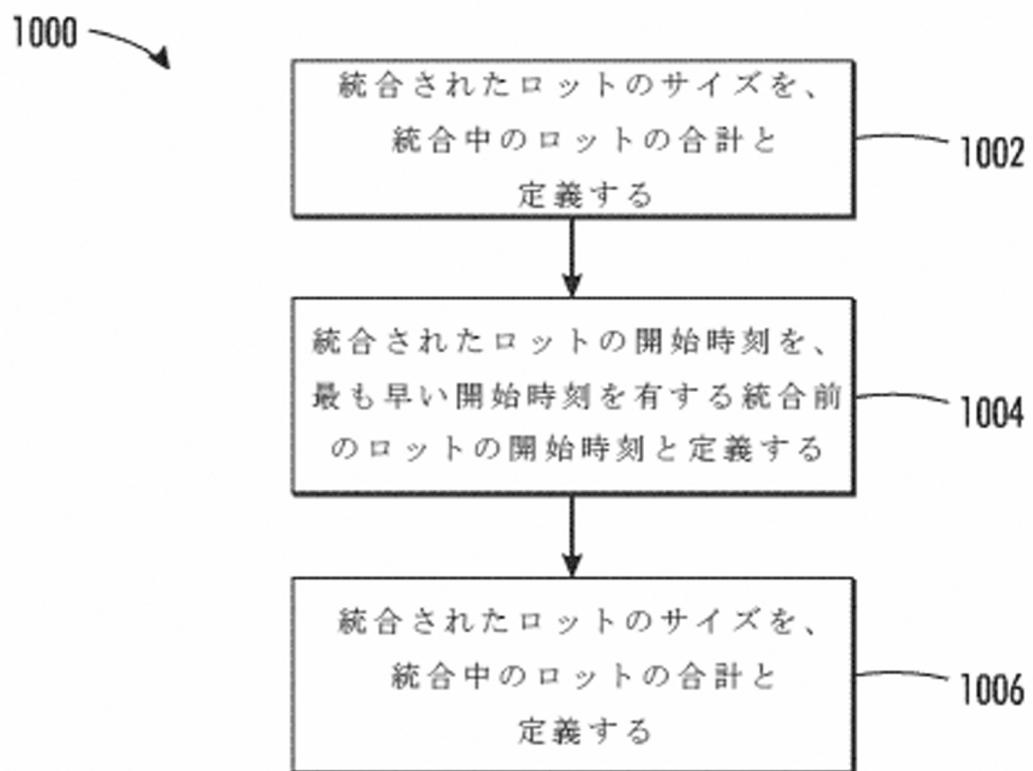


図10

【図11】

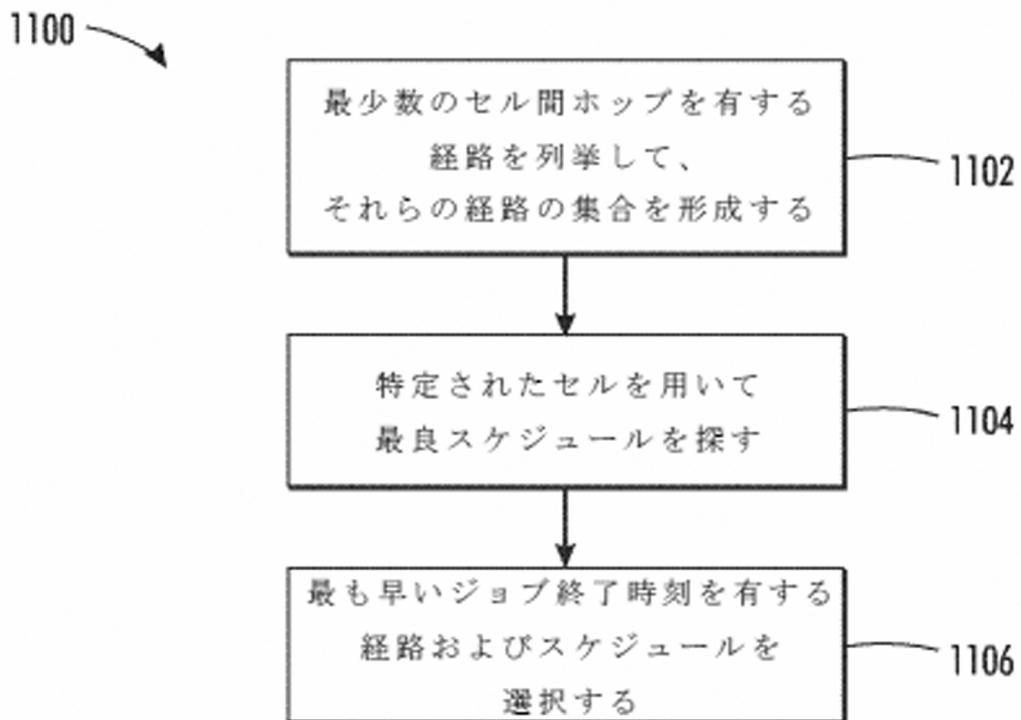


図11

【図12】

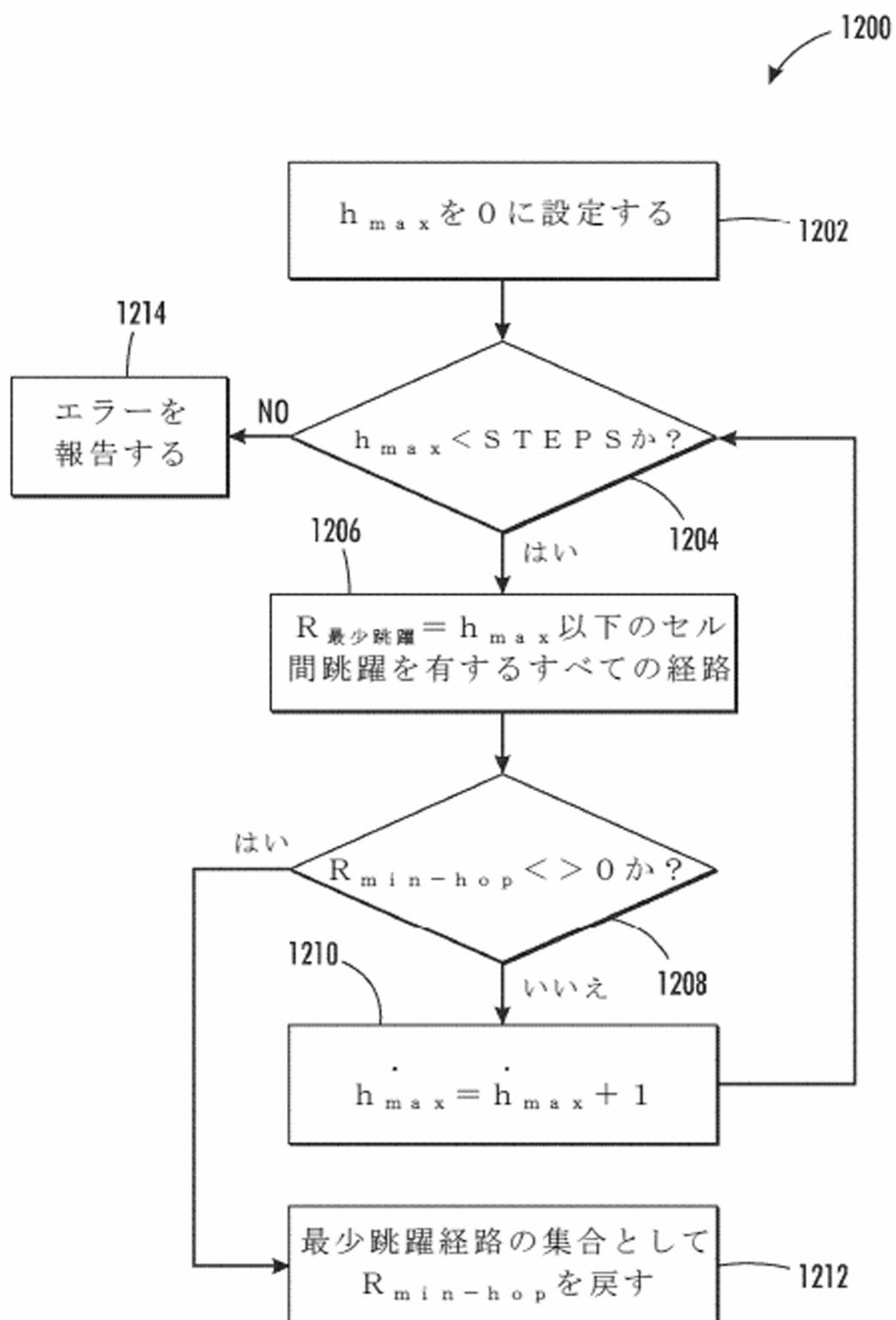


図12

フロントページの続き

(74)代理人 100079049
弁理士 中島 淳
(74)代理人 100084995
弁理士 加藤 和詳
(72)発明者 ロン・チョウ
アメリカ合衆国 カリフォルニア州 95129 サンノゼ ホワイトビック・ドライヴ 103
4
(72)発明者 サドヘンデュ・ライ
アメリカ合衆国 ニューヨーク州 14450 フェアポート マイルデンホール・リッジ 16
(72)発明者 ミン・ピン・ドウ
アメリカ合衆国 カリフォルニア州 94306 パロ・アルト エルスワース・プレース 70
6

審査官 川東 孝至

(56)参考文献 米国特許第07872767(US, B1)
米国特許出願公開第2006/0072148(US, A1)
米国特許出願公開第2002/0166005(US, A1)
米国特許第07079266(US, B1)
米国特許出願公開第2004/0196496(US, A1)
特開2001-326152(JP, A)
特開2012-068975(JP, A)

(58)調査した分野(Int.Cl., DB名)

G05B 19/418