

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5666613号  
(P5666613)

(45) 発行日 平成27年2月12日 (2015. 2. 12)

(24) 登録日 平成26年12月19日 (2014. 12. 19)

(51) Int. Cl.

F I

G 0 6 Q 50/30 (2012. 01)  
B 6 1 L 25/02 (2006. 01)G 0 6 Q 50/30  
B 6 1 L 25/02

請求項の数 32 (全 56 頁)

(21) 出願番号 特願2012-538965 (P2012-538965)  
 (86) (22) 出願日 平成22年11月11日 (2010. 11. 11)  
 (65) 公表番号 特表2013-511095 (P2013-511095A)  
 (43) 公表日 平成25年3月28日 (2013. 3. 28)  
 (86) 国際出願番号 PCT/US2010/056316  
 (87) 国際公開番号 W02011/060122  
 (87) 国際公開日 平成23年5月19日 (2011. 5. 19)  
 審査請求日 平成25年11月11日 (2013. 11. 11)  
 (31) 優先権主張番号 12/752, 980  
 (32) 優先日 平成22年4月1日 (2010. 4. 1)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 61/260, 342  
 (32) 優先日 平成21年11月11日 (2009. 11. 11)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 502208397  
 グーグル インコーポレイテッド  
 アメリカ合衆国 カリフォルニア州 94  
 043 マウンテン ビュー アンフィシ  
 アター パークウェイ 1600  
 (74) 代理人 100077539  
 弁理士 飯塚 義仁  
 (74) 代理人 100114742  
 弁理士 林 秀男  
 (74) 代理人 100125265  
 弁理士 貝塚 亮平

早期審査対象出願

最終頁に続く

(54) 【発明の名称】 公共交通機関トリップ計画のためのトランジットルート決定システム

(57) 【特許請求の範囲】

【請求項 1】

或る出発地点から或る目的地点へのジャーニーの公共的通行ルートを決定するために、コンピュータで実施される方法であって、前記公共的通行ルートには公共交通車両を持つ少なくとも1つの公共交通システムを使用し、前記方法はコンピュータにより実行され、且つ、

前記少なくとも1つの公共交通システムにおける複数の乗換パターンを第1のデータベース内に記憶するステップと、該記憶された乗換パターンの各々は、前記少なくとも1つの公共交通システムに関連する一対の通行駅間のルートを記述し、かつ、前記一対の通行駅間にある通行駅において公共交通車両を少なくとも1回乗り換えることを含み、

複数の公共通行駅を記述する情報であって各公共通行駅毎の場所を含む該情報を第2のデータベース内に記憶するステップと、

出発地点から目的地点への公共的通行ルートのリクエストをクライアント装置から受信するステップと、該公共的通行ルートは、少なくとも1つの公共交通システムを使用する前記出発地点から前記目的地点への道順を含み、

前記出発地点の或る半径距離内にある前記少なくとも1つの公共交通システムの通行駅を前記第2のデータベースを使用して決定し、それにより前記出発地点の前記半径距離内の通行駅を含む起点駅リストを生成するステップと、

前記目的地点の前記半径距離内にある前記少なくとも1つの公共交通システムの通行駅を前記第2のデータベースを使用して決定し、それにより前記目的地点の前記半径距離内

10

20

の通行駅を含む目的駅リストを生成するステップと、

前記起点駅リスト中の１つの起点駅と前記目的駅リスト中の１つの目的駅とを含む通行駅の賢明な組み合わせのペア毎に、前記起点駅から前記目的駅へ移動するための該賢明な組み合わせの１ペアにおける前記起点駅及び前記目的駅の間の中間通行駅での前記少なくとも１つの公共交通システムの公共交通車両の乗換を示す１つの記憶された乗換パターンを、前記複数の乗換パターンを記憶した前記第１のデータベースから検索するステップと、

前記検索された乗換パターン毎に、或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも１つの最適な公共的通行ルートを決

10

定するステップと

を備えることを特徴とする方法。

【請求項２】

前記出発地点は前記クライアント装置のユーザの現在の地理的位置であることを特徴とする請求項１に記載の方法。

【請求項３】

前記目的地点は前記クライアント装置のユーザが望む行き先の地理的位置であることを特徴とする請求項１に記載の方法。

【請求項４】

20

前記出発地点又は前記目的地点は或る通行駅の地理的地位であることを特徴とする請求項１に記載の方法。

【請求項５】

前記リクエストは、更に、前記出発地点からの出発時刻又は前記目的地点への到達時刻を含むことを特徴とする請求項１に記載の方法。

【請求項６】

前記１つの記憶された乗換パターンを前記複数の乗換パターンを記憶した前記第１のデータベースから検索する前記ステップは、

前記１つの記憶された乗換パターン用の複数の乗換駅ペアからなる１つのセットを検索するステップであって、前記複数の乗換駅ペアのそれぞれが、その乗換駅ペアに関連付けられた１つの駅に先行する連続する一対の駅を示すものと、

30

前記１つの記憶された乗換パターンを生成するために、前記複数の乗換駅ペアからなる１つのセットを連結するステップ

を備えることを特徴とする請求項１に記載の方法。

【請求項７】

前記少なくとも１つの公共交通システムの前記通行駅の賢明な組み合わせのペア毎の前記１つの記憶された乗換パターンを、該１つの記憶された乗換パターンにより表された順番で、弧によって連結されたノードのシークエンスとして表すグラフを生成するステップを更に備えることを特徴とする請求項１に記載の方法。

【請求項８】

40

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも１つの最適な公共的通行ルートを決

定する前記ステップは、  
前記グラフ内の各ノードを、そのノードに関連付けられた或る通行駅で乗車することになる前記少なくとも１つの公共交通システムの公共交通車両を表す駅ノード、及び、そのノードに関連付けられた或る通行駅に公共交通車両が到着している間乗車したままとする前記少なくとも１つの公共交通システムの公共交通車両を表す乗車ノードとして表すことにより、前記グラフを拡張するステップと、

前記起点駅から各後続駅への１以上の直通接続ルートを決

50

ルとを表すものと、

前記 1 以上の直通接続ルート毎に、前記拡張されたグラフ内のノードの連なりを、前記直通接続ルートを表す弧により連結するステップであり、前記ノードの連なりにおける各ノードはそれぞれ、前記 1 つの記憶された乗換パターンに関連付けられた駅を表すものとを備えることを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を決定するステップと、前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続を決定するステップと、

前記起点駅から前記目的駅へのルートを形成するために、前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を、前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続に合併するステップと、

前記拡張されたグラフ内のノードの連なりを、前記形成されたルートを表す弧により連結するステップと

を備えることを特徴とする請求項 8 に記載の方法。

【請求項 10】

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定する前記ステップは、

多次元のコストに基づいて前記起点駅から前記目的駅への前記少なくとも 1 つの最適な公共的通行ルートを決定するために、前記拡張されたグラフにパレートダイクストラ計算を適用することにより、前記少なくとも 1 つの最適な公共的通行ルートを計算するステップであって、前記多次元のコストは駅ペアの間の時間長とペナルティを含むもの、を備えることを特徴とする請求項 8 に記載の方法。

【請求項 11】

前記ペナルティは、金銭的成本、前記起点駅から前記目的駅に到達するために必要な公共交通車両の乗換数、及び、徒歩コストの少なくとも一部に基づくものであることを特徴とする請求項 10 に記載の方法。

【請求項 12】

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定する前記ステップは、

1 次元のコストに基づいて前記起点駅から前記目的駅への前記少なくとも 1 つの最適な公共的通行ルートを決定するために、前記拡張されたグラフにダイクストラ計算を適用することにより、前記少なくとも 1 つの最適な公共的通行ルートを計算するステップであって、前記 1 次元のコストは、前記起点駅から前記目的駅へのトリップを完遂するために要する総時間、前記起点駅から前記目的駅に到達するために要する乗換数、又は、金銭的成本のいずれか 1 つを含むものである

ことを特徴とする請求項 8 に記載の方法。

【請求項 13】

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定する前記ステップは、前記少なくとも 1 つの最適な公共的通行ルートを決定するために A \* 探索アルゴリズムを適用することを含むことを特徴とする請求項 8 に記載の方法。

【請求項 14】

或る出発地点から或る目的地点へのジャーニーの公共的通行ルートを決定するためにコンピュータにより実行可能なコードからなるコンピュータプログラムであって、前記公共的通行ルートには公共交通車両を持つ少なくとも 1 つの公共交通システムを使用し、コンピュータプロセッサに、

前記少なくとも 1 つの公共交通システムにおける複数の乗換パターンを第 1 のデータベース内に記憶するステップと、該記憶された乗換パターンの各々は、前記少なくとも 1 つの公共交通システムに関連する一対の通行駅間のルートを記述し、かつ、前記一対の通行

駅間にある通行駅において公共交通車両を少なくとも1回乗り換えることを含み、

複数の公共通行駅を記述する情報であって各公共通行駅毎の場所を含む該情報を第2のデータベース内に記憶するステップと、

出発地点から目的地点への公共的通行ルートのリクエストをクライアント装置から受信するステップと、該公共的通行ルートは、少なくとも1つの公共交通システムを使用する前記出発地点から前記目的地点への道順を含み、

前記出発地点の或る半径距離内にある前記少なくとも1つの公共交通システムの通行駅を前記第2のデータベースを使用して決定し、それにより前記出発地点の前記半径距離内の通行駅を含む起点駅リストを生成するステップと、

前記目的地点の前記半径距離内にある前記少なくとも1つの公共交通システムの通行駅を決定し、それにより前記目的地点の前記半径距離内の通行駅を含む目的駅リストを生成するステップと、

前記起点駅リスト中の1つの起点駅と前記目的駅リスト中の1つの目的駅とを含む通行駅の賢明な組み合わせのペア毎に、前記起点駅から前記目的駅へ移動するための該賢明な組み合わせの1ペアにおける前記起点駅及び前記目的駅の間の中間通行駅での前記少なくとも1つの公共交通システムの公共交通車両の乗換を示す1つの記憶された乗換パターンを、前記複数の乗換パターンを記憶した前記第1のデータベースから検索するステップと、

前記検索された乗換パターン毎に、或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも1つの最適な公共的通行ルートを決定するステップと、

前記少なくとも1つの最適な公共的通行ルートを記述する情報を前記クライアント装置に送信するステップと

を実行させることを特徴とするプログラム。

#### 【請求項15】

前記1つの記憶された乗換パターンを前記複数の乗換パターンを記憶した前記第1のデータベースから検索する前記ステップは、前記コンピュータプロセッサに、

前記1つの記憶された乗換パターン用の複数の乗換駅ペアからなる1つのセットを検索するステップであって、前記複数の乗換駅ペアのそれぞれが、その乗換駅ペアに関連付けられた1つの駅に先行する連続する一対の駅を示すものと、

前記1つの記憶された乗換パターンを生成するために、前記複数の乗換駅ペアからなる1つのセットを連結するステップ

を実行させることを特徴とする請求項14に記載のプログラム。

#### 【請求項16】

前記コンピュータプロセッサに、更に、前記少なくとも1つの公共交通システムの前記通行駅の賢明な組み合わせのペア毎の前記1つの記憶された乗換パターンを、該1つの記憶された乗換パターンにより表された順番で、弧によって連結されたノードのシーケンスとして表すグラフを生成するステップを実行させることを特徴とする請求項14に記載のプログラム。

#### 【請求項17】

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも1つの最適な公共的通行ルートを決定する前記ステップは、前記コンピュータプロセッサに、

前記グラフ内の各ノードを、そのノードに関連付けられた或る通行駅で乗車することになる前記少なくとも1つの公共交通システムの公共交通車両を表す駅ノード、及び、そのノードに関連付けられた或る通行駅に公共交通車両が到着している間乗車したままとする前記少なくとも1つの公共交通システムの公共交通車両を表す乗車ノードとして表すことにより、前記グラフを拡張するステップと、

前記起点駅から各後続駅への1以上の直通接続ルートを決定するステップであって、直通接続ルートは、前記起点駅から1つの後続駅へのルートと、前記1つの記憶された乗換

10

20

30

40

50

パターンに関連付けられた前記起点駅と前記後続駅の間の中間通行駅での停車スケジュールとを表すものと、

前記 1 以上の直通接続ルート毎に、前記拡張されたグラフ内のノードの連なりを、前記直通接続ルートを表す弧により連結するステップであり、前記ノードの連なりにおける各ノードはそれぞれ、前記 1 つの記憶された乗換パターンに関連付けられた駅を表すものを実行させることを特徴とする請求項 16 に記載のプログラム。

【請求項 18】

前記コンピュータプロセッサに、

前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を決定するステップと、

前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続を決定するステップと、

前記起点駅から前記目的駅へのルートを形成するために、前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を、前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続に合併するステップと、

前記拡張されたグラフ内のノードの連なりを、前記形成されたルートを表す弧により連結するステップと

を実行させることを特徴とする請求項 17 に記載のプログラム。

【請求項 19】

或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定する前記ステップは、前記コンピュータプロセッサに、

多次元のコストに基づいて前記起点駅から前記目的駅への前記少なくとも 1 つの最適な公共的通行ルートを決定するために、前記拡張されたグラフにパレートダイクストラ計算を適用することにより、前記少なくとも 1 つの最適な公共的通行ルートを計算するステップであって、前記多次元のコストは駅ペアの間の時間長とペナルティを含むもの、を実行させることを特徴とする請求項 17 に記載のプログラム。

【請求項 20】

或る出発地点から或る目的地点へのジャーニーの公共的通行ルートを決定するためのコンピュータシステムであって、前記公共的通行ルートには公共交通車両を持つ少なくとも 1 つの公共交通システムを使用し、前記システムは、

コンピュータプロセッサと、

前記コンピュータプロセッサに実行されるよう構成されたコンピュータ実行可能なコードを記憶するコンピュータ読み取り可能な記憶媒体とを備え、

前記コードが実行されるときに、

前記少なくとも 1 つの公共交通システムにおける複数の乗換パターンを第 1 のデータベース内に記憶することと、ここで、該記憶された乗換パターンの各々は、前記少なくとも 1 つの公共交通システムに関連する一对の通行駅間のルートを記述し、かつ、前記一对の通行駅間にある通行駅において公共交通車両を少なくとも 1 回乗り換えることを含み、

複数の公共通行駅を記述する情報であって各公共通行駅毎の場所を含む該情報を第 2 のデータベース内に記憶することと、

出発地点から目的地点への公共的通行ルートのリクエストをクライアント装置から受信することと、ここで、該公共的通行ルートは、少なくとも 1 つの公共交通システムを使用する前記出発地点から前記目的地点への道順を含み、

前記出発地点の或る半径距離内にある前記少なくとも 1 つの公共交通システム通行駅を前記第 2 のデータベースを使用して決定し、それにより前記出発地点の前記半径距離内の通行駅を含む起点駅リストを生成することと、

前記目的地点の前記半径距離内にある前記少なくとも 1 つの公共交通システム通行駅を前記第 2 のデータベースを使用して決定し、それにより前記目的地点の前記半径距離内の通行駅を含む目的駅リストを生成することと、

前記起点駅リスト中の１つの起点駅と前記目的駅リスト中の１つの目的駅とを含む通行駅の賢明な組み合わせのペア毎に、前記起点駅から前記目的駅へ移動するための該賢明な組み合わせの１ペアにおける前記起点駅及び前記目的駅の間の中間通行駅での前記少なくとも１つの公共交通システムの公共交通車両の乗換を示す１つの記憶された乗換パターンを、前記複数の乗換パターンを記憶した前記第１のデータベースから検索することと、

前記検索された乗換パターン毎に、或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも１つの最適な公共的通行ルートを決することと、

前記少なくとも１つの最適な公共的通行ルートを示す情報を前記クライアント装置に伝送することと、

前記少なくとも１つの最適な公共的通行ルートを記述する情報を前記クライアント装置に送信すること、

を実行するように構成されたシステム。

【請求項２１】

前記出発地点は前記クライアント装置のユーザの現在の地理的位置であることを特徴とする請求項２０に記載のシステム。

【請求項２２】

前記目的地点は前記クライアント装置のユーザが望む行き先の地理的位置であることを特徴とする請求項２０に記載のシステム。

【請求項２３】

前記出発地点又は前記目的地点は或る通行駅の地理的地位であることを特徴とする請求項２０に記載のシステム。

【請求項２４】

前記リクエストは、更に、前記出発地点からの出発時刻又は前記目的地点への到達時刻を含むことを特徴とする請求項２０に記載のシステム。

【請求項２５】

前記１つの記憶された乗換パターンを前記複数の乗換パターンを記憶した前記第１のデータベースから検索することは、

前記１つの記憶された乗換パターン用の複数の乗換駅ペアからなる１つのセットを検索することであって、前記複数の乗換駅ペアのそれぞれが、その乗換駅ペアに関連付けられた１つの駅に先行する連続する一対の駅を示すもの

前記１つの記憶された乗換パターンを生成するために、前記複数の乗換駅ペアからなる１つのセットを連結すること

からなることを特徴とする請求項２０に記載のシステム。

【請求項２６】

前記コードが実行されるときに、

前記少なくとも１つの公共交通システムの前記通行駅の賢明な組み合わせのペア毎の前記１つの記憶された乗換パターンを、該１つの記憶された乗換パターンにより表された順番で、弧によって連結されたノードのシーケンスとして表すグラフを生成すること、を更に実行するように構成された請求項２０に記載のシステム。

【請求項２７】

前記或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも１つの最適な公共的通行ルートを決することは、

前記グラフ内の各ノードを、そのノードに関連付けられた或る通行駅で乗車することになる前記少なくとも１つの公共交通システムの公共交通車両を表す駅ノード、及び、そのノードに関連付けられた或る通行駅に公共交通車両が到着している間乗車したままとする前記少なくとも１つの公共交通システムの公共交通車両を表す乗車ノードとして表すことにより、前記グラフを拡張することと、

前記起点駅から各後続駅への１以上の直通接続ルートを決定することと、ここで、直通接続ルートは、前記起点駅から１つの後続駅へのルートと、前記１つの記憶された乗換

10

20

30

40

50

パターンに関連付けられた前記起点駅と前記後続駅の間の中間通行駅での停車スケジュールとを表し、

前記 1 以上の直通接続ルート毎に、前記拡張されたグラフ内のノードの連なりを、前記直通接続ルートを表す弧により連結すること、  
からなり、ここで、前記ノードの連なりにおける各ノードはそれぞれ、前記 1 つの記憶された乗換パターンに関連付けられた駅を表すものである、請求項 26 に記載のシステム。

【請求項 28】

前記コードが実行されるときに、

前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を決定することと、

前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続を決定することと

10

、  
前記起点駅から前記目的駅へのルートを形成するために、前記起点駅から少なくとも 1 つの後続駅への 1 以上の直通接続を、前記少なくとも 1 つの後続駅から前記目的駅への 1 以上の直通接続に合併することと、

前記拡張されたグラフ内のノードの連なりを、前記形成されたルートを表す弧により連結すること、

を更に実行するように構成された請求項 27 に記載のシステム。

【請求項 29】

前記或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定することは、

20

多次元のコストに基づいて前記起点駅から前記目的駅への前記少なくとも 1 つの最適な公共的通行ルートを決定するために、前記拡張されたグラフにパレートダイクストラ計算を適用することにより、前記少なくとも 1 つの最適な公共的通行ルートを計算することからなり、前記多次元のコストは駅ペアの間の時間長とペナルティを含む、  
ことを特徴とする請求項 27 に記載のシステム。

【請求項 30】

前記ペナルティは、金銭的成本、前記起点駅から前記目的駅に到達するために必要な公共交通車両の乗換数、及び、徒歩コストの少なくとも一部に基づくものであることを特徴とする請求項 29 に記載のシステム。

【請求項 31】

30

前記或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定することは、

1 次元のコストに基づいて前記起点駅から前記目的駅への前記少なくとも 1 つの最適な公共的通行ルートを決定するために、前記拡張されたグラフにダイクストラ計算を適用することにより、前記少なくとも 1 つの最適な公共的通行ルートを計算することからなり、前記 1 次元のコストは、前記起点駅から前記目的駅へのトリップを完遂するために要する総時間、前記起点駅から前記目的駅に到達するために要する乗換数、又は、金銭的成本のいずれか 1 つを含む、

ことを特徴とする請求項 27 に記載のシステム。

【請求項 32】

40

前記或る特定時刻での前記乗換パターンの具体例である前記起点駅から前記目的駅への少なくとも 1 つの最適な公共的通行ルートを決定することは、前記少なくとも 1 つの最適な公共的通行ルートを決定するために A \* 探索アルゴリズムを適用することを含むことを特徴とする請求項 27 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、公共交通機関トリップ (trip: 旅行、移動) 計画に関し、より詳しくは、処理されたトランジット (transit: 交通、通行、乗換) 情報を用いて公共交通機関の指示を提供するために必要な計算時間と資源を極小化することである。

50

## 【 0 0 0 2 】

関連する出願の参照

この出願は、2009年11月11日に出願された米国特許仮出願番号第61/260342号に基づく利益を主張し、当該仮出願に記載された全ての記載内容を援用するものである。

## 【 背景技術 】

## 【 0 0 0 3 】

公共交通機関インフラストラクチャーの向上、環境意識の高まり、及び、ガソリン料金の値上がりに伴い、多くの人々が各種形態の公共交通機関を使用し始めた。彼ら自身が所有する車両を使用するよりも、人々は、移動の必要に際して、いや増しに、鉄道、バス及び/又はフェリーを使用している。この公共交通機関の使用の増加のため、公共トランジット計画システムが発達している。これらシステムは、種々の形式の公共交通機関を経由する、スタート地から目的地までの間の移動の指示をユーザに供給する。

10

## 【 0 0 0 4 】

典型的には、これらのシステムにおいて、スタート地と目的地を含む問い合わせが受信され、トランジット計画システムは、1以上の形式の公共交通機関を用いた目的地に到達するための指示を漸次供給する。例えば、指示は、トリップの目的地に到達するためにどの公共トランジット車両（例えばバス、電車など）を使い、どの公共トランジット地点で停まるべきかのシーケンスを含む。指示は、旅中の種々のトランジット地点で必要とされる別のトランジット車両への乗換を含む。従って、計画システムは、公共交通機関を用いたトリップを容易に計画できる情報を人々に提案する機構を提供する。

20

## 【 0 0 0 5 】

典型的には、或るトリップのための指示を決定するために、これら従来の公共トランジット計画システムは、クエリ（問い合わせ）時間に、目的地に到達するための最適なルートを決めるためにトリップのスタートから行き先地点までの種々のルート进行分析する。この手法は、2地点間に比較的少数の可能なルートしかないときには、利用可能なトランジットの選択肢の数が限られているため、有益である。しかし、公共交通機関インフラストラクチャーの拡張と、異なる交通システムの間での乗換能力（例えば、共通の駅でのバスと電車の乗換）とのため、任意のスタート地及び目的地の間での多数の可能なルートが著しく増えている。従って、前記クエリ時間の間に目的地に到達するための最適ルートを計算するために必要な時間が劇的に増加しており、それにより、結果を得るためにユーザが待たなければならない時間が増加する。

30

## 【 発明の概要 】

## 【 0 0 0 6 】

公共トランジット旅行（travel:トラベル）計画システム及び方法は、公共トランジット（transit:交通機関、輸送、通行）を使ったジャーニー（journey:旅行）又はトリップ（trip:旅行）のためのクエリ（問い合わせ）に応じて、クエリ時間より前に、最適な公共トランジットのルートを決めるために、前処理されたトランジット情報を使用するように構成される。最適な公共トランジットのルートは、或るトリップのための時間とその他の要素に関連する最良のルートであり、これは所与の出発地点から目的地に到達するために公共交通機関及び/又は徒歩のみを使うものである。公共トランジット旅行計画システムは、クエリ時間より前に、任意の2つのトランジット駅の間での最適な乗換パターンを決めるためのトランジット情報（基本的な公共トランジットのスケジュールを表す情報）を処理する。より具体的には、乗換パターンは、目的地に到着するためになされるべき1以上のトランジット駅でのトランジット車両の乗換のシーケンスを表す。クエリ時間より前に、最適な公共トランジットのルートを決めるためトランジット情報を前処理することにより、クエリ時間の間の、或る公共トランジットルート用のユーザクエリを遂行するための計算量が極小となる。一般に、システムは前クエリ計算及び後クエリ計算を実行する。

40

## 【 0 0 0 7 】

50



クエリ時間より前に、システムは、種々のトランジット機関から受信したトランジット情報を既知のトランジット駅間の最適公共交通機関乗換パターンを計算する。記憶された各トランジットトリップは、起点位置と目的位置を含むトランジット情報により表される。トランジット駅におけるトランジット車両の1以上の停車スケジュール(すなわちルート)は、各記憶されたトリップから検索される。トランジットグラフは、各記憶されたトリップのルートを、弧によって連結された一連のノードとして表すものとして生成される。トランジットグラフ内の各ノードは、トランジット駅で実行される1つのイベントであって、そのトリップに関連するトランジット車両のイベントを表す。イベントの一例は、トランジット駅に到着する又はトランジット駅を出発するトランジット車両である。トランジットグラフ内に表された各トランジット駅のペア毎の最適乗換パターンは、乗換の数、旅の時間及び/又はその他要素の点から、最良の乗換ルートを表すものとなるように計算される。前述の通り、最適乗換パターンは、グラフ内の駅ペアの間のトランジット駅での1以上の乗換を表している。トランジット駅のペア毎の各最適乗換パターンは、クエリ時間における後の使用のために、記憶される。

10

#### 【0008】

クエリが受信されたら、システムは、前記クエリに含まれる所与のスタート地点と目的地点の間の公共トランジットルートを決定するために記憶された最適乗換パターンを使う。前記クエリを受信した後、システムは、スタート地点近在の駅を表す起点駅リストを生成するために、出発地点のラジアル距離内のトランジット駅を決定する。システムは、同様に、目的位置近在の駅を表す目標駅リストを生成するために、目的位置のラジアル距離内のトランジット駅を決定する。起点駅リストからの起点駅と目的駅リストからの目的駅とを示すトランジット駅の賢明な組み合わせのペア毎に、記憶された乗換パターンが検索され、その乗換パターンは起点駅と目的駅の間の中間駅でのトランジット車両の乗換を表す。クエリ時間より前に乗換パターンが既に計算されているので、システムはただ乗換パターンを検索しさえすればよい。システムは、そして、検索された乗換パターンに基づいて、起点駅から目的駅までの最適ルートを少なくとも1つ決定する。そして、最適ルートは、クエリを提示したユーザのクライアント装置に伝送される。

20

#### 【0009】

この概要及び以下の詳細な説明に記載された特徴及び利点は包括的なものではない。多くの付加的な特徴及び利点が、本願の図面、明細書及び特許請求の範囲を閲覧する当業者にとって明らかになるだろう。

30

#### 【図面の簡単な説明】

#### 【0010】

【図1】一実施形態に従う、コンピュータ処理環境の概念的ブロック図。

#### 【0011】

【図2】一実施形態に従う、クエリ時間より先の、公共交通機関情報の前計算処理のフローチャート。

#### 【0012】

【図3】一実施形態に従う、トランジットグラフを生成する処理のフローチャート。

#### 【0013】

40

【図4】一実施形態に従う、トランジットグラフの一例を説明する図。

#### 【0014】

【図5A】一実施形態に従う、グローバル駅を決定する処理のフローチャート。

【図5B】一実施形態に従う、グローバル駅を決定する処理のフローチャート。

#### 【0015】

【図6】一実施形態に従う、直通接続を決定する処理のフローチャート。

#### 【0016】

【図7】一実施形態に従う、直通接続トリップの一例を示す図。

#### 【0017】

【図8】一実施形態に従う、グローバル駅を使わない乗換パターンを計算する処理のフロ

50

ーチャート。

【0018】

【図9】一実施形態に従う、グローバル駅を含む乗換パターンを計算する処理のフローチャート。

【0019】

【図10A】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

【図10B】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

【図10C】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

10

【図10D】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

【図10E】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

【図10F】一実施形態に従う、グローバル駅を用いたヒューリスティック乗換パターンを計算する処理のフローチャート。

【0020】

【図11】一実施形態に従う、乗換パターンを圧縮する処理のフローチャート。

【0021】

20

【図12】一実施形態に従う、ユーザクエリを遂行するためにクエリ時間に実行される乗換パターンを圧縮する処理のフローチャート。

【0022】

【図13】一実施形態に従う、起点位置及び目標位置に近在の駅を決定する処理のフローチャート。

【0023】

【図14】一実施形態に従う、クエリグラフを生成する処理のフローチャート。

【0024】

【図15】一実施形態に従う、クエリグラフの一例を示す図。

【0025】

30

【図16】一実施形態に従う、クエリグラフを拡張する処理のフローチャート。

【0026】

【図17】一実施形態に従う、拡張されたクエリグラフの一例を示す図。

【0027】

【図18】一実施形態に従う、拡張されたクエリグラフから最適なトリップを決定する処理のフローチャート。

【0028】

【図19】一実施形態に従う、最適なトリップを含むユーザインタフェースの一例を示す図。

【発明を実施するための形態】

40

【0029】

これら図面は、本発明の種々の実施形態を説明するためにのみ描かれている。当業者は、以下の議論から、ここに開示された発明の本質から逸脱することなく、ここで説明された構成及び方法とは別の実施形態を採用できることが容易に理解できる。

#### システム概要

【0030】

公共トランジット旅行 (travel) 計画システム及び方法は、ジャーニー (journey: 旅行) のための最適公共トランジットルートを決めるために、クエリ時間に先立って、トランジット情報の広範な前処理を用いるように構成されている。最適公共トランジットルートは、時間及び/又は、その他の要素に関して最良の、出発地点から行き先地点に到達す

50

るために使用されるルートである。最適公共トランジットルートは、公共交通機関のみを使った（複数の異なる種類の公共交通機関が1つのルート内で使用されうる）、また、徒歩も含みうる、出発地点から行き先地点までの指示を備える。ルートは、この旅程にどのトランジット駅が使用されるかを表しており、また、目的地に到達するために必要とされる1以上のトランジット車両への任意の乗換（異なる種類の輸送（トランスポート）車両の間の乗換を含む）も表している。

#### 【0031】

トランジット旅行計画システムは、例えばバス路線や鉄道路線など公共交通機関を提供するトランジット機関からトランジット情報を受信する。トランジット情報は、1以上の形式の交通機関を用いる、種々の場所の間のトリップ（trip:旅行、移動）を表す。トランジット情報は公共トランジットのトリップのスケジュール情報を表しており、そのスケジュール情報は、当該トリップに関連する公共トランジット車両が駅に到着及び/又は駅から出発する時間のリストを表している。トランジット旅行計画システムは、クエリ時間に先行して、任意の2つのトランジット駅間のルートを表す最適乗換パターンを決定するために、トランジット情報を処理する。乗換パターンは、各旅の中で、どこでトランジット車両の乗換がなされるかを表している。従って、トランジット旅行計画システムは、何れの所与の駅のペアについても、可能な限りでの最良のルートを決

10

#### 【0032】

最適乗換目標を決定するために、トランジット旅行計画システムは、トランジットグラフ及びそれに関するトランジットテーブルであって、トランジットグラフの情報をテーブル形式で表すものを、生成する。トランジットグラフは、トランジット機関からの、例えば1週間又は1ヶ月など或る日数窓分のトランジット情報により指定される可能なトリップを表している。一般的に、各既知のトリップ毎に、トランジットグラフは、有方向の弧により連結された一連のノードを備えており、各ノードは、例えば特定の時間の公共トランジット車両の到着又は出発等の、駅でのイベントを表している。ノードのペアは、ペアにおける公共トランジット車両の移動を表す有方向の弧により連結される。

20

#### 【0033】

トランジット旅行計画システムは、トランジットグラフ内のトリップを表すために、第1の駅（起点ノード）から始まり第2の駅（目的ノード）に終わるトリップのトランジット情報を検索する。トランジット情報は、トランジット駅での公共トランジット車両についての、実際の時間及び位置（例えばスケジュール）の情報を備える。トランジット旅行計画システムは、そのトリップに関連するトランジット情報に基づいてトランジットグラフを作成する。トランジット旅行計画システムは、各トランジット車両の駅への到着及び/又は駅からの出発毎に、トランジットグラフに、特定駅の特定時間のイベントを表すノードを挿入する。

30

#### 【0034】

トランジット旅行計画システムは、トリップのルートを表す有方向の弧を使って複数ノードを連結する。各弧は起点ノードから目標ノードに到達するまでのコストに関連付けられている。トランジット旅行計画システムは、各弧のコストを決定するために、そのトリップに関連するトランジット情報を使用する。トランジット旅行計画システムは、トランジット情報が受信されたトリップ毎に、トランジットグラフに表される日数窓内の1日毎のトランジットグラフが作成されるまで、上述の処理を繰り返す。

40

#### 【0035】

トランジット旅行計画システムは、トランジットグラフの作成の後、当該トランジットグラフに基づいて、一続きのトランジットテーブルを作る。トランジットテーブルは、トランジットグラフ内の各ノード及び弧、トランジットグラフ内の各トリップが有効なときの日数窓内の日数、及び、駅の地理的位置を表す情報を含む。トランジットテーブルは、基本的には、トランジットグラフを表すテーブルである。一実施形態において、トランジット情報から、トランジットグラフを作るのではなく、トランジットテーブルが生成される。

50

## 【 0 0 3 6 】

トランジットグラフ及びトランジットテーブルを用いて、トランジット旅行計画システムは、全ての直通接続トリップを決定する。直通接続トリップは、起点駅から目的駅まで乗り換え無しのトリップである。また、1以上の停止がトリップ中になされるが、目的駅に到達するために乗り換えを行う必要がないならば、起点駅から目的駅は直通接続である。更に、トランジットトラベルトラベルシステムは、トランジットグラフ中のどの駅がグローバル駅(global station:汎用駅)であるかを決定する。グローバル駅は、長い距離のトリップの間にトランジット車両の乗り換えが頻繁に行われる駅である。直通接続情報はテーブル形式で記憶されてよい。

## 【 0 0 3 7 】

トランジット旅行計画システムは、トランジットグラフと直通接続情報テーブルを用いて、任意の所与の起点駅から、最小のコストで、トランジットグラフ内の任意の目的駅(すなわち行き先)到達するためのルートを表す最適乗換パターンを決定する。一実施形態において、乗換パターンは、公共トランジットルートの途中で、ユーザが公共交通機関の車両から降り(すなわち下車する)て、別の公共交通機関の車両に乗る(すなわち乗り換える)種々の公共トランジット駅(ここでは単に「駅」という)における、乗換のシーケンスを表す。好ましくは、乗換パターンは、駅間のトラベル時間、待ち時間及びその他のコスト要素を考慮に入れるコスト関数に関して(コストの状態により)最適化される。一般に、旅の乗換パターンは、トランジット車両の乗換が行われる駅とともに起点及び目的駅を含む。例えば、起点駅A, 目的駅Fであり、駅D, Eで乗換が行われるとする。この例では、乗換パターンは、A D E Fである。一実施形態において、乗換パターンは、乗り換える駅のための、時刻に論及することなく、駅のシーケンスを表しており、それゆえ、パターンは、単に具体的時刻で予め計算されたトリップではなく、複数の異なるスタート時間に関して具体例を挙げて示すことができるパターンとなる。

## 【 0 0 3 8 】

一般に、トランジット旅行計画システムは、最適乗換ルートを決断するために種々の方法論を使用する。方法論は、大域駅の概念を使用してもよいし、しなくても良く、或いは、乗換パターンを決断するためにヒューリスティックアプローチを使用してもよい。使用される方法論に関わらず、トランジット旅行計画システムは、所与の起点駅から目標駅に到達するためのコストに関して最良の経路を決断するために、例えばダイクストラアルゴリズムなどのグラフ探索アルゴリズムを用いる。従って、トランジット旅行計画システムは、トランジットグラフ内の起点駅と目的駅との全ての賢明な組み合わせのペアに対して、1以上の最適乗換パターンをもたらし、或いは、ルートが存在しなければ、乗換パターンを出さない。

## 【 0 0 3 9 】

乗換パターンは、クエリ処理時間における後の使用のために、記憶される。一実施形態において、各乗換パターンは、個別に且つ完全な形で記憶される。別の例として、トランジット旅行計画システムは、各乗換パターンを完全に記憶するのではなく、乗換パターンを圧縮する。乗換パターン内に含まれる情報は、基本的に、乗換パターン内に含まれる情報の冗長性のために、圧縮されうる小部分に分解されている。乗換パターンが圧縮されると、トランジット旅行計画システムは、圧縮された乗換パターンを記憶する。

## 【 0 0 4 0 】

クエリ時間において、記憶された乗換パターンは、起点位置から目的地までの指示のためのユーザクエリに応じた乗換ルートを生成するために使用される。一般に、ユーザは、或るジャーニーについて、すなわち、ユーザの現在位置又は興味のある位置から目的地までの指示を要求する。起点位置及び目的地は、それぞれ、ユーザの現在位置及び興味の或る場所よりもむしろ、起点駅及び目的駅に関連付けられている。要求は、出発又は到着イベントに関連する日付及び/又は時間もまた含む。トランジット旅行計画システムは、そして、起点駅及び目的駅に近在する乗換駅を決断する。従って、起点及び目的に近い乗換駅は、異なる起点位置及び目的地のペアを形成するためにペア化されうる。乗換

10

20

30

40

50

パターンは、予め計算されている記憶された乗換パターン情報の駅のペア毎に決定される。一般に、トランジット旅行計画システムは、駅のペアに関連する、記憶された乗換パターンを検索する。

【 0 0 4 1 】

乗換パターンを用いて、トランジット旅行計画システムはクエリグラフを生成する。クエリグラフは、起点位置から目的地までの最適ルートを決断するために使用される。クエリグラフは、クエリに関連するトランジット情報のみが含まれる。乗換パターン内の各駅は、クエリグラフ内のノードとして表され、ノードのペアはトランジットグラフと同様に有方向の弧により連結される。クエリグラフが生成されると、トランジット計画システムは、乗換パターン内の駅ペア間の複数の直通接続トリップを表す直通接続トランジット情報を用いて、クエリグラフ内の駅間の直通接続を決断する。

10

【 0 0 4 2 】

直通接続の決定は、検索クエリ内で受信された日付及び/又は時間により決定される。或る検索が或る日付の出発時間を含んでいる場合、指定された日付の、出発時間よりも後に出発する直通接続のみが検索される。同様に、到着時間が指定された場合は、その到着時間よりも前に到着する直通接続のみが決定される。直通接続トリップが決定されたら、トランジット旅行計画システムは、直通接続トリップのコストに基づく計算を行うことにより最適トリップを決断する。計算は、例えば時間又はトランジット車両の多様性など種々の要素に基づいて最適な複数のトリップをもたらす。そして、最適トリップは、検索クエリを遂行するために、ユーザに供給される。

20

【 0 0 4 3 】

従って、クエリ時間において、トランジット情報がトランジット旅行計画システムにより既に処理されているので、クエリを遂行するには非常に少ない時間しか要さない。例えば、カリフォルニア州サンノゼから、カリフォルニア州サンフランシスコへの最良の直通接続トリップを要求するクエリは、サンノゼからサンフランシスコへの記憶された直接接続を検索するために、前処理されたトランジット情報における相対的に短時間低価格の探索として処理できる。別の例として、クエリが、イギリスのドミニントンパーク乗り換えで、イタリアのムジェロから、オランダのアッセンへのトリップを要求する場合、クエリ時間よりも先に、最適乗換パターンが既に計算され記憶されているので、クエリは処理された乗換情報上の2つの直通接続クエリ経由で遂行できる。トランジット旅行計画システムは、ムジェロからオランダへの直通接続、及び、オランダからドミニントンパークへの直通接続の、2つの探索を実行する。従って、この発明のトランジット旅行計画システムは、公共トランジット指示のユーザクエリが遂行されるクエリ時間を最短で提供する。

30

【 0 0 4 4 】

図1を参照すると、この発明の一実施形態に従うコンピュータ処理環境の概念的ブロック図が示されている。トランジットサーバ100(すなわち、トランジット旅行計画システム)は、特定の最適公共トランジットトリップを供給するために記憶された最適乗換パターンの広範な前処理アプローチを用いて、所与の日付及び時間の公共トランジットジャーニーのためのクエリに応じた公共トランジット(すなわち公共交通機関)指示をユーザに提供する。図1が一実施形態を示している一方で、この発明の全ての実施形態が、ここに記載される動作の実行及び種々のタイプのデータの記憶のために、コンピュータシステムの使用を必要とすることが理解されるべきである。以下の議論において、全ての動作、ステップ、データなどが、コンピュータシステム及びプログラミングロジックの組み合わせにより実装されるものであり、人の頭の中でのステップ又は実体化されない抽象的な概念を通じたどんな方法でも実行されないことは当然である。

40

【 0 0 4 5 】

一実施形態において、公共トランジットジャーニーは、公共交通機関のみを使用した、時間を特定された、起点(つまり出発)地点から目的(つまり行き先)地点へのルート又は経路からなる。種々の形式の公共交通機関は、バス、ライトレール(軽量軌道)、地下鉄、トロリー(市街電車)、通勤列車、フェリー又は飛行機などを含んでよい。当業者は

50

、上記列挙したもの以外にも公共交通機関の別の形式があり、それを公共交通トリップ上の地点間を移動するための乗り物として使用できることを理解するだろう。以下では、簡略化のために、公共交通トリップを「トリップ」と称する。

【 0 0 4 6 】

前処理された最適乗換パターン情報を使用することにより、トランジットサーバ 1 0 0 は、或るトリップのクエリを実行するためにクエリ時間に最小量の計算を実行する。これにより、ユーザの要求に対する最も迅速な応答時間が可能となる。特定時間におけるスタート地点から行き先地点へのトランジット指示のためのユーザのクエリに応じて、トランジットサーバ 1 0 0 はユーザに、所望されたジャーニーのための 1 以上の最適公共交通ルートを提供する。

10

【 0 0 4 7 】

一実施形態において、最適公共交通ルートは、スタート地点近在の駅から目的地地点近在の駅までのルートであって、コストに関して最適化されたルートである。トリップのコストは、一実施形態に従い目的地地点近在の駅に到達するための時間及び/又は目的地地点近在の駅に到達するために必要な乗換数に基づいて良い。

【 0 0 4 8 】

図 1 に示す通り、トランジットサーバ 1 0 0 は、種々のモジュールを備える。周知の通り、「モジュール」との用語は、指定された機能を提供するために使用されるコンピュータロジックを指すものである。従って、モジュールは、汎用プロセッサを制御するためのハードウェア、ファームウェア、及び/又はソフトウェアにより実装され得る。一実施形態において、モジュールは、記憶装置に記憶され、メモリにロードされ、そして、プロセッサにより実行されたプログラムコードファイルであるか、又は、実体的なコンピュータ読み取り可能な記憶媒体（例えば R A M、ハードディスクもしくは光学的又は磁氣的ディスク）に記憶されたコンピュータプログラム製品（例えばコンピュータにより実行可能な命令群）として提供され得る。更に、当業者たちは、図 1 に示したトランジットサーバ 1 0 0 の別の実施形態が、ここに記載したのとは異なる及び/又は別のモジュールを持ち得ること、そして、その機能性が異なる方法でモジュールに分配され得ることを理解するだろう。

20

【 0 0 4 9 】

一実施形態において、トランジットサーバ 1 0 0 は、クライアント 1 3 1 とネットワーク 1 3 7 経由で通信しており、このネットワーク 1 3 7 は、典型的にはインターネットであるが、次の例示に制限されないが L A N（ローカルエリアネットワーク）、M A N（メトロポリタンエリアネットワーク）、W A N（ワイドエリアネットワーク）、携帯機器、有線又は無線ネットワーク、プライベートネットワーク又はヴァーチャルプライベートネットワークの任意の組み合わせを含んで良い。単一のクライアント 1 3 1 のみが示されているが、膨大な数（例えば数百万）のクライアントがサポートされ、いかなる時でもトランジットサーバ 1 0 0 と通信できる。クライアント 1 3 1 は、或るトリップ向けの指示のためのトランジットサーバ 1 0 0 へのクエリを供給するために、例えばモジラ又はインターネットエクスプローラなどのブラウザ 1 3 3 を実行する。クライアント 1 3 1 は、種々の異なるコンピューティング装置を含む。クライアント装置 1 3 1 の例は、パーソナルコンピュータ、情報端末、携帯電話、移動電話、スマートフォン又はラップトップコンピュータなどである。当業者にとって明白な通り、この発明は上述に列挙した装置に限定されない。

30

40

【 0 0 5 0 】

一実施形態において、トランジットサーバ 1 0 0 は、フロントエンドインターフェース 1 0 1、トランジット情報インタフェース 1 2 9、事前計算モジュール 1 0 3、クエリ解決モジュール 1 1 5、ルート情報データベース 1 2 5、及び、トランジット情報データベース 1 2 7 を備える。フロントエンドインターフェース 1 0 1 はクライアント 1 3 1 からトランジットクエリを受信する。一実施形態において、クライアント 1 3 1 は、或るジャーニーについてのクエリをトランジットサーバ 1 0 0 に送信する。クエリは、フロントエ

50

ンドインターフェース 101 により受信されて、クエリを遂行するために、クエリ解決モジュール 115 と通信される。

【0051】

トランジット情報インタフェース 129 は、例えば航空路、バス路線、或いは、その他任意の公共交通機関トリップを提供する公共トランジット機関など、複数の異なる公共トランジット機関から公共トランジットシステムのトランジットスケジュールに関するトランジット情報を受信する。一実施形態において、トランジットサーバ 100 と通信している各トランジット機関は、例えば、[http://code.google.com/transit/spec/transit\\_feed\\_specification.html](http://code.google.com/transit/spec/transit_feed_specification.html)に記載されたグーグルトランジットフィードスペシフィケーション (GTF S) (グーグルは登録商標) などの指定された形式に従うトランジット情報を供給する。GTF S 形式のトランジット情報を受信することにより、トランジットサーバ 100 は、異なるトランジット機関から均一のデータ形式で情報を受信できるようになる。一実施形態において、トランジット情報は、トランジット車両が運行する暦日付を表す当該トリップのための公共交通機関スケジュールと、該トリップの途中で停止する駅を現す駅情報 (例えば住所) を含む。例えば、トリップにかかる金銭上のコスト、異なるトランジット駅での停止時間など、トリップのその他の属性が、トランジット情報に含まれていてもよい。トリップのためのトランジット情報が受信されたら、トランジット情報インタフェース 129 は、その情報をトランジット情報データベース 127 に記憶する。実際には、トランジット情報データベース 127 は、複数の異なるトランジット機関からのトランジット情報を記憶する。例えば、典型的な都市圏においては、トランジット情報データベース 127 は、公共バスシステムからのバススケジュール情報、地方通勤列車、ライトレール及び長距離列車のための列車スケジュール情報、地下鉄システムのための地下鉄スケジュールなどを記憶する。

【0052】

一実施形態において、事前計算モジュール 103 は、クエリ時間に先行して、トランジット情報に関する事前計算処理を実行する。事前計算モジュール 103 は、トランジット情報データベース 127 内で示された任意の 2 駅間の全最適ルートの乗換パターンのセットを決定するために、トランジット情報データベース 127 に記憶されたトランジット情報を用いる。事前計算モジュール 103 は、また、トランジット情報を用いて、どの駅が別の駅から 1 以上の乗り換えをすることなしに直接到達できるかを決定する。これらの駅は、直通接続される、と呼ばれるものである。事前計算モジュール 103 は、ルート情報データベース 125 内の乗換パターンのセットと直通接続情報を記憶する。

【0053】

クエリ解決モジュール 115 は、特定時間におけるスタート地点から目的地点までの指示のためのユーザクエリを遂行するために、最適トリップルートを決する。クエリ解決モジュール 115 は、最適トリップを決するために、事前計算処理モジュール 103 により供給された前処理された最適乗換パターン及び直通接続を用いる。一般的には、クエリ時間において、クエリ解決モジュール 115 は、クエリを遂行するために、ルート情報データベース 125 からクエリに関連するスタート地点及び目的地点に関連付けられたいくつかの乗換パターンを検索すること、及び、最適トリップを決するために該乗換パターンから具体例を挙げて (インスタンスを作成して) 各トリップを評価することにより、縮減された計算のセットを実行する。特定のクエリに関連する乗換パターンの数が少ないので、乗換パターンの総数には依存せずに、スタート及び目的地点に関連付けられた乗換パターンのみを評価すればよく、最悪の場合のクエリ処理時間でも、概して、とても早い。

【0054】

図 1 に示す通り、事前計算モジュール 103 は、トランジットグラフモジュール 105、グローバル駅選択モジュール 107、乗換パターン計算モジュール 109、乗換パターン圧縮モジュール 111、及び、直通接続モジュール 113 を備える。当業者は、事前計算モジュール 103 がこの発明の別の実施形態においては他のモジュールを含んでよいこ

10

20

30

40

50

とを理解するだろう。次に図 2 を参照すると、事前計算モジュール 103 の事前計算処理の一実施形態は、以下の複数の機能的ステージを備える。

【0055】

200：トランジットグラフ及びトランジットテーブルを生成する。

【0056】

201：グローバル駅を選択する。

【0057】

203：直通接続トリップを決定する。

【0058】

205：乗換パターンを計算する。

10

【0059】

207：乗換パターンを圧縮する。

【0060】

事前計算処理の各機能的ステージは以下に詳しく説明される。

#### 1. トランジットグラフ及びトランジットテーブルの生成

【0061】

事前計算処理の第 1 ステージにおいて、トランジットグラフモジュール 105 はトランジットグラフ及びそれに関連するトランジットテーブルを生成する（ステップ 200）。トランジットグラフは、トランジット情報データベース 127 に記憶された、例えば 1 週間や 1 ヶ月など所与の期間で可能なトリップの全ての表現である。一般に、トリップとは、乗換無しで特定の時間に種々の駅で 1 つのトランジット車両により行われる停車を表す。言い換えれば、トリップとは、所与の時刻に乗換無しで一連の複数の停車（又は停車無し）を行う 1 つのトランジット車両である。一実施形態において、ジャーニーは、起点駅から目的駅へ移動するために要する 1 以上のトリップを表す。すなわち、起点駅から目的駅へのジャーニーは、目的駅に到達するために別のトランジット車両への乗換を要する場合、一連の複数トリップとなるだろう。従って、ジャーニーは、目的駅に到達するために 1 以上のトランジット車両の使用を含むものとなる。トランジットグラフを生成することにより、トランジットグラフモジュール 105 は、所与の機関における異なる公共トランジット機関によるトリップの広範なネットワークを作成できる。トランジットグラフの構造は、以下で更に詳しく説明する。

20

30

【0062】

一般に、トランジットグラフは、既知のトリップ毎に、そのトリップを表現するために、弧によって連結された一連のノードを備える。トリップ内の各ノードは、特定時間における或るトランジット駅での公共トリップ車両によるイベントを表す。有方向の弧は第 1 の駅を表す 1 つのノード（起点ノード）を第 2 の駅を表す 1 つのノード（目的ノード）に連結するものであり、公共トランジット車両が第 1 の駅から第 2 の駅へ移動し、且つ、第 2 の駅で停車する。例えば、カリフォルニア州マウンテンビュー（Mountain View, CA）駅からカリフォルニア州サンフランシスコ（San Francisco, CA）駅へのカルトレイン（Cal Train）経由でのジャーニーはカリフォルニア州パロアルト（Palo Alto, CA）在の駅での 1 つの停止を含む。従って、トランジットグラフは、パロアルト（Palo Alto）駅への到着イベントに有方向の弧により連結されたマウンテンビュー（Mountain View）駅でのイベントを表すノードを含み、パロアルト（Palo Alto）駅への到着イベントが更にサンフランシスコ（San Francisco）駅を表すノードに有方向の弧により連結される。

40

【0063】

次に図 3 を参照すると、トランジットグラフモジュール 105 により実行されるトランジットグラフを生成するための処理のフローチャートの一実施形態が示されている。説明の便宜上、トランジットグラフの生成は、トランジットグラフによって表現された期間内の 1 つのトリップを参照して、説明される。一実施形態において、後述する通り、トランジットグラフ内の特定の弧は、その弧を生成したトリップに関するトリップ情報に関連付けられている。この情報は、時間窓内で関連付けられた旅が有効な日々を表す。従って、

50



ここに示されたトランジットグラフモジュール 105 によって実行される 1 つのトリップのための処理は、全ての既知のトリップの情報を含むトランジットグラフ生成するために、既知のトリップ毎に繰り返される。従ってトランジットグラフモジュール 105 は、ここに示される教示を用いて、トランジット情報データベース 127 内に記憶された公共トランジット情報に表されていたのと同様に新しい期間内で可能な全てのトリップを表すように、該新しい期間のためにトランジットグラフを更新することもできる。

#### 【0064】

先ず、トランジットグラフモジュール 105 は、起点駅を出発し目的駅で終わるトリップのトランジット情報を検索する（ステップ 300）。一実施形態において、トランジットグラフモジュール 105 は、トランジット情報データベース 127 から、或るトリップに関するトリップ情報を検索して、当該トリップをトランジットグラフ内の他のトリップと識別するためにトリップ識別子（ID）を割り当てる。前述の通り、或るトリップのトリップ情報は、そのトリップの操業日スケジュールと、当該トリップ中に停車する時刻と駅の位置を表す駅情報を含む。言い換えれば、トランジットグラフモジュール 105 は、起点駅から目的駅までのトリップのためのルートを決する。このルートは該ルート中で出発及び到着イベントがなされ停車の予定がある駅を表す。

#### 【0065】

トランジットグラフモジュール 105 は、トランジット情報を用いて、トランジット情報に基づくトランジットグラフを作成（つまり生成）する（ステップ 301）。すなわち、トランジットグラフモジュール 105 は、トランジット情報から、トリップのための所与の起点駅及び目的駅及び全ての中間駅を決定し、停車及び／又は乗り換えが出発及び目的駅の間でなされる。トランジットグラフモジュール 105 は、また、各駅に関するスケジュール情報を決定する。この情報はいつ公共トランジット車両が駅に到着及び駅から出発するかを表す。

#### 【0066】

車両の各駅への到着及び／又は各駅からの出発毎に、トランジットグラフモジュール 105 は、それぞれ到着イベントと出発イベントを表すノードを 2 つまで、それぞれの時刻に、トランジットグラフに挿入する。従って、車両が駅に出発／到着しないときは、このトリップではこの駅で出発イベント又は到着イベントが行われないので、このトリップのためには、この駅にノードが生成されない。トランジット情報からトランジットグラフ内のノードが生成されると、各ノードは、いずれかの駅の特定時間におけるトランジット車両に関連付けられる。一実施形態において、トランジットグラフ内のノードは、駅ノード又は車中（機内）ノードのどちらでもあり得る。駅ノードは、トリップをなすためにユーザが乗車（つまり入場）できる駅の車両を表す。対照的に、車中ノードは、ノードに関連付けられた駅に位置されたユーザが現在乗っている公共トランジット車両を表す。トランジットグラフ中のノードは、一実施形態に従い、トリップのルートを示す互いに経由する弧に連結される。一般に、駅で乗車されるトランジット車両を表す乗車弧、人がジャーニーを完遂するためにトランジット車両に乗る別の駅まで歩かなければならない場合の最適徒歩を含む駅で降車されるトランジット車両を表す降車弧、人がトランジット駅で待つことを表す待機弧、及び、トリップの 2 つの停車の間で車両に乗車したままでいることを表すトランジット弧の 4 種類の弧がある。

#### 【0067】

ここで、弧は、駅 A 及び駅 B の乗車及び駅ノードに関連して説明される。下付き文字 “O” は、乗車ノードを表し、下付き文字 “S” は駅ノードを表す。また、記号 “ $\rightarrow$ ”（矢印）は 2 ノード間の弧を表す。以下は、駅間で行われるトランジットイベントを表す説明である。

1.  $A_s \rightarrow B_s$  は、駅 A でのトランジット車両の乗車及び、駅 A と駅 B との間での停車無し
2.  $A_s \rightarrow B_s$  は、駅 A での車両の降車及び、駅 B への徒歩移動に対応する。
3.  $A_s \rightarrow A_s$  は、駅 A での車両の降車及び、同じ駅 A での待機に対応する。

4.  $A_0$ 、 $B_0$ は、駅Aでトランジット車両に乗車すること、及び、少なくとも駅Bに到着するまで同じ車両に乗り続けたままにしていることに対応する。

5.  $A_s$ 、 $A_e$ は、駅Aでの待機及びその時点で駅Aを出発するトランジット車両に乗りしないことに対応する。

#### 【0068】

上述した通り、トランジットグラフモジュール105は、各トリップのトランジット情報を、そのトリップの間に行われるイベントを決定するために、分析する。イベントに基づいて、トランジットグラフモジュール105は、対応するノード（駅又は乗車のいずれも）をトランジットグラフ内に配置して、ノードを弧で連結する。ノードのペアを連結する弧の種類は、連結されるノードが駅ノード又は乗車ノードのいずれであるかに基づく。例えば、トランジット情報が駅Aでトランジット車両に乗りして駅Bまで乗換無しで移動することを表すものとする、トランジットグラフモジュール105は、トランジットグラフ内に駅Aに関連付けられた駅ノードを含め、且つ、トランジットグラフ内に駅Bに関連付けられた乗車ノードを含める。従って、それらノードを連結する弧は乗車弧となる。別の例では、乗車ノードが駅ノードに連結されたものとする、それらノードを連結する弧は降車弧である。

#### 【0069】

待機弧を表すために、トランジットグラフには時間の概念が表現される。ここで図4を参照すると、トランジットグラフの一部が示されている。この図において、トランジットグラフ400は、Y軸及びX軸に関して組織されたノードを示す。トランジットグラフのY軸は時間を表し、X軸は駅を表す。ここで、X及びY軸はトランジットグラフ内の時間の使用を説明するために便利のように説明されているのであり、ルート情報データベース125に記憶されたトランジットグラフの現実の実施においては、X又はY軸を含まず、また、グラフのノードも図4に示すような空間的な方法で配置されるのではないことは、理解される。

#### 【0070】

この図において、Y軸は1日の時間表現である。トランジットグラフ内の時間の概念を説明するために、X軸に表された駅毎に、トランジットグラフモジュール105は、トランジットグラフに、Y軸に沿って配列されるように描かれた一連のノードを追加する。駅に関連付けられた一連の各ノードは、そのノードに関連付けられた特定の時刻に駅で行われるトランジットイベントを表す。

#### 【0071】

トランジットグラフに示す通り、駅 $S_1$ は、異なる時間間隔の一連のノードを含む。“S”が付されたノードは駅ノードであり、“O”が付されたノードは乗車ノードである。所与の駅に関連付けられた各駅ノードは、特定の時刻にその駅で行われる何らかのイベントに関連付けられている。一実施形態において、全ての駅の駅ノードは、その特定の駅で待つ人を表すために、時間に基づいて順次リンクされている。例えば、トランジットグラフにおいて、 $S_1$ の駅ノード群は時間内で時順次に連結されており、 $T_1$ から $T_4$ 間の期間に駅 $S_1$ で人が待つという概念を表している。一実施形態において、トランジットグラフモジュール105は、所与の駅の各駅ノードを弧経由で次の出発イベントが行われる所与の駅に連結することにより、トランジットグラフに示された情報からトリップを推測する。

#### 【0072】

ここで、説明の簡易化のために、乗車弧、降車弧及び徒歩弧は時間に言及せずに、説明するものとする。トランジットグラフモジュール105が、駅B及びCで途中停車するが乗換無しで、起点駅Aから目的駅Dまでを表すトランジット情報からのジャーニーを表現しているものと仮定する。なお、このシナリオでは、1つのトリップがジャーニーとみなされている。このジャーニーはこの表記法を使い、次のように表現される。

$A_s$ 、 $B_0$ 、 $C_0$ 、 $D_0$ 。

#### 【0073】

10

20

30

40

50

上記の表記法につき、 $A_{\circ}$ 、 $B_{\circ}$ は、駅Aで乗車され、駅Bに移動する車両を表す。 $B_{\circ}$ 、 $C_{\circ}$ は、駅Bで乗車すること、そして駅Cに車両が到達するまで車両に居ることを表す。 $C_{\circ}$ 、 $D_{\circ}$ は、 $B_{\circ}$ 、 $C_{\circ}$ と同様なシナリオであるが、駅Dが目的駅を表し、乗車ノードが行き先への到着を表している。

#### 【0074】

トランジットグラフ内のジャーニーを表現するために、トランジットグラフモジュール105は、各到着ノード／出発イベントをノードとして表現する。しかし、前述した通り、各ノードは、駅ノード又は乗車ノードのいずれにも対応する。トランジットグラフモジュール105は、トリップ内の駅ペアのためにどのノードを使用するかを、駅間で実行されるトランジットイベントに基づいて、決定する。前述した例を用いると、トランジット車両は駅Aで乗車されて、トランジット車両は乗換無しで駅Bに移動するので、トランジットグラフモジュール105は、駅Aでのイベントを駅ノードとして表し、駅Bでのイベントを乗車ノードとして表すだろう。次に、車両は駅Bから出発し、駅Cでのみ停車し乗換無しで、目的駅Dに移動するので、駅C、Dでのイベントは、トランジットグラフモジュール105により、乗車ノードとして表されるだろう。

#### 【0075】

別の例として、駅Cで乗換が行われることを除いて上述と同じジャーニーを仮定する。従って、駅Cで乗換が行われるので、このジャーニーには2つのトリップを要する。このジャーニーはトランジットグラフにおいて以下のように表されるだろう。

$A_{\circ}$ 、 $B_{\circ}$ 、 $C_{\circ}$ 、 $C_{\circ}$ 、 $D_{\circ}$ 。

#### 【0076】

上記の表記法を使うと、 $B_{\circ}$ 、 $C_{\circ}$ は、駅Bから駅Cに車両が到着することを表す。 $C_{\circ}$ 、 $C_{\circ}$ は、駅Cの車両が出発したこと、車両から離れ、駅Cから出発する別の車両に乗るために、駅Cで待つ人を表す。

#### 【0077】

前述の例を続けると、トランジットグラフ内において、駅A及びBでのイベントの表現は、上記説明により、駅Aでのイベントについては駅ノードであり、駅Bでのイベントについては乗車ノードとなる。トランジットグラフモジュール105は、駅Cについて、2つのノード経由の駅Cのトランジットイベントを、駅ノード及び乗車ノードと表すだろう。トランジットグラフモジュール105は、駅Bの乗車ノードから、駅Cでの到着イベントとそのときにトランジット車両に居ることを表すCの乗車ノードに弧を連結する。トランジットグラフモジュール105は、駅Cの乗車ノードから、駅Cで車両から出てその駅で次の車両を待っていることを表すCの駅ノードに別の弧を連結する。

#### 【0078】

更にトランジットグラフ内の時間の使用を説明するために、図4に戻る。先に述べた通り、この図では、Y軸は1日の時間表現である。X軸に表された駅毎に、トランジットグラフモジュール105は、トランジットグラフに、Y軸に沿って配列されるように描かれた一連のノードを追加する。駅に関連付けられた一連の各ノードは、そのノードに関連付けられた特定の時刻に駅で行われるトランジットイベントを表す。トランジットグラフモジュール105は、ノード毎に時刻情報を決定するために、記憶されたトランジット情報からのスケジューリング情報を使用する。従って、トランジットグラフモジュール105は、駅毎に、車両が駅に到着するノードから出発する時間内の全ての事例のためのノードを含む。

#### 【0079】

一実施形態において、トランジットグラフ内のトリップのノードに連結される各弧は、トランジットグラフモジュール105により決定されるコストに関連付けられている。各コストは、弧に連結されている起点ノードから目的ノードに到達するための種々の要素に関する価格を表しており、それは金銭的成本を含むか又は含まない。トランジットグラフモジュール105は、各トリップに関連付けられた記憶されたトランジット情報を、弧のコストを決定するために、分析する。

## 【 0 0 8 0 】

一実施形態において、或る弧のためのコストは、多次元的コスト関数である。第1次元のコスト関数は、時間に基づく。トランジットグラフモジュール105は、目的ノードへの到達時間と起点ノードからの出発時間の間の時間の違いを決定するために、トランジット情報を分析する。目的ノードに関連付けられた駅に到達するための総時間は、コスト関数の第1次元として、目的ノードと起点ノードに連結された弧に関連付けられている。一実施形態において、時間長は秒単位で表されるが、別の実施形態においては別の時間表現も使用できる。

## 【 0 0 8 1 】

第2次元のコスト関数は、一実施形態に従えば、金銭的成本、乗換総数、及び/又は徒歩のコストを含む種々の要素に基づく。これら種々の要素の集合は、総コストに含まれるペナルティになる。トランジットグラフモジュール105は、トリップの弧の脚に関連付けられたためのペナルティを決定する。

10

## 【 0 0 8 2 】

一例として、駅で乗換が行われたとき、弧のペナルティは増加し得る。ジャーニーの金銭的成本は、また、ペナルティに因数分解される。駅に到達するためのより大きい金銭的成本は、ジャーニーが1以上のトランジット車両の使用を含むときには、より高いペナルティとなる。また、トランジット車両を乗り換えるために或る駅から別の駅まで歩く距離は、トリップのコストに因数分解される。ペナルティにおける異なる要素の重み付けは、トランジットサーバ100のシステム管理者によって調整される。

20

## 【 0 0 8 3 】

トランジットグラフモジュール105は、トランジット情報データベース127の情報からのジャーニーを表すノードをトランジットグラフ内に生成するために、上記の処理を実行する。トランジットグラフモジュール105は、期間内で有効な記憶されたジャーニー毎に、トランジットグラフがトランジットグラフ内に表された全てのトリップを表すまで、処理を繰り返す。

## 【 0 0 8 4 】

一実施形態において、日毎に新しいトランジットグラフを作成するのではなく、トランジットグラフの駅毎に、トランジットグラフモジュール105は、その日にその駅で行われる出発イベント毎に駅ノードを作成する。トランジットグラフモジュール105は、待機弧経由で駅ノードを連結する。その日の最後の駅ノードは、その日の最初の駅ノードに戻る待機弧409に割り当てられる。これは、前日の夜から、その翌日にその駅で行なわれる次の出発イベントまで、駅で待機することを表す。トランジットグラフ内の乗車ノードに関しては、各乗車ノードは、1以上のトリップに関連付けられており、従って、本質上、所与の期間内の日数内で1以上のトリップが有効なとき（つまり操業しているとき）を表す情報を含む。

30

## 【 0 0 8 5 】

図4において、トランジットグラフ400は、一実施例に従う、時間と公共トランジット駅の関数である。トランジットグラフ400は、グラフ内で弧405によって連結されたノードとして、種々の時刻 $T_1 \sim T_8$ で種々の駅 $S_1$ 及び $S_6$ の間で公共トランジット車両により行われるイベント（つまり到着又は出発）を表す。トランジット情報を用いて、トランジットグラフモジュール105は、3つのジャーニーを含むトランジットグラフ400を作成している。トランジットグラフ400内の各弧は、起点ノードが表す駅から目的ノードが表す駅まで移動するためのコスト407を含む。

40

## 【 0 0 8 6 】

例えば、トランジットグラフ400に描かれた1つのジャーニーは、時刻 $T_4$ における駅 $S_1$ から時刻 $T_8$ における駅 $S_6$ 間でのトリップである。トランジットグラフ400に示す通り、時刻 $T_4$ における駅 $S_1$ のノードは、ノード内に位置する“S”により記されたとおり駅ノードである。時刻 $T_5$ における駅 $S_3$ のノードは、ノード内に位置する“O”により記されたとおり乗車ノードである。同様なノード指定が、トランジットグラフ400内

50

の残りのノードにも使用される。

【 0 0 8 7 】

ジャーニーによって示されたルートは、時刻  $T_4$  に駅  $S_1$  から出発し、続いて時刻  $T_5$  に駅  $S_3$  で停車し、時刻  $T_6$  に駅  $S_4$  での別の停車により後続される車両を表す。時刻  $T_6$  の駅  $S_4$  から、時刻  $T_8$  の駅  $S_6$  にて、車両は行き先に到達する。なお、ジャーニーは、また、時刻  $T_4$  から時刻  $T_5$  の間に駅  $S_2$  を通過する車両を表しているが、車両が駅  $S_2$  に止まらないので時刻  $T_4$  から時刻  $T_5$  の間の駅  $S_2$  を表すノードは作成されない。

【 0 0 8 8 】

また、トランジットグラフ 4 0 0 は、また、時刻  $T_1$  における駅  $S_2$  から時刻  $T_8$  における駅  $S_6$  へのジャーニーを含み、同様に、時刻  $T_1$  における駅  $S_3$  から時刻  $T_8$  における駅  $S_6$  へのジャーニーも含む。なお、時刻  $T_1$  に駅  $S_2$  から出発する車両は時刻  $T_4$  に駅  $S_1$  から出発する車両と同じ時刻に駅  $S_3$  に到着する。一実施形態において、時刻  $T_5$  における駅  $S_3$  の単一イベントは、両方のイベントを表す。

【 0 0 8 9 】

更に、トランジットグラフ 4 0 0 は、車両が駅に到着するノードから出発する時間内のすべての事示す一連の駅  $S_1$  のノード 4 0 3 を例示している。また、時刻  $T_4$  だけは、駅  $S_1$  から出発する車両を描いており、説明の便宜上、駅  $S_1$  のノードからの出発を描く弧が省略されていることを理解されたい。駅  $S_1$  の最後の駅ノードを表すノードは、駅  $S_1$  が運営している日の先頭の時刻を表す駅ノードに戻るようになっている。

【 0 0 9 0 】

図 3 に戻ると、トランジットグラフが作成されると、トランジットグラフモジュール 1 0 5 は、そのトランジットグラフに基づいて一連のトランジットテーブルを作成する。一実施形態において、トランジットテーブルは、トランジットグラフの弧とノードに関連付けられた情報を表す。トランジットグラフは、本質的には、トランジットグラフ内の各ノードと弧についての情報及びトランジットグラフに関するその他の情報を表すトランジットグラフのテーブル表現である。トランジットグラフモジュール 1 0 5 は、トランジットグラフ内の情報を、種々のトランジットテーブルに記憶する。トランジットテーブルは更なる詳細を後述される。なお、一実施形態において、トランジットグラフを生成し、それに続いてそのトランジットグラフからトランジットテーブルを作成するのではなく、トランジットテーブルはトランジット情報から直接作成される。

【 0 0 9 1 】

トランジットグラフモジュール 1 0 5 は、トランジット情報に基づいて、トリップマスクのセットを作成する（ステップ 3 0 3）。一実施形態において、トリップマスクのセットは、全ての可能なトリップをリスト表示し且つ何時そのトリップが有効かを示すテーブルとして表される。テーブルの各行はトリップのトリップマスクを表す。各トリップマスクは、車両がトリップマスク内に示された特定のトリップをする期間内の日付を表す。一実施形態において、トリップマスク内の期間は 6 0 日である。別の実施形態では別の期間を使用できる。トリップマスクの一例は以下の通りである。

【表 1】

トリップマスクの例

トリップID	09/6/8	09/6/9	09/8/7	09/8/8
$T_1$	0	1	1	1
$T_2$	0	0	0	0
$T_3$	0	0	1	1
$T_N$	0	1	1	0

【 0 0 9 2 】

テーブル内の各行は特定のトリップに関連付けられている。各欄は選択された期間内の日又は日付を表す。一実施形態において、トリップマスク内の所与の日の“ 1 ”はその日にトリップが有効であることを示し、“ 0 ”はその日にトリップが有効でないことを示す

。別の実施形態では、トリップが有効か無効かを示すために別の表記方法が使用されうる。トリップマスクが作成されたら、トランジットグラフモジュール 105 はそのトリップマスクをトランジット情報データベース 127 に記憶する。

#### 【0093】

上記の例において、トリップマスクはトリップ  $T_1$ 、 $T_2$  から  $T_N$  をリスト表示している。残りの欄は、特定の開始日（例えば 2009 年 6 月 8 日）から末日（例えば 2009 年 8 月 8 日）の間の日付を表す。各日付に対して、トリップ毎にその日にトリップが有効か又は無効化を示す“1”又は“0”のどちらかがリスト表示される。例えば、2009 年 6 月 8 日には、トリップ  $T_2$  のみが有効であり、トリップマスクにリスト表示された他の全てのトリップは使用できない。次の日は  $T_1$  と  $T_N$  だけが使用できる。

#### 【0094】

トリップマスクが作成されると、トランジットグラフモジュール 105 は、ノードテーブルを作成する（ステップ 305）。一実施形態において、トランジットグラフモジュール 105 は、トランジットグラフに基づいてノードテーブルを作成する。ノードテーブルは、各ノードの識別子（ID）と各ノードに関連付けられた駅の駅 ID を指定する。ノードテーブルは、更に、各ノードに関連するノード種類を指定する。ノード種類は、そのノードが駅ノードか又は乗車ノードかを示す。ノードテーブルは、また、各ノードに関連付けられたイベントの時刻も含む。イベントは、ノード種類に応じて到着又は出発イベントになるだろう。最後に、ノードテーブルは、何時ノードが有効化を示すためにトリップマスクを指すトリップ ID を含む。なお、一実施形態において、駅ノードは特定のトリップの一部でない駅で単に待っている人に関連付けられるので、乗車ノードのみが、トリップ ID に関連付けられていてもよい。ノードテーブルが作成されたら、トランジットグラフモジュール 105 は、ノードテーブルをトランジット情報データベース 127 に記憶する。下記は、ノードテーブルの一例を示す。

#### 【表 2】

ノードテーブルの例

ノードID	駅ID	イベント時刻	ノード種類	トリップID
$N_1$	A	8:00	乗車	$T_8$
$N_2$	B	16:00	駅	無効
$N_3$	C	2:00	駅	無効
$N_4$	A	12:00	乗車	$T_{345}$
$N_N$	D	20:00	乗車	$T_{50}$

#### 【0095】

示されたテーブルは、5 つの欄：ノード ID、駅 ID、イベント時刻、ノード種類及びトリップ ID、を持つ。ノード ID の欄は、トランジットグラフの全てのノード  $N_1$  から  $N_N$  のリスト表示である。各ノードは駅 ID、イベント時刻、ノード種類及びトリップ ID に対応付けられている。例えば、ノード  $N_1$  は、駅 A に関連付けられている。テーブルは、ノード  $N_1$  が 8 時 00 分のイベント時刻を持つ乗車ノードであることを示す。従って、その乗車ノードは、車両から降車する人無しで駅 A に到着するトランジット車両、又は、駅 A に到着し駅 A で車両から人が降りるトランジット車両を表し得る。ノードテーブルは、また、乗車ノードが、そのノードに関連付けられた日付が有効であることを示すトリップマスクに戻るトリップ ID  $T_8$  に関連付けられていることを示す。別の例として、ノード  $N_3$  は、駅 C に関連付けられており、駅ノードである。ノード  $N_3$  に関連付けられたイベントは、2 時 00 分に行われる。ノードが駅ノードなので、2 時 00 分に行われるイベントは、駅 C での待機イベント又は乗車イベントである。前述の通り、駅ノードはトリップ ID へ戻すことができない。

#### 【0096】

ノードテーブルが作成された後、トランジットグラフモジュール 105 は、弧テーブルを作成する（ステップ 307）。弧テーブルは、トランジットグラフ内の各弧に関連する

情報を指定する。一実施形態において、弧テーブルは、各弧のペナルティとともに各弧の起点ノードIDと目的ノードIDを指定する。弧自身は明白には識別されない。むしろ、弧は、弧によって連結された特定のノードに基づいて、推測される。トランジットグラフ内の各ノードは特有のノードIDに関連づけられているので、弧は、或る弧によって連結されているノードのノードIDに基づいて決定され得る。弧テーブルは、また、起点及び目的ノードにより推測された各弧に関連づけられたペナルティを表す。弧テーブルが作成されると、トランジットグラフモジュール105は、弧テーブルをトランジット情報データベース127に記憶する。下記は、弧テーブルの一例を示す。

【表3】

弧テーブルの例

起点ノードID	目的ノードID	ペナルティ
N <sub>1</sub>	N <sub>24</sub>	100
N <sub>50</sub>	N <sub>32</sub>	150
N <sub>8</sub>	N <sub>10</sub>	200
N <sub>24</sub>	N <sub>94</sub>	50

## 【0097】

テーブルは、起点ノードID、目的ノードID及びペナルティの、3つの欄を含む。起点ノードIDと目的ノードIDの欄は、どのノードが弧によって連結されているかを推測するトランジットグラフ内の全ての起点及び目的ノードのペアをリスト表示する。例えば、(N<sub>1</sub>, N<sub>24</sub>)の起点ノードIDと目的ノードIDは、これら2つのノードが弧によって連結されていることを示す。この例では、その弧は、100のペナルティを持つ。

## 【0098】

弧テーブルが作成された後、トランジットグラフモジュール105は、駅位置テーブルを作成する(ステップ309)。駅位置テーブルは、トランジットグラフ内の各駅の地理的位置を示す。一実施形態において、各駅の地理的位置は駅の緯度及び経度座標により表される。別の例として、例えば駅の住所など、地理的位置を表す別の方法が使用され得る。駅位置テーブルが作成されると、トランジットグラフモジュール105は、駅位置テーブルをトランジット情報データベース127に記憶する。下記は、駅位置テーブルの一例を示す。

【表4】

駅位置テーブルの例

駅ID	地理的位置
A	(33.58, -85.85)
B	(41.48, -120.53)
C	(38.22, -122.28)
D	(37.37, -121.92)

## 【0099】

上記の例では、駅位置テーブルは、トランジットグラフ内の全ての駅をリスト表示する駅ノードID欄を含む。駅位置テーブルは、また、各駅IDの地理的位置を緯度及び経度座標でリスト表示する地理的位置欄を含む。例えば、駅Dは、カリフォルニア州サンノゼ地域に対応する北緯37度37分、西経121度92分の地理的位置を持つ。

## 【0100】

一実施形態において、トランジットグラフモジュール105は、前述の弧テーブルの作成を援助するために、近在駅テーブルを作成する(ステップ311)。具体的には、トランジットグラフモジュール105は、乗換弧を作成するために近在駅テーブルを用いる。前述した通り、乗換弧(人がその人が到着した駅で乗り換えるか、又は、別の駅へ歩くかに応じて、徒歩を含むか若しくは含まない)は、各乗車ノードから、その駅自身を含め人が到着した駅に近在の全ての駅で最も早く到達するイベントへ、作成される。乗換弧をメ

10

20

30

40

50

モリに記憶するよりもむしろ、近在駅テーブルが記憶され、乗換弧の要求があり次第、乗換弧が作成される。テーブル内の駅毎に、トランジットグラフモジュール 105 は、記憶されたトランジット情報を用いて、所与の駅からのラジアル距離内にある全ての駅を決定する。ラジアル距離内の駅は、或る駅の“近在”又は“ローカル”とみなされる。一実施形態において、ラジアル距離は 2 マイルであってよく、また、別の実施形態ではその他の距離であってよい。従って、システム管理者によって決定されるトランジットサーバ 100 のラジアル距離に応じて、所与の駅は近在の駅を持たないかもしれない。近在駅テーブルが作成されると、トランジットグラフモジュール 105 は、近在駅テーブルをトランジット情報データベース 127 に記憶する。下記は、近在駅テーブルの一例を示す。

【表 5】

近在テーブルの例

駅ID	近在駅IDのリスト
A	{(A, 60), (D, 60), (Q, 60), (R, 120), (Z, 120), (Y, 3200)}
B	{(F, 120)}
C	無し
D	{(D, 60), (M, 60), (E, 120), (K, 180)}

## 【0101】

テーブルは、駅 ID と近在駅 ID のリストとの、2 つの欄を含む。駅 ID 毎に、近在駅 ID のリストは、“近在”駅とみなされる駅と、駅から近在駅に乗り換えるのに必要な最小の時間長を示している。すなわち、近在駅 ID のリストは、指定されたラジアル距離内にあるこれらの駅と、近在駅に到達するのに必要な時間を示す。例えば、駅 A は駅 D、駅 Q、駅 R、駅 Z 及び駅 Y に近い。駅 A から駅 D に乗り換えるには、例えば最小時間 60 秒が乗換実施のために必要である。対照的に、駅 C は近在駅が何もない。更に、一実施形態においては、或る駅はそれ自身の近在駅 ID のリストに含まれてよいことに留意されたい。これは、単一の駅において或るトランジット車両から別のトランジット車両に乗り換えるために必要となる最小時間長（すなわち安全バッファ）を示す。例えば、駅 A にとって、その近在駅 ID のリストは、駅 A で乗換を完遂するには最小時間 60 秒を要することを示す。なお、図示の例では駅 B のための近在駅 ID のリストに示すように、その駅で乗換が行われれば、全ての駅がそれら自身の近在駅 ID のリストに含まれるのではないことに留意されたい。

## 2. グローバル駅選択

## 【0102】

図 2 に戻ると、事前計算処理の第 2 ステージ（随意）において、グローバル駅（グローバル駅）選択（GGS）モジュール 107 はグローバル駅を選択（つまり決定）する（ステップ 201）。一実施形態において、グローバル駅は、長い接続（コネクション）の間に、乗換が行われるであろう駅である。すなわち、グローバル駅は、長距離に広がるトリップにおいて乗換が頻繁に行われる駅である。GGS モジュール 107 は、或る駅で乗り換えるトランジットルートの数に基づいて、又は、時間独立ヒューリスティックに基づいて、グローバル駅を選択する。各実施例は更に詳しく後述されるだろう。

## 【0103】

次に図 5 A を参照すると、或る駅で乗り換えるルートの数に基づいてグローバル駅を計算するために GGS モジュール 107 によって実行される方法のフローチャートが示されている。GGS モジュール 107 は、トランジットグラフを入力として受信する（ステップ 505）。GGS モジュール 107 は、トランジットグラフから、各駅で行われる乗換の数を決定する（ステップ 500）。GGS モジュール 107 は、トランジットグラフにおいて各駅で車両乗換が行われる回数を決定するために、そのトランジットグラフを分析する。GGS モジュール 107 は、各駅で行われる乗換の数に基づいてグローバル駅を識

10

20

30

40

50



別する（ステップ501）。一実施形態において、グローバル駅とみなされるために或る駅で行われるべき乗換の最小数を指定するように、乗換閾値が適用される。GGSモジュール107は、乗換閾値に従い各駅をフィルタし、残った駅がグローバル駅とみなされる（ステップ503）。

#### 【0104】

次に、図5Bを参照すると、時間独立ヒューリスティックに基づいてグローバル駅を決定するためのGGSモジュール107の処理が示されている。GGSモジュール107は、トランジットグラフを入力として受信する（ステップ505）。GGSモジュール107は、次に、トランジットグラフ内の各駅の駅ノードを、単一のノードにグループ化する（ステップ507）。前述の通り、各駅は車両が駅に到着又は駅から出発する種々の時間を表す1以上のノードを持つ。これらノードの、部分集合は駅ノードである。GGSモジュール107は、各駅の駅ノードを、その駅のための駅ノードグループを形成するようにグループ化する。例えば、図4に示す駅 $S_1$ の8つのノードは、それらが全て駅ノードとすると、8つのノードからなる駅ノードグループを形成するように、グループ化されるだろう。

#### 【0105】

GGSモジュール107は、それから、乗車ノードをグループ化する（ステップ509）。一実施形態において、GGSモジュール107は、乗車ノードグループを形成するために、トランジットグラフ内で、同じ先行駅のシークエンスを持ち乗車ノードによって表される車両間で追い越しが無い乗車ノードをグループ化する。すなわち、GGSモジュール107は、その一日のうちの類似するトリップをグループ化する。一実施形態において、類似するトリップは、互いに追い越さず、トリップの間は同じ駅のシークエンスを辿るトリップである。GGSモジュール107は、トリップのグループ毎に、そのグループの各トリップの第1の乗車ノードを、或る乗車ノードグループにグループ化する。なお、乗車ノードグループは、そのグループが類似するトリップからできているので、1つの駅の乗車ノード群を表すことに留意されたい。そして、GGSモジュール107は、そのグループの各トリップの第2の乗車ノードを、第2の乗車ノードグループにグループ化する。そして、全ての乗車ノードがグループ化されるまで、これを繰り返す。

#### 【0106】

GGSモジュール107がトランジットグラフの駅ノードと乗車ノードをグループ化したら、GGSモジュール107は、グループ間の弧を追加する（ステップ511）。一実施形態において、GGSモジュール107は、トランジットグラフのグループ化されていない表現（リプレゼンテーション）を分析し、トランジットグラフ内の異なる駅ノードグループ又は乗車ノードグループに属する2つのノード間に少なくとも1つの弧が存在するかどうかを決定する。GGSモジュール107は、トランジットグラフ内の2つのノード間に弧が存在していた場合、異なるグループ間に弧を追加する。

#### 【0107】

GGSモジュール107は、トランジットグラフ内のグループ化された表現（リプレゼンテーション）における弧毎に、コストを決定する（ステップ513）。一実施形態において、2グループ間の弧毎に、GGSモジュール107は、トランジットグラフから、トランジットグラフ内の弧経路で連結された各ノードのペアであって、ペアになっている各ノードが連結されている2つの異なるグループに属するものを決定する。本質的に、GGSモジュール107は、トランジットグラフから、互いに連結されたグループのペアに属し、且つ、連結されたノードのペアのセットを作成する。GGSモジュール107は、そのセットから、どのノードのペアが最小コストに関連づけられているかを決定し、そして、グループ化されたトランジットグラフ内の2つのグループ間の弧に最小コストを割り当てる（ステップ515）。

#### 【0108】

一実施形態において、トランジットグラフのグループ化された表現は、トランジットグラフ内で時間を持っていたノードが時間相を持たないノードグループにフラット化されて

10

20

30

40

50

いるので、時間独立（時間に依存しない）とされる。トランジットグラフのグループ化された表現における各ノードグループが時間を横断して類似するノードを表すので、結果として得たトランジットグラフは、ノードグループがもう特定の時間に関連付けられていないため、時間独立（時間に依存しない）とみなされる。

#### 【0109】

次に、GGSモジュール107は、コストの点での最短経路を決定するために、毎回ランダムな駅グループをソースとして取り（すなち、この駅ノードグループに属する全ての結ばれたノードが最短経路探索のソースとなる）ダイクストラのアルゴリズムを繰り返し適用する（ステップ517）。なお、ダイクストラのアルゴリズムの動作は、周知技術であるため、ここでは説明しない。一実施形態において、GGSモジュール107は、時間

10

#### 【0110】

GGSモジュール107は、ダイクストラのアルゴリズムにより得たツリー内の各駅ノードグループのためのスコアを決定する（ステップ519）。一実施形態において、GGSモジュール107は、スコア付けされる所与の駅ノードグループより下流のサブツリー内の駅ノードグループの数に基づいて、当該所与の駅ノードグループにスコアを割り当てる。GGSモジュール107は、ダイクストラのアルゴリズムにより得た、異なるツリーからの各駅ノードグループのスコアを合計することにより、各駅ノードグループのための

20

#### 【0111】

各駅ノードグループのための総スコアが生成されたら、GGSモジュール107は、総スコアに基づいてグローバル駅を選択する（ステップ521）。一実施形態において、GGSモジュール107は、小さい割合の駅ノードグループしか選択しない。GGSモジュール107は、総スコアに基づいて降順にグローバル駅を配列して、グローバル駅の上から1パーセントを選択し、グローバル駅のセットを得る（ステップ503）。

### 3. 直通接続の事前計算

#### 【0112】

図2に戻ると、事前計算処理の第3ステージにおいて、直通接続モジュール113は、トランジットグラフを用いて、直通接続情報を事前計算する（ステップ203）。一実施形態において、直通接続トリップは、起点駅から目的駅まで乗換の無いトリップである。また、トリップは起点駅から目的駅までの間に中間駅での停車も含んでいてもよく、中間駅で乗換が行われないならばそのトリップは直通接続トリップとみなされる。トランジットグラフに関して、直通接続トリップは、トランジットグラフ内の駅ノードから出発して、トランジットグラフ内の目的駅に関連付けられた乗車ノードに到着する、乗換無しのトリップである。

30

#### 【0113】

次に、図6を参照すると、直通接続モジュール113によって実行される直通接続を計算するための方法のフローチャートである。一実施形態において、直通接続モジュール113は、トランジットグラフを入力としてとる（ステップ505）。直通接続モジュール113は、トランジットグラフを決定（すなわち捜し出す）して（ステップ601）、トリップを識別する。すなわち、直通接続モジュール113は、その日の所与の時刻に複数駅のセットを通過する単一のトランジット車両を現すトランジットグラフ内の経路を識別する。

40

#### 【0114】

直通接続モジュール113は、それから、トランジット路線を決定する（ステップ603）。一実施形態において、トランジット路線は、起点及び目的駅を連結する複数トリップのセットであり、路線内の各トリップが同じ順番又はシーケンスで同じ駅を通過する。すなわち、トランジット路線内の各トリップは、同じ起点駅と、同じ目的駅と、該起点

50

駅と該目的駅の間に位置し、そのトリップに関連付けられたトランジット車両が停車する、同じ中間駅のセットとを持つ。一実施形態において、トランジット路線の各トリップは、また、同じペナルティを持っている。従って、2つトリップが同じ起点駅と同じ目的駅とを持ち、同じ駅を同じ順番で通過するとしても、その2つのトリップのペナルティが異なっていれば、それらトリップは同じトランジット路線には含まれないだろう。なお、中間駅での停車がなければ、トランジット路線の或るトリップは、起点駅と目的駅のみを含むだろう。トランジット路線内のトリップは、起点駅、出発駅、及び/又は、トリップ中に停車がなされる中間駅からの出発/到着の時刻によって互いに区別される。

#### 【0115】

一実施形態において、トランジット路線内のトリップ毎に、直通接続モジュール113は、トリップ内の各駅にポイント性質に関連付ける。各駅は、トリップ中のポイントと、そのポイントで行われるイベントを表すポイント性質とを考慮される。直通接続モジュール113は、各駅ノードにポイント性質“乗車ポイント”を割り当て、各乗車ノードにポイント性質“降車ポイント”を割り当てる。直通接続モジュール113は、各ノードにポイント指標(インデックス)を割り当てて、下記の2つの特質に基づき各ノードに関連付けられたポイントを使用して、標準的順番でトリップを整列する。2つの特質は次の通り

1. トリップ中のより高い指標を持つポイントからより低い指標を持つポイントへの経路がない。

#### 【0116】

2. より低い指標を持つ乗車ポイントからより高い指標を持つポイントへの経路が常にある。

一実施形態において、直通接続モジュール113は、また、各ノードに累積ペナルティを割り当てる。累積ペナルティは、トリップにおける乗車ポイントと降車ポイントの間の総ペナルティを表す。累積ペナルティは、ノード間のペナルティを決定するためのノードペアに連結された弧の分析の必要性を回避するために、各ノードに追加される。むしろ、直通接続モジュール113は、降車ポイントの累積ペナルティから乗車ポイントの累積ペナルティを、引き算することにより、累積ペナルティを決定できる。直通接続モジュール113が上述した情報をトリップに含ませたら、そのトリップ情報が記憶される。したがって、直通接続トリップに関連付けられたトランジットグラフ内の各ノードは、一意のトリップに対応付けられるだろう。各トリップは、ノードに関連付けられたトランジット路線を示す路線ID、トランジット路線からのトリップを示すトリップID、及び、ポイント指標を含む。

#### 【0117】

次に図7を参照すると、起点駅を示す駅ノード(例えば $A_S$ )と、弧701により連結された複数の乗車ノード(例えば $B_O$ 、 $D_O$ 、 $E_O$ 及び $F_O$ )を含むトランジット路線の表現の一例が示されている。各ノードは、直通接続モジュール113によって割り当てられたポイント指標703に関連付けられている。図7は、また、各弧に割り当てられたペナルティも描いており、そのペナルティは当該弧により連結されたノード間のペナルティを表す。最適トリップについての情報を含むトランジット路線が以下に示されている。トランジット路線は、直通接続データベース125における直通接続トリップの記憶構造を描いている。

【表 6】

トランジット路線の例

T グラフ 駅ID	A	B	D	E	F
ポイント指標	0	1	2	3	4
ポイント性質	乗車	降車	降車	降車	降車
累積ペナルティ	0	1	6	7	11
トリップ23の時刻	8:00	8:15	8:30	9:45	10:00
トリップ55の時刻	9:00	9:15	10:30	9:50	10:15
トリップ60の時刻	11:00	11:15	12:00	12:30	13:00
トリップ100の時刻	11:30	12:00	13:00	13:15	13:30
トリップ18の時刻	12:00	12:15	12:45	13:00	15:00

10

## 【 0 1 1 8 】

トランジットグラフから全てのトランジット路線が決定されたら、直通接続モジュール 1 1 3 は、トランジット路線を分析する（ステップ 6 0 5）。直通接続モジュール 1 1 3 は、トランジット路線から直通接続テーブルを決定する。直通接続テーブルは、整列された駅のペアと、そのペア内の起点駅での乗車とそのペア内の目的駅での降車とを示すペア毎の路線のセットとのリストである。一般に、直通接続モジュール 1 1 3 は、トランジットグラフ内の可能な駅のペアのための直通接続を決定する（ステップ 6 0 7）。下記は直通接続テーブルの一例である。

20

【表 7】

直通接続テーブルの例

駅ペア	路線 I D
(A, B)	L <sub>1</sub> , L <sub>4</sub> , L <sub>5</sub> 及びL <sub>9</sub>
(A, C)	L <sub>2</sub> 及びL <sub>3</sub>
(B, A)	無し
(B, C)	L <sub>4</sub> , L <sub>5</sub> 及びL <sub>9</sub>
...	...

## 【 0 1 1 9 】

上記直通接続テーブルの例は、駅ペアのリストと、ペア内の起点駅から目的駅までのトリップを含むトランジット路線を示す路線 I D の各ペア毎のリストを描いている。例えば、駅ペア (A, B) は、L<sub>1</sub>、L<sub>4</sub>、L<sub>5</sub>及びL<sub>9</sub>の路線 I D に関連付けられている。これは、これら路線 I D に関連付けられたトランジット路線が、人に駅 A で乗車して駅 B で降車させるトリップを持つことを示す。別の例では、駅ペア (B, A) は、人が駅 B で乗車して駅 A で降車できるトリップを持たないものとみなされる。

30

## 4. 乗換パターンの計算

## 【 0 1 2 0 】

図 2 に戻ると、事前計算処理の第 4 のステージにおいて、乗換パターン計算 (TPC) モジュール 1 0 9 は最適乗換パターンを計算する（ステップ 2 0 5）。前述の通り、ジャーニーの乗換パターンは、そのジャーニーの行き先に到達するために行う必要がある 1 以上の駅でのトランジット車両乗換のシーケンスを表す。最適乗換パターンは、最小のコストで起点駅から目的駅に到達するためのルートを表す。なお、最適乗換パターンは、何らかの特定の時間において且つ何らかの距離コストの点で最適な、ジャーニーの乗換パターンである。従って、異なる乗換パターンに従う、異なるジャーニーは、その日の別の時刻で、且つ、最適化がトランジットの時間長、徒歩の量、乗換の数、又はその他の距離に基づくかどうかに応じて、最適でありうる。一実施形態において、TPC モジュール 1 0 9 は、3 つの異なる実施形態に従い乗換パターンを計算する。それは、グローバル駅無しの正確な乗換パターンの計算、グローバル駅有りの正確な乗換パターンの計算、及び、グローバル駅有りのヒューリスティック乗換パターンの計算である。各実施形態はここで説

40

50

明される。

#### A．グローバル駅無しの正確な乗換パターンの計算

##### 【 0 1 2 1 】

一実施形態において、T P Cモジュール109は、グローバル駅を利用せずに乗換パターンを計算する。図8を参照すると、この実施例に従い、最適乗換パターンを計算する（ステップ811）ためにT P Cモジュール109によって実行される方法のフローチャートが示されている。一実施形態において、図8に示す方法は、各駅毎に、他の全ての駅に対して、各駅のペアの最適乗換パターンを決定するために、実行される。説明の便宜上、方法は、出発（つまり起点）駅及び目的駅に関して説明される。

##### 【 0 1 2 2 】

一実施形態において、T P Cモジュール109は、入力としてトランジットグラフを取る（ステップ505）。T P Cモジュール109は、トランジットグラフ内の起点駅の駅ノード全てを初期化する（ステップ801）。すなわち、T P Cモジュール109は、起点駅の駅ノード全てにコスト0というラベルを割り当てる。従って、各駅ノードは、そのラベルにリスト表示された時間長0、及び、ペナルティ0を持つ。前述の通り、各駅の駅ノードは、駅からの異なる出発時刻を表す。

##### 【 0 1 2 3 】

駅ノードが初期化されたら、T P Cモジュール109は、出発地点から目的駅の各ノードまでの各駅ノードからパレートダイクストラ検索を実行する（ステップ803）。一実施形態において、パレートダイクストラ検索は、T P Cモジュール109が、ダイクストラアルゴリズムに典型的な単一のコストの代わりに、例えば「ペナルティ、時間長」などの多次元ラベルをノードにタグ付けすることを除き、ダイクストラ検索と同様なグラフの探索である。パレートダイクストラ検索は、全てのノード用にパレート最適ラベルのセットを保持する。ダイクストラでは1つの最適コストでだけあったが、パレートダイクストラ検索において多次元ラベルを持つことは、多数の潜在的な最適ラベルをもたらす。例えば、パレートダイクストラ検索により得たラベルセットを検討すると、そのセットは{(ペナルティ=3、時間長=10)、(ペナルティ=5、時間長=7)及び(ペナルティ=8、時間長=2)}を含む。この例では、いずれのラベルも多次元コストの点で他よりも厳密により良いものではないため、全てのラベルが最適とみなされる。

##### 【 0 1 2 4 】

パレートダイクストラ検索の結果として、T P Cモジュール109は、各起点ノードから、目的駅の特定の目的ノードへの最適ルートを提供する。各ルートは、起点駅と目的駅の間で乗換が行われる種々の駅を含む。一実施形態において、ルート内の各ノードは、対応するラベルセットを持っており、そのラベルセットはペナルティと時間長に関するコストを特定する。T P Cモジュール109は、トランジットグラフによって記述されたルート内の弧に関連付けられたコストに基づいて、ラベルセット内のラベルを割り当てる。

##### 【 0 1 2 5 】

T P Cモジュール109は、駅毎に、支配的ラベルを決定する（ステップ805）。一実施形態において、パレートダイクストラ検索が完了した後、トランジットグラフ内の各ノードは、最適ラベルのセットに関連付けられる。T P Cモジュール109は、目的駅のラベルから、最適ラベルが決定された起点駅に後戻りすることにより、最適経路を決定する。

##### 【 0 1 2 6 】

一実施形態において、T P Cモジュール109は、各乗車ノードのラベルセットを緩める（減ずる）。一実施形態によれば、ラベルセットの軽減は、T P Cモジュール109が1つのラベルセットからパレート最適性を保っている別のラベルセットにラベルを伝播することを含む。すなわち、第1のラベルセットから第2のラベルセットにラベルを追加したとき、全てのコスト次元において他のラベルに支配されているラベルは、第2のラベルセットから取り除かれる（フィルタされる）。一実施形態において、T P Cモジュール109は、以下の条件を満たすならば、経路1のコストペア( $d_1$ ,  $p_1$ )が経路2のコス

10

20

30

40

50

トペア (  $d_2, p_2$  ) より支配的であることを決定する。

$d_1 \leq d_2$  且つ  $p_1 \leq p_2$

TPCモジュール109は、経路1の時間長が経路2の時間長よりも短い又は等しく、且つ、経路1のペナルティが経路2のペナルティよりも小さい又は等しい場合、経路1のコストペアが経路2のコストペアより支配的であることを決定する。言い換えれば、コストの全ての次元において経路1が経路2よりも良い場合、経路1は経路2より支配的である。例えば、経路1が3600秒の時間長と100のペナルティを持っている一方、経路2が5400秒の時間長と200のペナルティを持っている場合、経路1は時間長及びペナルティの両方に関して経路2より支配的である。従って、ラベルセットの軽減は、支配的なラベルを取り除くことにより、第2のラベルセットが数を増やさないようにする。むしろ、各ラベルセットが軽減を用いてラベルの数を減少される。軽減の後にラベルセット内に残るラベルは、支配的なラベルとみなされる。一実施形態において、2つのラベルが時間長とペナルティに関して等しい場合、より早い到着時刻のラベルが支配的となる。

10

#### 【0127】

例えば、駅Aの乗車ノードを検討する。TPCモジュール109は、ラベルセットを仮想的待機弧に沿って伝えることにより、駅Aの各乗車ノードのラベルセットを軽減する。例えば、TPCモジュール109は、時刻“T”における駅Aの或る乗車ノードの或るラベル ( ペナルティ = 100、時間長 = 600 ) を、同じ駅Aの全ての乗車ノードに伝える。すなわち、時刻“T+dt”における駅Aの別の乗車ノードは、ラベル ( ペナルティ = 100、時間長 = 600+dt ) を受信する。TPCモジュール109は、それから、上述した支配的要素に基づいて、各乗車ノードのラベルセット内にて、他のラベルに支配されているラベルを取り除く。一実施形態において、駅ノードのラベルセットは、上述と同様の方法で、乗車ノードに軽減される。

20

#### 【0128】

支配的ラベルが決定されたら、TPCモジュール109は、各乗車ノードに残っている各支配的ラベルから、検索が実行された起点ノードへ最適経路を後戻りすることにより、最適乗換パターンを計算する ( ステップ807 )。TPCモジュール109は、また、経路が決定されるのと同様に、各最適経路に対して乗換パターンを決定して良い。すなわち、TPCモジュール109は、各最適経路内に示されたノードを分析して、どのノードがトランジット車両乗換に関連付けられているかを決定する。各ノードが特定の駅に関連付けられているので、TPCモジュール109は、乗換が行われるノードに基づいて、経路から乗換パターンを抽出できる。

30

#### 【0129】

最適経路から乗換パターンを決定することの一例は、以下の通りである。最適経路を返された駅Aから駅Fまでのパレートダイクストラ検索を検討する。返された最適経路の1つは、一連の駅A B D E Fに関連付けられた一連のノードである。TPCモジュール109は、経路内の各ノードに関連付けられた情報を分析して、駅B及びEで行われる乗換を決定する。従って、この例では、TPCモジュール109は、この経路から乗換パターンA B E Fを抽出する。

40

#### 【0130】

乗換パターンが決定されたら、TPCモジュール109は、乗換パターンを後の使用のために記憶する ( ステップ809 )。一実施形態において、支配的ラベルは、各駅の乗換パターンの計算が完了したとき、捨てられる。

### B. グローバル駅ありの正確な乗換パターンの計算

#### 【0131】

一実施形態において、TPCモジュール109は、グローバル駅選択モジュール107により決定されたグローバル駅を使って乗換パターンを計算する。トランジットグラフから最適乗換パターンを決定するために、TPCモジュール109は、グローバル駅選択モジュール107により決定されたグローバル駅のために、1つのグローバル駅検索を行う

50

。一実施形態において、グローバル駅検索は、経路を決定すべきトランジットグラフ内のグローバル駅から、トランジットグラフ内の該グローバル駅から到達可能な他の全てのノードまでのパレートダイクストラ検索である。すなわち、ＴＰＣモジュール１０９は、トランジットグラフ内のそれぞれ別の駅に到達するための最適経路を決定するために、各グローバル駅から全グラフ探索を行う。一実施形態において、例えばトリップの時間長及びペナルティなど異なる抽出条件に基づく、グローバル駅からトランジットグラフ内の駅に到達するための最適経路が１以上ある。

#### 【０１３２】

次に図９を参照すると、このアプローチに従い、乗換パターンを決定するためにＴＰＣモジュール１０９が実行する方法が示されている。ＴＰＣモジュール１０９は、ルート情報データベース１２５から全てのグローバル駅を決定する（ステップ９００）。一般に、グローバル駅毎に、ＴＰＣモジュール１０９は、グローバル駅から、１つのグローバル駅検索を実行する（ステップ９０１）。つまり、ＴＰＣモジュール１０９は、トランジットグラフ内の、グローバル駅に到達しうる他の全てのノードへの最適経路を決定するために、各グローバル駅的全ノード（駅ノードと乗車ノード）から、トランジットグラフ内で１つのダイクストラ検索を実行する。或るグローバル検索において、グローバル駅的全ノードは、時間長及びペナルティの二次元において、コスト０に初期化される。

#### 【０１３３】

トランジットグラフ内のグローバル駅からのグローバル検索の結果として返された最適経路毎に、ＴＰＣモジュール１０９は、返された経路からグローバル乗換パターンを決定する（ステップ９０３）。一実施形態において、グローバル乗換パターンは、グローバル検索の結果として得られる乗換パターンである。ＴＰＣモジュール１０９は、返された各経路において示されたノードを分析すること、及び、どのノードがトランジット車両乗換に関連付けられているかを決定することにより、グローバル乗換パターンを決定する。各ノードが特定の駅に関連付けられているので、ＴＰＣモジュール１０９は、グローバル駅無しの乗換パターン計算に関して上述した通り乗換が行われるノード経路から乗換パターンを抽出できる。

#### 【０１３４】

グローバル乗換パターンが決定されたら、ＴＰＣモジュール１０９は、ローカル乗換パターンを決定するために、ローカル検索を実行する（ステップ９０５）。一実施形態において、ローカル検索は、抽出条件に基づく非グローバル駅（つまりローカル駅）のノードからの検索である。一実施形態において、抽出条件は、トランジットグラフ内のローカル駅の駅ノードから別の駅への最短経路を決定するために、トランジットグラフを探索するためのものであり、その最短経路は経路内のグローバル駅での乗換を示さない。すなわち、ＴＰＣモジュール１０９は、トランジットグラフ内のそれぞれ別の駅への最適経路を決定するために、各ローカル駅の駅ノードからの検索を実行する。

#### 【０１３５】

ＴＰＣモジュール１０９は、そして、最適経路からローカル乗換パターンを決定する（ステップ９０７）。一般に、ローカル乗換パターンは、ローカル検索の結果として得られる乗換パターンである。ＴＰＣモジュール１０９は、各返された経路に示されたノードを分析して、どのノードがトランジット車両乗換に関連付けられているかを決定する。前述の通り、各ノードが特定の駅に関連付けられているので、ＴＰＣモジュール１０９は、乗換が行われるノードに基づいて、経路から乗換パターンを抽出できる。

#### 【０１３６】

ＴＰＣモジュール１０９は、更に、乗換パターンが受動的部分を含むかどうかを決定するために、各抽出された乗換パターンを分析する。一実施形態において、乗換パターンの受動的部分とは、乗換パターン内のグローバル駅での乗換が後続する一連の駅である。すなわち、乗換パターンが、グローバル駅で行われた乗換を示している場合、乗換パターン内の残りの駅のシーケンスは、乗換パターンの受動的部分とみなされる。起点駅から始まり、乗換が行われるグローバル駅で終わる乗換パターンの部分は、乗換パターンの能動

10

20

30

40

50

的部分とみなされる。一実施形態に従うと、乗換パターンの能動的部分は、グローバル駅を含む。一実施形態において、乗換パターンに含まれるグローバル駅は、ローカル検索がそこから実行されるローカル駅のアクセス駅とみなされる。アクセス駅は、本質的に、ローカル駅に関連付けられたグローバル駅であり、ローカル駅からのトランジット車両が、アクセス駅無しでは到達できない目的駅に到達できるようにするものである。従って、各ローカル駅は、ローカル乗換パターンの決定の結果として得られる1以上のアクセス駅を具備する。なお、ローカル乗換パターンは、受動的部分を持たなくてもよい。ローカル乗換パターン内のグローバル駅で乗換が行われない場合、又は、ローカル乗換パターン内の最後の駅がグローバル駅の場合、ローカル乗換パターンの全体が能動的とみなされる。

【0137】

ローカル乗換パターンが決定されたら、TPCモジュール109は、汎用乗換パターンとローカル乗換パターンを、ルート情報データベース125に記憶する(ステップ909)。一実施形態において、各ローカル乗換パターンの能動的部分のみが記憶される。

#### C. グローバル駅有りのヒューリスティック乗換パターン計算

【0138】

一実施形態において、TPCモジュール109は、種々のヒューリスティックアプローチに基づいて、グローバル駅選択モジュール107により決定されたグローバル駅を用いて、トランジットグラフの乗換パターンを計算する。一実施形態において、グローバル駅有りの乗換パターンを計算するために、TPCモジュール109により、下記のアプローチが使用される。

【0139】

i. ペナルティにより軽減された時間長

【0140】

ii. 多数の乗換後のローカル検索を停止する

【0141】

iii. グローバル駅の乗車でローカル検索を停止する

【0142】

iv. 全ての最適トリップを説明するために複数の乗換パターンの部分集合を使用する

【0143】

v. 最初及び最後の徒歩を取り除く

【0144】

各アプローチは以下に詳しく説明される。

#### i. ペナルティにより軽減された時間長

【0145】

次に図10Aを参照すると、経路のコストに基づいて乗換パターンを決定するためにTPCモジュール109により行われる方法が示されている。以下の説明は、起点駅と目的駅に関してなされる。一実施形態において、起点駅又は目的駅のいずれかがグローバル駅である。一実施形態において、下記の処理は、トランジットグラフ内の全ての可能な起点ノード毎に、TPCモジュール109により繰り返される。それにより、TPCモジュール109は、トランジットグラフから起点駅毎の支配的経路を決定し、結果として起点駅毎に乗換パターンのセットを得る。

【0146】

まず、TPCモジュール109は、トランジットグラフ上で起点駅からパレートダイクストラ検索を実行する(ステップ1001)。起点駅のノード毎に、パレートダイクストラ検索が実行され、その結果として可能な各目的駅のノードへの経路を得る。従って、ダイクストラ検索は、起点駅から各目的駅の複数の最適経路をもたらす。

【0147】

ダイクストラ検索の一部として、TPCモジュール109は、コストに基づくダイクストラ検索の結果として得た経路の中から、支配的経路を決定する(ステップ1003)。TPCモジュール109は、どの経路が支配的かを決定するために、経路のコストペアの

10

20

30

40

50



累積を分析する。一実施形態において、支配的経路は、検索結果として得た他の全ての経路のコストに比較して、時間長とペナルティに関してより少ないコストを持つ。

【0148】

一実施形態によれば、TPCモジュール109は、以下の条件を満たすならば、経路1のコストペア( $d_1, p_1$ )が経路2のコストペア( $d_2, p_2$ )より支配的であることを決定する。

第1条件:  $d_1 \leq d_2$  且つ  $p_1 \leq p_2$ 、又は、

第2条件:  $d_1 < d_2$  且つ  $p_1 > p_2$ 、並びに、 $[d_1 + A \times (p_1 - p_2)] \leq d_2$ 、  
ここで  $A = 8$

【0149】

第1条件において、上述の通り、TPCモジュール109は、経路1の時間長が経路2の時間長よりも短い場合、経路1のコストペアが経路2のコストペアより支配的であることを決定する。加えて、経路1のコストペアが経路2のコストペアより支配的であるには、経路1のペナルティが、経路2のペナルティよりも小さいか又は等しくなければならない。

【0150】

TPCモジュール109が経路1の時間長は経路2の時間長よりも短いけれども、経路1のペナルティは経路2のペナルティよりも大きいと判定した場合、TPCモジュール109は、ルート1の時間長と、重さ要素Aが乗算されるルート1とルート2のペナルティの差分と、の合計に基づく計算を実行する。上記の一実施形態では  $A = 8$  と示されている。しかし、別の実施形態では、別の値が使用され得る。計算の結果が、ルート2の時間長よりも小さいか又は等しい場合、TPCモジュール109は、経路1が経路2より支配的であると決定する。一例として、経路1が3600秒の時間長と500ペナルティを持っている一方、経路2が4000秒の時間長と100のペナルティを持っているとする。この例では、経路1の時間長が経路2の時間長より小さいので第2条件が適用されるが、経路1のペナルティは経路2のペナルティよりも大きい。支配的経路を決定するために、第2条件の式が使用される。上記の式を使用すると、 $[3600 + 8 \times (500 - 100)]$  は4400であり、経路2の時間長よりも小さいくも等しくもないため、経路1は経路2に対して支配的ではない。従って、この例の経路1は経路2に対して支配的でない。支配的経路が決定された、TPCモジュール109は支配的経路の乗換パターンを記憶する(ステップ1005)。

【0151】

上記の処理がトランジットグラフ内の全ての可能な起点駅毎に繰り返されるので、TPCモジュール109は、記憶された最適乗換パターンのセットを結果として得る。

ii. 多数の乗換後のローカル検索を停止する

【0152】

次に、図10Bを参照すると、人口密集地域ではトリップのルートは一般に最小の乗換数でグローバル駅に到着し、田園地方では多数駅の中で乗換が行われる駅は少ししかないとするヒューリスティックに基づいて、グローバル駅への乗換パターンを決定するためにTPCモジュール109により行われる方法が示されている。一実施形態において、このヒューリスティックは、2つのトランジット車両乗換以内でグローバル駅に到達しなければならない(つまりトリップの間に3つのトランジット車両が使用される)という条件下で、動作する。別の実施形態では、異なる乗換数が使用され得る。

【0153】

グローバル駅への乗換パターンを決定するために、TPCモジュール109は、トランジットグラフ内の所要の出発ノードからローカル検索を行う(ステップ1007)。上述の通り、ローカル検索は、トランジットグラフ内の他の駅への最適経路を決定するための、ローカル駅の駅ノードからの検索である。TPCモジュール109は、出発ノードから開始してトランジットグラフ内の他のノードに広がるトランジットグラフ内の最短経路を捜し出す。TPCモジュール109は、その経路から、グローバル駅が、出発ノードから

10

20

30

40

50

出発して2つのトランジット車両乗換以内で到達されているかどうかを、決定する（ステップ1009）。具体的には、出発ノードから開始して、トランジットグラフ内のルートに沿って、各後続ノード毎に、TPCモジュール109は、乗換が行われるか、又は、ノードに関連付けられた駅がグローバル駅かを決定する。グローバル駅に到達するまえに多くとも2つの乗換が行われるならば、グローバル駅に至る駅の経路は乗換パターンとみされる。

#### 【0154】

例えば、トランジットグラフ内のローカル駅Aの出発ノードが経路に含まれており、経路に沿う第1グローバル駅が駅Gとする。駅Aのノードは、A B C D E F Gの経路を表す一連のノードに連結される。駅Aから駅Gの間で乗換が行われなとする。駅Aから開始して、TPCモジュール109は、経路を捜し出し、各後続駅がグローバル駅であるか、又は、その後続駅で乗換が行われるかどうかを決定する。この例では、駅Aから駅Gの間で乗換が行われなないので、TPCモジュール109は、駅Gまで到達したら経路をトラバースする（スキャンする）のを停止する。従って、経路A B C D F F Gのための乗換パターンは、駅A及びCの間で乗換が行われなことを示すAGである。

#### 【0155】

非グローバル駅で乗換が行われる場合、TPCモジュール109は、乗換の実行を捜し出し、且つ、経路内のグローバル駅又は非グローバル駅での第3の乗換に到達するまで処理を繰り返す。TPCモジュール109が2つの乗換以内でグローバル駅に到達したと決定した場合は、その経路に関連付けられた乗換パターンが決定される。TPCモジュール109が最大2つの乗換の後または3つ目の乗換でグローバル駅に到達していないと決定した場合は、分析中の経路は無視されて、PCモジュール109は、別の非グローバル駅を対象として上記の処理を繰り返す。TPCモジュール109は、グローバル駅への乗換パターンのセットを決定するために、トランジットグラフ内の全ての非グローバル駅の各ノードに対して分析を行う。

#### 【0156】

一例として、上記で用いたA B C D E F Gの経路において、駅Gがグローバル駅でありトランジット車両乗換が駅Cと駅Fで行われるものとする。駅Aから開始して、TPCモジュール109は、各後続駅がグローバル駅であるか、又は、その後続駅で乗換が行われるかどうかを決定する。駅Cで乗換が行われるので、TPCモジュール109は乗換を追跡する。グローバル駅Gに到達する前に駅Fで2番目の乗換が行われるのみであるため、TPCモジュール109は、経路A C F Gを、駅Aからグローバル駅Gまでの乗換パターンとみなす。

#### 【0157】

次に、駅B、C及びFで乗換が行われるとする。この例では、グローバル駅Gに到達する前に3つのトランジット車両の乗換が行われるので、TPCモジュール109は、第3番目の乗換に到達したら、汎用駅への経路を広げるのを停止する。一般に、3番目の乗換で経路の拡張を停止することにより、TPCモジュール109は、多くの経路を検討からはずし、且つ、最適経路の事前計算を実質的により速くできる。

#### 【0158】

前述の処理が、トランジットグラフ内のローカル駅とグローバル駅の賢明な組み合わせのペアの全てに対して繰り返し行われるので、TPCモジュール109は、乗換パターンのセットを結果として得て、そして、それを記憶する。

iii. グローバル駅の乗車でローカル検索を停止する

#### 【0159】

次に図10Cを参照すると、ローカル駅からグローバル駅への乗換パターンを決定するためにTPCモジュール109によって実行される方法が示されている。なお、説明の便宜上、以下の説明は、トランジットグラフ内の1つのノードからの乗換パターンの決定に関して記載している。当業者にとっては、以下の説明が、グローバル駅への乗換パターンを決定するために、トランジットグラフ内のローカル駅の各駅ノード毎に繰り返され得る

ことが明白である。

【0160】

一実施形態において、TPCモジュール109は、ローカル駅の所与の駅ノードからローカル検索を実行する(ステップ1011)。TPCモジュール109は、トランジットグラフ内の駅ノードから始まる経路を捜し出して、トランジットグラフ内の他のノードに広げる。TPCモジュール109は、その経路から、いつグローバル駅の乗車ノードが到達されるかを決定する(ステップ1013)。より具体的には、駅ノードから始まり、トランジットグラフ内の経路に沿う後続のノード毎に、TPCモジュール109は、トランジットグラフ内の後続のノードがグローバル駅に関連付けられた乗車ノードであるか決定する。グローバル駅の後続のノードに到達したら、TPCモジュール109は、グローバル駅の乗車ノードに至る経路の乗換パターン(駅ノードを含む)を決定する(ステップ1015)。一実施形態において、後述する通り、グローバル駅に停車しないジャーニーを説明するための縫い合わせ(スティッチング)が要求される。

10

【0161】

例えば、前述のように、駅C及びDで乗り換えるローカル駅Aからグローバル駅Gの経路A B C D E F Gを検討する。経路内で駅Gに関連付けられたノードが乗車ノードと仮定する。TPCモジュール109は、グローバル駅に関連付けられた乗車ノードに到達するまで経路をたどる。従って、この場合、駅Gが経路内で到達されるグローバル駅の最初の乗車ノードであるため、経路A C D Gが乗換パターンとみなされる。駅Aと駅Gの間で乗換が行われないならば、その経路の乗換パターンは、乗換が行われないことを示すA Gになる。前述の通り、乗換パターンは、トリップの出発及び目的駅を含む。

20

【0162】

別の例として、駅Eもまたグローバル駅であり、経路内の駅Dに関連付けられたノードが乗車ノードと仮定する。この場合、TPCモジュール109は、グローバル駅の最初の乗車ノードに到達するまで、経路をたどる。この場合、駅Gの乗車ノードではなく、駅Dの乗車ノードがそれである。従って、A C D Eの乗換パターンが生成される。

【0163】

上述の処理がトランジットグラフ内のローカル駅とグローバル駅の賢明な組み合わせのペア毎に繰り返されるので、TPCモジュール109は、その結果として乗換パターンのセットを得て、それからそれを記憶する。

30

iv. 全ての最適トリップを説明するために複数の乗換パターンの部分集合を使用する

【0164】

次に、図10Dを参照すると、トランジットグラフ内の全ての駅から乗換パターンを決定するために、TPCモジュール109により実行される方法がしめされている。図10Dに示されたアプローチが、起点駅からの目的駅へのトリップ数を減少するために使用される。一般に、下記のアプローチは、全ての乗換パターンのセットを目的駅への最適トリップに、目的駅への全ての最適トリップをカバーするのに十分な部分集合によって置き換えるために使用される。一実施形態において、トランジットグラフ内の全ての駅ペア毎に、駅間の乗換パターンを決定するために、以下の記述が繰り返される。

【0165】

40

一実施形態において、TPCモジュール109は、トランジットグラフ内の起点駅の各ノード(すなわち起点駅)から目的駅のノードへのローカル検索を実行する(ステップ1017)。その検索から、TPCモジュール109は、起点駅からの種々の出発時間における、起点駅から目的駅への最適経路のセットを、結果として得る。TPCモジュール109は、検索から返された最適経路毎に、起点駅から目的駅への乗換パターンを決定して(ステップ1019)、最適乗換パターンのセットを形成する。最適乗換パターンのセットは、特定の日における起点駅から目的駅への全ての可能なトリップをカバーするだろう。

【0166】

TPCモジュール109は、セット内の乗換パターンに含まれた冗長なトリップを決定

50

する（ステップ1021）。すなわち、起点駅から目的駅へのトリップを表す乗換パターン毎に、TPCモジュール109は、同じ起点駅から目的駅への、他の乗換パターンを決定する。TPCモジュール109は、両方の乗換パターンを保持しておく必要性を除去するために、乗換パターンの1つが起点駅から目的駅へのトリップをカバーできるかどうかを決定する。一実施形態において、運行日毎に時刻T'の第2のトリップが存在するとき、1以上の乗換パターンは時刻Tのトリップをカバーしており、その第2のトリップは同じ乗換パターンを持つ決定されたセットからの乗換パターンを持ち、出発地点から第1のトリップよりも早く出発せず、且つ、第1のトリップのコストをわずかに閾値よりも超えるコストを持つ。

【0167】

例えば、所与のトリップのためのシナリオを検討する。一日を通じてそのトリップのための乗換パターンABCを持つ多数の最適トリップがあるとする。それら最適トリップの中で、トリップは、駅Aを22時40分に出発して駅Cに23時25分に到着し、200のペナルティを持つものである。更に、乗換パターンADCの一度限りのトリップがある。その一度限りのトリップは、駅Aを22時38分に出発して駅Cに23時22分に到着し、180のペナルティを持つ。たとえ一度限りのトリップが、22時38分に出発したいユーザにとってより良い結果であっても、一度限りのトリップは、乗換パターンABCに関連付けられたトリップへの支持から、却下される。というのも、乗換パターンABCに関連付けられたトリップは、後の時刻に同じ起点駅から出発するのであるから、却下されたトリップがいつ実行されるときはいつでも、実行されるトリップを提供するからである。更に、乗換パターンABCのトリップの最低のコスト（すなわち3分遅れの到着時刻で、20の余分なペナルティ）は閾値より低い。

v. 最初及び最後の徒歩を取り除く

【0168】

一実施形態において、TPCモジュール109は、最適と思われるが、出発及び行き先地点に近在の複数の駅がトランジットサーバ100によって検討されるクエリ時間の間は最適ではない駅間の経路を取り除くために、初期の及び徒歩ヒューリスティックを使用する。一例として、400メートル離れており、同じバス路線により提供されていないバス駅A及びBを検討する。クエリ時間における、このヒューリスティックのために、バス駅AからBへの徒歩が最適と決定される。しかし、ヒューリスティックなしには、駅Aを出発するローカル検索からの最適の結果は、駅Aを出発して、駅Cに遠回りしてから駅Bに到達するトリップである。従って、このヒューリスティックは、TPCモジュール109がAからBへのトランジットトリップを取り除くことを可能とする。駅Aから駅Bへは単に歩くのが最適だからである。

【0169】

次に、図10Eを参照すると、ローカル起点駅から近隣のローカル駅への徒歩コストを決定するためにTPCモジュール109により実行される方法である。一実施形態において、TPCモジュール109は、所与のローカル駅からのローカル検索を実行する（ステップ1023）。ローカル検索から、TPCモジュール109は、ローカル駅に近在の全ての駅を決定する（ステップ1025）。一実施形態において、TPCモジュール109は、例えば1マイルなどラジアル距離で近在駅を決定する。近在駅毎に、TPCモジュール109は、検索が実行されたトランジット駅から近在駅への徒歩コストに等しいコストラベルを加える。一実施形態において、徒歩のコストは、トランジット駅から近在駅へ歩くのにかかる推測の時間量を表す。

【0170】

例えば、ローカル駅Aを仮定する。TPCモジュール109は、トランジットグラフ内で半径2マイル内の全ての近在駅を決定するためにローカル駅Aからローカル検索を実行する。検索は、駅Aの半径2マイル内にある駅M、K及びGのリストを結果としてもたらす。TPCモジュール109は、それから、近在駅のそれぞれに、駅Aからその近在駅への徒歩のコストを示すラベルを追加する。例えば、駅Aから駅Mに或る人が歩くのに平均

10

20

30

40

50

で1800秒(0.5時間)かかるとする。TPCモジュール109は、駅Mに1800秒のコストを示すラベルを追加する。

【0171】

一実施形態において、TPCモジュール109は、前述と同様な処理をグローバル駅毎に実行する。所与のグローバル駅に対して、TPCモジュール109は、トランジットグラフ内でグローバル駅から到達可能な全ての近在駅を決定するために、グローバル検索を実行する。近在駅毎に、TPCモジュール109は、近在駅の駅ノードにコストを追加する。一実施形態において、TPCモジュール109は、ノードに関連付けられた乗換ペナルティと徒歩コストに基づいてグローバル駅にペナルティを追加する。

【0172】

次に、図10Fを参照すると、近在駅から目的駅までの徒歩コストを決定するために、TPCモジュール109が実行する方法が示されている。例えば、起点駅から目的駅への所与のトリップは、起点駅から目的駅に直接行くよりもむしろ、起点駅から目的駅の近在駅を通過して、その近在駅から目的駅まで歩いたほうが最適であるかもしれない。

【0173】

一実施形態において、トリップの目的駅は、到着ループを備える。到着ループは、目的駅の乗車ノードのループと、弧によって連結されたそれらラベルである。TPCモジュール109は、到着ループで弧を緩和して(ステップ1029)、それぞれ対応する乗車ノードで最適な、とはいえ実際のところは最適ではない、トリップのラベルを取り除く。というのも、より早く到着するというコストの点でより良いトリップが存在するからである。

【0174】

TPCモジュール109は、近くにある駅を決定するために目的駅でローカル検索を実行する(ステップ1031)。TPCモジュール109は、近在駅の乗車ノードから目的駅の駅ノードへの弧を追加することにより、目的駅の到着ループを拡張する(ステップ1033)。一実施形態において、それら弧は、近在駅から目的駅への徒歩を示す徒歩弧を表す。

【0175】

前述の処理を、ローカル駅毎に繰り返すことにより、TPCモジュール109は、気か

5. 乗換パターンの圧縮

【0176】

一実施形態において、事前計算処理の最終局面は随意である。乗換パターン圧縮モジュール111は、前述した乗換パターンを完全な形で保存するよりも、効率的に保存することを目的として、TPCモジュール109により決定された乗換パターンを圧縮する。一般に、乗換パターン圧縮モジュール111は、各乗換パターンを小さい部分に分割する。乗換パターンを圧縮することにより、乗換パターンは、一般により小さい断片に分解され、それらのサイズのために、高い冗長性を持ち、それら断片をより効果的に記憶できる効果的な方法で圧縮されるだろう。

【0177】

図11を参照すると、乗換パターンを圧縮するために乗換パターン圧縮モジュール111が実行する処理の流れが示されている。後述の処理は、前述したTPCモジュール109により決定された全乗換パターン毎に繰り返される。

【0178】

一実施形態において、乗換パターン圧縮モジュール111は、入力として乗換パターンを取る(ステップ1100)。乗換パターン圧縮モジュール111は、トランジットグラフ内の、単一の駅の実体を用いた繰り返しの駅の連なりをそれぞれ置き換えるために、乗換パターンを随意でフィルタする(ステップ1101)。例えば、次のようなジャーニーの乗換パターンを考える：駅Aで乗車し、駅Bに到着し、駅Bで乗車し、駅Cに到着し、駅Dで乗

10

20

30

40

50

車して駅Fに到着する。言い換えれば、ジャーニーの乗換パターンは、A B B C D Fである。乗換パターンの全ての乗車イベントはトランジットグラフの駅ノードを表し、全ての到着イベントはトランジットグラフの乗車ノードを表す。この例では、乗換パターン圧縮モジュール111は、駅Bが繰り返されており、単一の実体“B”を用いた駅の連なり“B B”に取って代わることを決定する。乗換パターン圧縮モジュール111は、本質的に、駅又は乗車ノードの間の区別をしない。従って、その結果として得るパターンは、A B B C D Fではなくて、A B C D Fになる。別の実施形態では、乗換パターンは、乗換パターン内の繰り返しの駅の連なりを取り除くためのフィルタをされない。

#### 【0179】

乗換パターンがフィルタされるか否かにかかわらず、乗換パターン圧縮モジュール111は、乗換パターンサフィックスを決定する(ステップ1103)。乗換パターンサフィックスは、サフィックスに関連付けられた駅に先行する乗換パターン内の2つの駅の連なりを示す。本質的に、2つの駅に先行されない何れの駅も除くと、乗換パターン内の各駅は、目的駅とみなされ得るものであり、従って、それらは関連付けられたサフィックスを持ち得る。従って、乗換パターン内の、2つの駅に先行された駅毎に、乗換パターン圧縮モジュール111は、サフィックスに関連付けられた駅に先行する乗換パターン内の2つの駅を示す乗換パターンサフィックスを決定する(ステップ1105)。起点駅から目的駅への乗換パターンサフィックスは、そのパターンのための、乗換パターンサフィックスセットとみなされる。乗換パターンサフィックスが決定されたら、乗換パターン圧縮モジュール111は、各乗換パターンサフィックスとそれに関連する駅(乗換指示セット)を記憶する(ステップ1107)ことにより、圧縮された乗換パターンを得る(ステップ1109)。このようにして乗換パターンを記憶することにより、乗換パターン圧縮モジュール111は乗換パターンのセグメント(区切られた部分)を記憶する。

#### 【0180】

圧縮された乗換パターンA B C D Fの記憶構成の一例は、下記の通りである。

【表8】

乗換パターンサフィックス	駅
{C, D}	F
{B, C}	D
{A, B}	C

#### 【0181】

上記の通り、関連付けられた乗換パターンサフィックスを持つ乗換パターンA B C D Fの駅は、F、D及びCである。乗換パターンサフィックスは、関連付けられた駅に到着するために使用される乗換パターン内の2つの駅の連なりを示す。上記の乗換パターンサフィックスの集合は、起点駅A及び目的駅Fのための、乗換パターンセットとみなされる。また、この記憶構成は、実際の乗換パターンを失くしており、駅AからFへの乗換パターンは、上記記憶構成から再構築できる。

#### 【0182】

更なる例として、所与の乗換パターンQ Y G L M K Rを挙げる。圧縮された乗換パターンQ Y G L M K Rの記憶構成の一例は、下記の通りである。

【表 9】

乗換パターンサフィックス	駅
{M, K}	R
{L, M}	K
{G, L}	M
{Y, G}	L
{Q, Y}	G

## 【0183】

10

この例では、関連付けられた乗換パターンサフィックスを持つ乗換パターン Q Y G L M K R の駅は、R、K、M、L 及び G である。各乗換パターンサフィックスは、サフィックスに関連付けられた駅に到着するために使用される、フィルタされた乗換パターン内の 2 つの駅の連なりを示す。例えば、駅 R に関して、駅 M 及び K の連なりは、フィルタされた乗換パターン内で駅 R に先行する駅の順を示す。テーブル内のその他の駅毎に、対応する乗換パターンサフィックスは、乗換パターンサフィックスに関連付けられた駅に先行する 2 つの駅の連なりを示す。乗換パターンサフィックスの集合は、起点駅 Q から目的駅 R への乗換パターンサフィックスのセットである。

## 【0184】

20

クエリ時間において、乗換パターンは乗換パターンサフィックスのセットから再帰的に再構築される。この処理は、更に詳細に後述される。また、乗換パターン自身は失われ、記憶領域の量は著しく減少する。

## III. クエリ時間処理

## 【0185】

図 1 に戻ると、前述の通り、クエリ解決モジュール 115 は、ユーザのクエリ（問い合わせ）に応じて最適ジャーニールートを決定するために、事前計算モジュール 103 により提供される乗換パターン計算を使用する。事前計算の間に決定された乗換パターンを用いることにより、クエリ解決モジュール 115 は、要求を遂行するために必要となる計算量を極小にできる。図 1 に示すとおり、クエリ解決モジュール 115 は、ローカル駅決定モジュール 117 と、クエリグラフ生成モジュール 119 と、クエリグラフ分析モジュール 121 と、トリップ供給モジュール 123 と、乗換パターン構築モジュール 135 を備える。次に、図 12 を参照すると、クエリ解決モジュール 115 によりクエリを処理するための行程の一実施形態は、下記の機能的ステージを備える。

30

## 【0186】

1200：近在駅を決定する

## 【0187】

1201：乗換パターンを構築する

## 【0188】

1203：クエリグラフを生成する

## 【0189】

1205：クエリグラフを拡張する

40

## 【0190】

1207：クエリグラフを評価する

## 【0191】

1209：最適ジャーニーを決定する

## 【0192】

1211：最適ジャーニーを供給する

## 【0193】

クエリを処理する各機能的ステージは以下に詳しく説明される。

1. 近在駅を決定する

50

## 【 0 1 9 4 】

図 1 3 は、一実施形態に従って、ローカル駅決定 ( L S D ) モジュール 1 1 7 により実行される、起点 (つまり出発) 地点と目的地点に関する近在駅を決定すること (ステップ 1 2 0 0 ) によりクエリを処理する第 1 ステージの処理フローを示す。一実施形態において、ローカル駅決定 ( L S D ) モジュール 1 1 7 は、入力としてクエリ 1 3 0 0 を受信する。クエリ 1 3 0 0 は、公共トランジット指示のために、ユーザから受信する。一実施形態において、クエリ 1 3 0 0 は、或るジャーニーのための起点地点及び目的地点を少なくとも含む。起点地点は、ユーザの現在の地理的位置又はユーザがトリップを開始したい任意の位置であり、一方、目的地点は、行き先の地理的位置である。起点及び目的地点は、トランジット駅の位置であってもよいし、そうでなくてもよい。一実施形態において、起点及び目的地点は住所の形式である。これら受信したクエリは、また、起点地点からの出発又は目的地点への到着の時刻及び/又は日付を含んでよい。例えば、クエリは、「午後 2 時発カリフォルニア州 9 4 0 4 1、マウンテンビュー、カリフォルニア通り 8 0 1 ( 8 0 1 California Street, Mountain View, CA94041 ) から、カリフォルニア州 9 4 1 0 4、サンフランシスコ、カリフォルニア通り 5 5 5 ( 555 California San Francisco, CA94104 ) へのルートが欲しい」と言明するだろう。

10

## 【 0 1 9 5 】

クエリ 1 3 0 0 を受信すると、 L S D モジュール 1 1 7 は、ステップ 1 3 0 1 において、起点地点と目的地点との両方を、各地点に対応する緯度及び経度座標に対応付ける。そして、 L S D モジュール 1 1 7 は、ステップ 1 3 0 3 において、起点地点と目的地点との緯度及び経度座標から、ラジアル距離内の全ての駅を決定する (つまり位置を示す)。一実施形態において、ラジアル距離は、起点及び目的地点の半径 2 マイル内の全ての駅が決定されるように、2 マイルに設定される。別の実施形態では、異なるラジアル距離を使用できる。

20

## 【 0 1 9 6 】

近在駅の位置を示した後、 L S D モジュール 1 1 7 は、起点地点に近在の全ての駅を含む駅 I D の起点駅リストを作成し (ステップ 1 3 0 5 )、且つ、目的地点に近在の全ての駅を含む駅 I D の目的駅リストを作成する (ステップ 1 3 0 7 )。また、起点駅から、それに関連付けられた近在駅のそれぞれへの徒歩コストもまた、起点駅リストに供給される。同様に、各近在駅から目的駅への徒歩コストもまた、目的駅リストに供給される。

30

## 2 . 乗換パターン構築

## 【 0 1 9 7 】

クエリ時間処理の第 2 ステージにおいて、乗換パターン構築モジュール 1 3 5 は、乗換パターンを作成する (ステップ 1 2 0 3 )。以下の乗換パターン構築の説明は、乗換パターンがグローバル駅を使って計算されたものであるとの仮定に基づいている。一実施形態において、乗換パターンを作成するために、乗換パターン構築モジュール 1 3 5 は、入力として、起点駅リスト 1 3 0 5 及び目的駅リスト 1 3 0 7 を受信する。乗換パターン構築モジュール 1 3 5 は、起点駅リスト 1 3 0 5 内の起点駅から、目的駅リスト 1 3 0 7 内の目的駅への乗換パターンを決定する。乗換パターンの決定は、起点駅がグローバル駅かローカル駅かに依存する。

40

## 【 0 1 9 8 】

一実施形態において、起点駅リスト 1 3 0 5 内のグローバル駅毎に、乗換パターン構築モジュール 1 3 5 は、記憶されたグローバル乗換パターンから、グローバル駅から目的駅リスト 1 3 0 7 内の各駅への乗換パターンを示すグローバル乗換パターンを決定する (つまり検索する)。一例として、グローバル駅とみなされた起点駅 A と、グローバル駅又はローカル駅のどちらでもありうる目的駅 L を検討する。乗換パターン構築モジュール 1 3 5 は、記憶されたグローバル乗換パターンから、駅 A から出発して駅 L で終わるグローバル乗換パターンを決定すると、駅 A から駅 L への 1 以上のグローバル乗換パターンを結果として得る。この処理は、起点駅リスト 1 3 0 5 内のグローバル駅毎に、目的駅リスト 1 3 0 7 内の各駅まで、繰り返される。

50



## 【 0 1 9 9 】

ー実施形態において、起点駅リスト 1 3 0 5 内のローカル駅毎に、乗換パターン構築モジュール 1 3 5 は、各ローカル駅に関連付けられたアクセス駅のセットを決定する。前述の通り、アクセス駅は、ローカル駅に関連付けられたグローバル駅である。各ローカル駅のアクセス駅のセットが決定されたら、乗換パターン構築モジュール 1 3 5 は、ローカル駅毎に、そのローカル駅からその各アクセス駅へのローカル乗換パターンを決定する。乗換パターン構築モジュール 1 3 5 は、そして、各アクセス駅から目的駅リスト 1 3 0 7 内の各目的駅へのグローバル乗換パターンを決定する。従って、ローカル駅から目的駅への乗換パターンは、ローカル乗換パターンとグローバル乗換パターンとの組み合わせである。

10

## 【 0 2 0 0 】

しかしながら、随意的乗換パターン圧縮が実行された場合には、乗換パターンそれ自身は記憶されないことに留意されたい。したがって、乗換パターン構築モジュール 1 3 5 は、ジャーニーのための完全な乗換パターンを作るために、乗換パターンを復元する。言い換えれば、乗換パターン構築モジュール 1 3 5 は、起点駅及び目的駅に関連付けられた乗換パターンサフィックスを用いてグローバル乗換パターンとローカル乗換パターンとを再帰的に作る。前述の通り、乗換パターンサフィックスセットは、駅のペアに関連付けられたサフィックスのセットを表す。セット内の各サフィックスは、乗換パターンサフィックスに関連付けられた駅に到達するのに先立つ 2 つの駅でのトランジット車両の乗換を表す 2 つの駅の連なりを表す。

20

## 【 0 2 0 1 】

例えば、所与の起点駅 A と目的駅 F のペアのための乗換パターンを決定するために、乗換パターン構築モジュール 1 3 5 は、A から F へのサフィックスのセットを決定する。駅 F に関連付けられたサフィックスの 1 つが ( D , F ) とする。このことは、A、・・・D、E、F に至る A から F への乗換パターンがいくつか存在することを示す。

## 【 0 2 0 2 】

乗換パターンの次の脚を作るために、乗換パターン構築モジュール 1 3 5 は、直前の乗換パターンサフィックスに記録された 2 つの駅に関連付けられた乗換パターンサフィックスセットを決定する。そして、乗換パターンサフィックスは連結され、乗換パターンの部分集合を結果として得る。乗換パターン構築モジュール 1 3 5 は、起点地点から目的駅までの乗換パターンが構築されるまで、乗換パターンサフィックスセットの検索を繰り返す。

30

## 【 0 2 0 3 】

前述した A、・・・D、E、F で終わる A から F への乗換パターンの例を続けると、乗換パターン構築モジュール 1 3 5 は、D に至るペア ( A , E ) のためのサフィックスのセットを検索する。この場合、乗換パターン構築モジュール 1 3 5 は、駅 D のためのサフィックス ( C , D ) 又は ( B , D ) を検索する。乗換パターン構築モジュール 1 3 5 は、そして、C に至るペア ( A , D ) のためのサフィックスと、B に至るペア ( A , D ) のためのサフィックスを検索する。これは、サフィックス ( A , C ) と ( B , C ) が検索される結果となる。乗換パターン構築モジュール 1 3 5 は、それから、B に至るペア ( B , D ) のためのサフィックスを決定し、結果としてサフィックス ( A , B ) と、サフィックス ( A , B ) をもたらす B に至るペア ( B , C ) とを得ることとなる。乗換パターン構築モジュール 1 3 5 は、起点駅 A から目的駅 D への乗換パターンを形成するために、乗換パターンサフィックスを連結する。上述したこの例では、A から F への乗換パターンセットは、( A , B , D , E , F )、( A , B , C , D , E , F ) 及び ( A , C , D , E , F ) である。

40

### 3 . クエリグラフ生成

## 【 0 2 0 4 】

クエリ処理の第 3 ステージにおいて、クエリグラフ生成 ( Q G G ) モジュール 1 1 9 は、クエリグラフを生成する ( ステップ 1 2 0 5 ) 。一般に、クエリグラフは、クエリに関

50

するトランジット情報のみを含み、時間独立である（時間に依存しない）。クエリグラフは、起点地点から目的地点への最適ルートを決断するために使用される。クエリグラフは、クエリグラフに含まれる情報が当該クエリにのみ関連しているため、最適ルートを決断するために必要とされる計算を極小にする。一実施形態において、クエリグラフは、クエリに関連する起点地点と目的地点、駅を表すノード、及び、駅を連結する弧を含む。クエリグラフ内の駅は、トランジット車両の乗換が行われる起点地点から目的地点に到達するための経路に含まれるトランジット駅を表す。クエリグラフ内の各弧は、ノードを連結する、駅ペア間の直通接続を表す。従って、弧は、その弧によって連結された目的駅に到達するための経路上で停車が行われる（しかし乗換はしない）1以上の中間駅を表す。更に、或る弧は、車両を出る駅から車両に乗車する駅までの徒歩をも表す。

10

#### 【0205】

図14を参照すると、受信したクエリに関するクエリグラフを生成するための、QGGモジュール119の処理フローが示されている。QGGモジュール119は、乗換パターン構築モジュール135により決定された起点地点と目的地点に関連付けられた乗換パターン1400を入力として取る。

#### 【0206】

駅ペア（すなわち起点駅と目的駅）毎の乗換パターンを受信したら、QGGモジュール119は、駅ペア毎の乗換パターンのセットを用いてクエリグラフを生成する（ステップ1403）。QGGモジュール119は、各駅ペアのセットを一連のノードとして表す（ステップ1401）ことにより、クエリグラフを生成する。すなわち、乗換パターン内に示された各駅は、クエリグラフ内のノードとして表され、そのノードは乗換パターンによって示された順に配列される。QGGモジュール119は、そして、乗換パターン何に示されたシーケンス（順列）に従って、各駅のペアを弧で連結する。弧はクエリグラフ内の移動の方向を表しており、クエリグラフ内のノードのペアを連結する。一実施形態において、各弧は、駅への歩行又はトランジット車両での駅への移動のいずれも表す。各弧の表現は、乗換パターン内の繰り返しシーケンスが乗換パターン圧縮のときにフィルタされているかどうかに応じて、クエリグラフに供給されてよい。この説明においては、乗換パターンは、繰り返しシーケンスのフィルタが成されているものである。

20

#### 【0207】

一実施形態において、QGGモジュール119は、起点地点と目的地点を表す2つの人為的なノードを除いて、トランジットグラフ内の各ノードのノードIDにトランジットグラフ内の駅IDを割り当てる。QGGモジュール119は、これら人為的なノードに、一実施形態によれば、例えば出発地点に“1”、行き先地点に“2”などという汎用的IDを与える。QGGモジュール119は、起点地点及び目的地点を表すノードを、クエリグラフ内の他のノードに、徒歩を表す弧によって連結する。QGGモジュール119が各ノードに識別IDを割り当てることを終えたら、クエリグラフが完成する（ステップ1405）。

30

#### 【0208】

図15を参照すると、“1”とラベル付けされたノードによって表された起点地点から“2”とラベル付けされたノードによって表された目的地点までのクエリグラフ1500の一例が示されている。また、起点駅から目的駅までの3つの乗換パターンだけが示されているが、クエリグラフが任意の数の乗換パターンを持っていて良いことは理解されるべきである。

40

#### 【0209】

この例では、各ジャーニーは、乗換パターンの凡例1501により支持されている。乗換パターンの凡例1501に示す通り、乗換パターン1は実線弧に関連付けられており、乗換パターン2は点線弧に関連付けられており、乗換パターン3は破線弧に関連付けられている。クエリグラフ1500は、乗換パターン1が{1ABC2}の乗換パターンを持つことを描いている。したがって、この乗換パターンは、起点地点から駅Aまで歩き、駅Aから駅Bまで通行し（すなわちトランジット車両又は徒歩のどちらかを使用する）、駅

50

B から駅 C まで通行し、そして、駅 C から目的地点まで歩くことを表している。一方、乗換パターン 2 は { 1 D E F G 2 } の乗換パターンを持っている。したがって、乗換パターン 2 に対しては、クエリグラフ 1 5 0 0 は、出発地点から駅 D まで歩き、駅 D から駅 E まで通行し、駅 E から駅 F まで通行し、駅 F から駅 G まで通行し、そして、駅 G から目的地点まで歩くことを示す。乗換パターン 3 は { 1 A F C 2 } の乗換パターンを持っており、トリップ 1 及び 2 と同様な方法で、記述され得る。前述の通り、各弧に関連付けられたトランジットの形式はクエリグラフによっては表現されず、クエリグラフはクエリグラフ内の駅で乗換が行われるか否かを示さない。

#### 4 . クエリグラフ拡張

【 0 2 1 0 】

クエリ処理の第 4 ステージは、随意の乗換パターンの圧縮ステップが実行され、且つ、乗換パターンの圧縮の間に乗換パターンがそれらに対応する駅のシークエンスのフィルタされたかどうかに依存して随意であることに留意されたい。しかし、前述の通り、説明の文脈では、乗換パターンがフィルタされることを想定している。クエリ処理の第 4 ステージにおいて、クエリグラフ分析 ( Q G A ) モジュール 1 2 1 は、Q G G モジュール 1 1 9 により生成されたクエリグラフを拡張する ( ステップ 1 2 0 5 ) 。前述の通り、クエリグラフは、駅を表すノード及びノードを連結するノードのみが含まれている。拡張していない形式のクエリグラフは、弧が徒歩の通行又はトランジット車両をあらわしているかどうかを表さない。クエリグラフを拡張することにより、Q G A モジュール 1 2 1 は、乗換パターン内の各駅間の可能なルートを決する。各ルートが特定時刻の乗換パターンの具体例であるから、決定される可能なルートは各駅で乗換が行われるかどうか及び到着 / 出発の時刻を表す。

【 0 2 1 1 】

図 1 6 を参照すると、クエリグラフを拡張するための、Q G A モジュール 1 2 1 の処理フローが示されている。Q G A モジュール 1 2 1 は、入力として、Q G G モジュール 1 1 9 により生成されたクエリグラフを取る ( ステップ 1 4 0 5 ) 。Q G A モジュール 1 2 1 は、起点地点と目的地点とを除いて、クエリグラフ内の各ノードを駅ノード及び乗車ノードに拡張する ( ステップ 1 6 0 0 ) 。従って、クエリグラフ内の各ノードは、拡張されたクエリグラフにおいて、2 つのノードによって表される。

【 0 2 1 2 】

次に、乗換パターンが、乗換パターン計算の間に、乗換パターン内の駅の繰り返しシーケンスのフィルタをしない乗換パターンの実施形態を検討する。この実施形態では、乗換パターンが駅で乗換が行われるか否かを示すものであるから、クエリグラフの構築は、結果として、拡張されたクエリグラフをもたらす。例えば、A B B C D F の乗換パターンにおいて、この乗換パターンは、人が駅 B に到着し駅 B でトランジット車両を出て、駅 B で別のトランジット車両に乗車することを示す。従って、この実施例では、クエリグラフは、駅 B の駅ノードと乗車ノードを持って生成される。クエリグラフは、本質的に、その拡張された形式で作成される。従って、この実施例では、クエリグラフの拡張に関して上述した教示はクエリグラフの生成の間に含まれる。

#### 5 . クエリグラフの評価

【 0 2 1 3 】

各ノードが拡張されたら、Q G A モジュール 1 2 1 は、拡張されたクエリグラフを評価する ( ステップ 1 2 0 7 ) 。一実施形態において、Q G A モジュール 1 2 1 は、直通接続データベース 1 2 5 に記憶された直通接続情報 ( ステップ 1 6 0 3 ) を使って拡張されたクエリグラフ内の駅間の直通接続を決する ( ステップ 1 6 0 1 ) 。すなわち、クエリグラフ内の乗換パターン毎に、Q G A モジュール 1 2 1 は、記憶された直通接続情報から、起点駅から後続の駅への直通接続があるかどうかを、その後続の駅から更に次に続く駅への直通接続があるかどうかを、以下同様に目的駅まで、決する。一実施形態において、受信したクエリが出発時刻を含んでいるので、Q G A モジュール 1 2 1 は、示された時刻よりも大きい又は等しい時刻に出発する直通接続のみを決する。受信したクエリが目

10

20

30

40

50

的駅への到達時刻を含んでいる場合は、QGAモジュール121は、示された時刻よりも小さい又は等しい時刻に、目的駅に到着する直通接続のみを決定する。したがって、QGAモジュール121は、クエリに関連する直通接続のみを検索する。

【0214】

一般に、QGAモジュール121は、乗換パターン内の各駅ペア間から1以上の直通接続を決定する。各直通接続は、各駅ペア間の異なるトリップを示す。例えば、トリップは、各直通接続トリップで使用されるトランジット車両の違いにより、互いに異なる。従って、直通接続は、トリップ内の各駅におけるまったく同じ到着/出発時刻を持つ駅ペア間のトリップを含むかもしれないが、トリップのために使われるトランジット車両の違いにより異なるものとなる。また、直通接続トリップは、トリップに関連付けられたスケジュール情報に基づいて差別化されてよい。例えば、単一のトランジット車両が、駅ペア間の複数の直通接続トリップを成してよく、その直通接続トリップは、各トリップのスケジュールによって特定される。

10

【0215】

例えば、乗換パターンABCを検討する。乗換パターン内の駅に関連付けられた直通接続を決定するために、QGAモジュール121は、駅Aから駅Bへの直通接続を検索する(取ってくる)。QGAモジュール121は、それから、駅Aから駅Bへの到着後に出発する、駅Bから駅Cへの直通接続を検索する。なお、駅間の直通接続は、トリップに関連付けられたトランジット車両が乗換駅に到達する前に停止する駅を本来的に示していることに留意されたい。例えば、駅Aから駅Bでは、これら駅間の直通接続は駅E及びFで停車を含む。

20

【0216】

直通接続トリップが検索されたら、QGAモジュール121は、起点駅から目的駅までのジャーニーを生成するために、直通接続を繋ぎ合わせる(スティッチ)。1以上の直通接続を縫い合わせる(つまり連結すること)は、起点駅から目的駅までのジャーニーをカバーするために、直通接続合併をすることを含む。

【0217】

例えば、再び乗換パターンABCを検討する。QGAモジュール121は、駅Aから駅Bへの直通接続と、駅Bへの到着後の出発時刻の駅Bから駅Cへの直通接続を決定する。駅Aから駅Bへの直通接続が1時00分に到着するものと仮定する。トリップを縫い合わせるためには、QGAモジュール121は、ユーザが駅Bで駅Cに向かうトランジット車両に乗り換えるために十分な時間を持つように1時00分よりもいくらか後に駅Cに向けて駅Bを離れる直通接続トリップを検索する。

30

【0218】

駅間の直通接続が検索されたら、QGAモジュール121は、1以上の直通接続により形成された乗換パターンのトリップを表すために、拡張されたクエリグラフ内のノードに弧を加える(ステップ1605)。一実施形態において、拡張されたクエリグラフは、トランジットグラフの説明と同様に4種類の弧を持ち、駅間で行われるトランジットイベントを説明するために前述したのと同じ概念を用いる。

【0219】

一実施形態において、起点駅と目的駅は、駅への歩行を表すために、それぞれ、起点駅の駅ノードと目的駅の乗車ノードとに連結される。その他の全てのノードに対して、QGAモジュール121は、直通接続クエリ経路で決定されたトリップに基づいて、前述の構成でノード間に弧を追加する。拡張されたクエリグラフにおいて各駅で行われる乗換を表すために、駅のペアに対して、QGAモジュール121は、起点駅の乗車ノードと乗換が行われる駅の駅ノードとの間に、弧を追加する。例えば、乗換パターンABCにおいて、駅Aで車両に乗り駅Bに移動し、駅Bで車両を離れてその駅Bで別の車両に乗り換える人を表すペアABを検討する。この例において、駅Aの乗車ノードは、乗換が行われることを表すために、駅Bの駅ノードに連結される。

40

【0220】

50

なお、トリップを作成するために使用される直通接続情報は、駅の出発又は到着時刻を示すスケジュール情報を持つことに留意されたい。従って、拡張されたクエリグラフ内の弧は、本質的に、スケジュール情報に関連付けられている。例えば、駅ノードAと乗車ノードBの間に2つの弧が存在する場合、第1の弧は8時00分の駅Aからの出発と12時00分の駅Bへの到着とを表し、一方、第2の弧は10時00分の駅Aからの出発と11時00分の駅Bへの到着とを表すだろう。弧は、また、直通接続トリップの間に停車が行われる駅への到着/駅からの出発を示してもよい。

#### 【0221】

図17を参照すると、図15に示すクエリグラフ1500の拡張されたクエリグラフ1700の一例が示されている。なお、トリップ凡例1701は3つのトリップを描いている：ジャーニー1、ジャーニー2、ジャーニー3である。ジャーニー1は、実線弧の経路により表されており、クエリグラフ1500の乗換パターン1に対応している。ジャーニー2及び3は、それぞれ、クエリグラフ1500の乗換パターン2、3に対応している。

#### 【0222】

クエリグラフ1700に示す通り、ジャーニー1、2及び3は、異なるタイプのノードを連結する弧の意味の説明に基づいて、駅で行われる乗換を描いている。具体的には、ジャーニー1及び3は1回の乗換を描いており、一方、ジャーニー2は2回の乗換を描いている。一例としてジャーニー1を使うと、拡張されたクエリグラフは、人は起点地点“1”から駅Aまで歩き、駅Aでトランジット車両に乗車することを描いている。トランジット車両は駅Bまで移動して、ユーザはそのトランジット車両から降りて、この乗換の実行が示された駅Bで第2の車両に乗る。駅Bから、第2のトランジット車両は駅Cまで移動し、ユーザはここで第2の車両を離れて、行き先地へ徒歩で向う。ジャーニー2に関しては、拡張されたクエリグラフは、駅Eと駅Fで乗換が行われることが描かれており、そのことは、両駅E及びFの乗車ノードと駅ノードを連結する弧によって表されている。

#### 【0223】

なお、クエリグラフ1500内に乗換パターン毎に、QGAモジュール121は、起点駅から目的駅までの1以上のジャーニーを見つけるだろう。拡張されたクエリグラフ1700において、説明の便宜上、乗換パターン毎に、1つのジャーニーのみが示されている。しかし、乗換パターン毎の駅のシーケンスは、起点駅から目的駅までの異なるトリップを表す駅間のトリップを1つ以上持ち得る。例えば、ジャーニー1に関連付けられた駅は、乗換パターン1 A<sub>1</sub> B<sub>1</sub> C<sub>1</sub> 2の間を連結する、異なる弧を3セット持っていてよく、各弧のセットは異なるトリップを表す。

### 6. 最適トリップの決定

#### 【0224】

図18を参照すると、最適ジャーニーを決定する(ステップ1209)ために、QGAモジュール121により実行される処理フローが示されている。一実施形態において、QGAモジュール121は、拡張されたクエリグラフ1607を用いて最適ジャーニーを決定する。QGAモジュール121は、クエリグラフの出発地点を初期化することにより始める(ステップ1801)。出発地点を初期化することにより、QGAモジュール121は、出発地点にコスト0をセットする。拡張されたクエリグラフ内の各弧に対して、QGAモジュール121は、コストを適用する(ステップ1803)。或るコストは起点地点から起点駅までの徒歩を表す弧に適用され、また、或るコストは目的駅から目的地点までの徒歩を表す弧にも適用される。コストは、それから、拡張されたクエリグラフ内の残りの弧に適用される。

#### 【0225】

QGAモジュール121は、直通接続テーブル1807により特定されたトリップのコストに基づいて、拡張されたクエリグラフ内の最適ジャーニーを決定するために、パレートダイクストラ計算を適用する(ステップ1805)。一実施形態において、パレートダイクストラ計算の結果は多様とみなされる複数の最適ジャーニー1809を返す。QGAモジュール121は、ジャーニーの乗換パターンが異なっていることを意味する地理

的に異なる経路を進むという点で異なっている複数の低コストジャーニーを決定する。別の実施形態では、QGAモジュール121は、多様な車両を使うジャーニーを決定する。例えば、QGAモジュール121は、バス経由の最適ジャーニー、地下鉄による最適ジャーニー、路面電車による最適ジャーニーを決定する。最後に、QGAモジュール121は、ジャーニーがルートにおいて共通しているものの、ジャーニーの行われる（つまり到着／出発する）時刻が異なるという点で、時間的に異なる最適ジャーニーを決定する。例えば、同じルートを持つ2つのジャーニーは、同じコストを持つだろうが、出発時刻は15分離れている。

#### 【0226】

別の実施形態において、最適ジャーニーを決定するためにパレートダイクストラ計算を適用するのではなく、QGAモジュール121は、別の技術を使って、拡張されたクエリグラフを探索してもよい。一実施形態において、QGAモジュール121は、例えばダイクストラアルゴリズムなど、1つの判定基準グラフ探索計算を使う。パレートダイクストラの適用は複数の判定基準でクエリグラフを評価するのに対して、ダイクストラアルゴリズムはクエリグラフを評価するために1つの判定基準のみを使うので、クエリグラフへのダイクストラアルゴリズムの適用は、最適ジャーニーのより迅速な決定をもたらす。しかし、クエリグラフを評価するためのダイクストラアルゴリズムの使用は、クエリグラフの評価に例えば時間又は車両乗換など単一の判定基準しか用いないため、パレートダイクストラ計算の適用の結果として得る最適ジャーニーのセットよりも多様性のより少ない最適ジャーニーのセットを結果としてもたらすことになる。

#### 【0227】

別の実施形態において、QGAモジュール121は、また、最適ジャーニーを決定するためのクエリグラフの評価を速くするために、更に別の技術を使用してもよい。例えば、QGAモジュール121は、最適ジャーニーを決定する速度及び／又はシステムの要求に従う結果の多様性を改善するために、当業者にA\*（Astar, エースター）探索アルゴリズムとして知られる目標指向検索、或いは、その他のユーリスティックを適用できる。

### 7. 最適ジャーニーの提供

#### 【0228】

クエリ処理の最終ステージにおいて、トリップ供給モジュール123は、ユーザに最適ジャーニーを供給する（ステップ1211）。一実施形態において、トリップ供給モジュール123は、結果ページ1900上の表示用に最適ジャーニーをフォーマットする。図19を参照すると、結果ページ1900の一例が示されている。クエリ入力部1901は、起点地点と目的地点が入力されるテキストボックスが描かれている。提案ルート部1903は、トランジットサーバ100により決定された最適ジャーニーを描いている。提案ルート部1903に示されたように、トランジットサーバ100は、同じトランジット時間長を持つが出発時間の異なる3つの最適ジャーニーを提供する。ジャーニー情報選択部1905は、ジャーニーのルートを説明する。ここに示された例では、ジャーニー1のルートが示されている。指示は、出発地点からカルトレイン（caltrain）駅まで歩くことを示している。また、図示していないが、詳細表示ボタン1907を押すことで、出発地点からカルトレイン（caltrain）駅までどうやって到達するかの指示が与えられる。それから、カルトレイン駅から、指示は、午前8時59分にマウンテンビューカルトレイン（Mountain View Caltrain）駅を出発して、午前10時02分にサンフランシスコカルトレイン（San Francisco Caltrain）駅に到着する列車233号に乗車することを示す。そしてサンフランシスコカルトレイン（San Francisco Caltrain）駅から、午前10時09分にタウンゼンド通り&4番通り（Townsend St&4thSt）から午前10時20分にサンソム通り&カリフォルニア通り（Sansome St&California St）に到着するバス10号に乗る。ユーザは、それから、行き先地点であるカリフォルニア通り555番地（555 California St）まで歩く。

### IV. むすび

## 【 0 2 2 9 】

この発明は 1 つの可能な実施形態に関する詳細を説明された。当業者たちは、この発明が別の実施形態においても実行できることを理解するだろう。第 1 に、構成要素の個々の名付け、用語の大文字使用、属性、データ構成、或いは、その他のプログラミング又は構造的様相は必須又は重要ではなく、発明を実装する機構又はその特徴は、異なる名前、形式又はプロトコルを持っていてよい。更に、システムは、前述の通りハードウェア及びソフトウェアの組み合わせで実装されてもよいし、或いは、完全にハードウェア実装されてもよい。また、本明細書に記載された種々のシステム構成要素の間の機能の個々の区分は、単に一例に過ぎず、必須ではない。単一のシステム構成要素により実行される機能は複数の構成要素により実行されてもよく、また、複数の構成要素により実行される機能は単一のシステム構成要素により実行されてもよい。

10

## 【 0 2 3 0 】

本開示のいくつかの部分は、この発明の特徴を、情報処理のアルゴリズム及び記号的表現の言葉で記述した。これらアルゴリズム的記述及び表現は、データ処理技術の当業者において、他の当業者に彼らの仕事の要旨を効果的に伝えるために、使われる手段である。これらの動作は、機能的に、又は、論理的に記述されたが、コンピュータプログラムにより実装できることは理解できる。更に、一般性を失することなしに、これらの動作の配置をモジュールとして又は機能的名前 (functional name; ファンクショナルネーム) により適用することが時として便利なことは証明されている。

20

## 【 0 2 3 1 】

特に述べられていない限り本開示から明らかな通り、上述の説明を通じて、例えば“処理する”、や“計算する (コンピューティング)”や“計算する (カリキュレイティング)”や“決定する”や“表示する”などといった用語を使用する説明は、コンピュータシステムの動作及び処理、又は、コンピュータシステムメモリ又はレジスタ又はその他の情報記憶装置、伝送又は表示装置の中で物理的 (電子的) 量として表されたデータを操作及び伝送する同様な電子計算装置に適用されることは理解される。

## 【 0 2 3 2 】

開示された発明のいくつかの側面は、ここではアルゴリズムの形式で説明された処理ステップ及び命令を含む。説明され且つ請求範囲に記載された本発明の、それら処理ステップや命令は、プログラム制御の下でコンピュータハードウェアの動作により実行され、人により行われる精神的ステップは含まない。同様に、説明され且つ請求範囲に記載された全ての種類のデータは、コンピュータシステムによって操作されるコンピュータ読み取り可能な記憶媒体に記憶されるものであり、それらは物理的実体の無い抽象的アイディアではない。

30

## 【 0 2 3 3 】

本発明は、また、ここに開示された動作を実行する装置に関する。この装置は、要求された目的のために特別に構築されたものであってもよいし、或いは、汎用コンピュータからなり、該コンピュータによって実行可能な、コンピュータ読み取り可能な媒体に記憶されたコンピュータプログラムにより、選択的に作動又は再構築される。かかるコンピュータプログラムは、例えば、以下の例示に限定されないが、フロッピーディスク (登録商標)、光学ディスク、CD-ROM、光磁気ディスク、リードオンリーメモリ (ROM)、ランダムアクセスメモリ (RAM)、EPROM、EEPROM、磁気又は光学カード、ASIC (特定用途向け集積回路) を含む任意の種類のディスクなど、コンピュータ読み取り可能な媒体、又は、電子的命令の記憶に適した任意の種類の媒体に記憶され、それぞれコンピュータシステムバスに接続される。更に、詳細な説明で言及したコンピュータは、単一のプロセッサからなるものであってもよいし、又は、計算処理能力を向上させるために複数のプロセッサを使用する構造であってもよい。

40

## 【 0 2 3 4 】

ここに開示したアルゴリズム及び動作は、任意の種類の又はブランドのコンピュータ又

50

はその他の装置により実行され得る。種々の汎用システムが、ここに示された技術に従うプログラムとともに使用されてよく、又は、要求された方法ステップを実行するためのより専用の装置の構成が便利なのは証明されているだろう。それら多様なシステムのために要求される構造は、等価な変形例とともに、当業者にとって明らかであろう。加えて、本発明は、どの特定のプログラミング言語に関しても記述されていない。種々のプログラミング言語がここに記載された通りこの発明の技術を実装するために使用されてよいこと、及び、いずれの特定言語に言及することも本開示を可能にすること、及び、この発明の最良の形態のために提供されていることは、理解されるだろう。

#### 【 0 2 3 5 】

この発明は、多数形態にわたる広く多様なコンピュータネットワークシステムに適している。この分野において、大規模ネットワークの構成及び管理は、例えばインターネットなどネットワーク経由で、異種のコンピュータ及び記憶装置と通信可能に接続された記憶装置及びコンピュータからなる。

#### 【 0 2 3 6 】

最後に、この明細書で使用された言葉は、原則的に、読み易さと説明の目的で選択されており、発明の主題を表現又は制限するために選択されたのではない。従って、この発明の開示は、説明的であることを意図しており、以下に述べる特許請求の範囲の発明の範囲を制限することを意図していない。

#### 【 符号の説明 】

#### 【 0 2 3 7 】

1 0 0    トランジットサーバ、 1 0 1    フロントエンドインターフェース、 1 0 3    事前計算モジュール、 1 0 5    トランジットグラフモジュール、 1 0 7    グローバル駅選択モジュール、 1 0 9    乗換パターン計算モジュール、 1 1 1    乗換パターン圧縮モジュール、 1 1 3    直通接続モジュール、 1 1 5    クエリ解決モジュール、 1 1 7    ローカル駅決定モジュール、 1 1 9    クエリグラフ生成モジュール、 1 1 9    モジュール、 1 2 1    クエリグラフ分析モジュール、 1 2 3    トリップ供給モジュール、 1 2 5    ルート情報データベース、 1 2 7    トランジット情報データベース、 1 2 9    トランジット情報インタフェース、 1 3 1    クライアント装置    1 3 3    ブラウザ、 1 3 5    乗換パターン構築モジュール、 1 3 7    ネットワーク、 4 0 0    トランジットグラフ、 4 0 3    ノード、 4 0 5 , 4 0 9    弧、 4 0 7    コスト、 1 5 0 0    クエリグラフ、 1 5 0 1    乗換パターン凡例、 1 7 0 0    クエリグラフ、 1 7 0 1    トリップ凡例、 1 9 0 0    結果ページ、 1 9 0 1    クエリ入力部、 1 9 0 3    提案ルート部、 1 9 0 5    ジャーニー情報選択部、 1 9 0 7    詳細表示ボタン

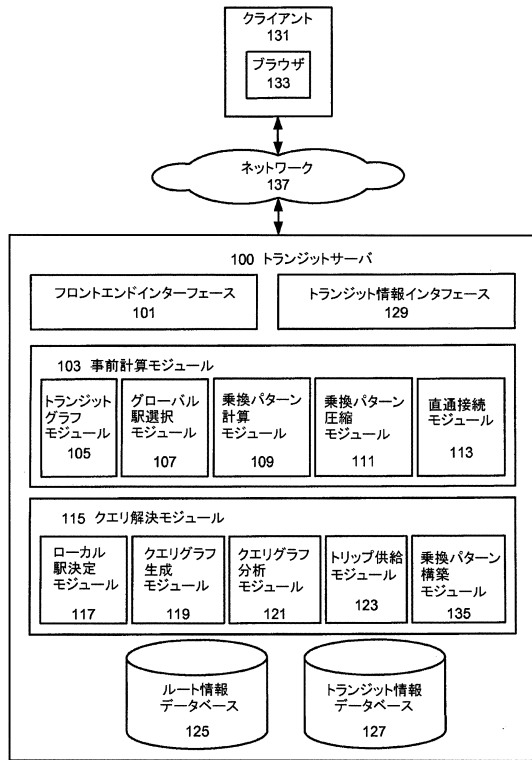
10

20

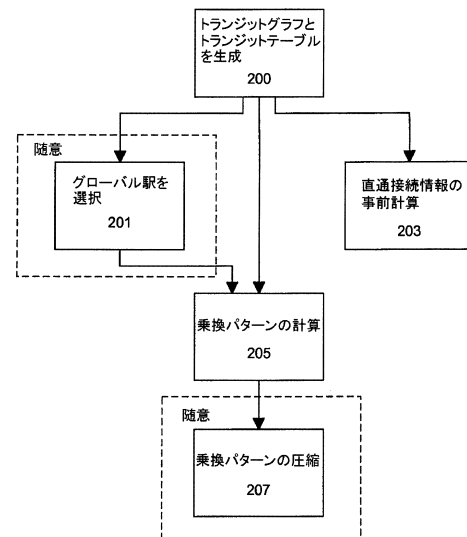
30



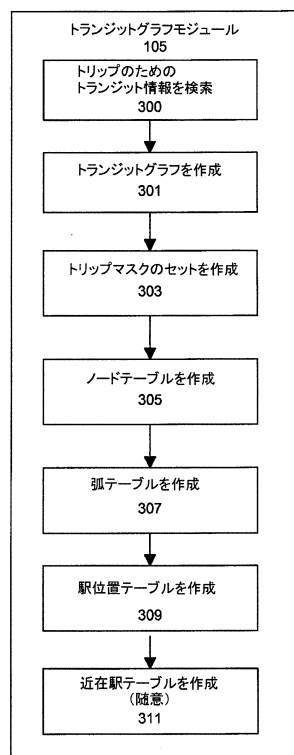
【図 1】



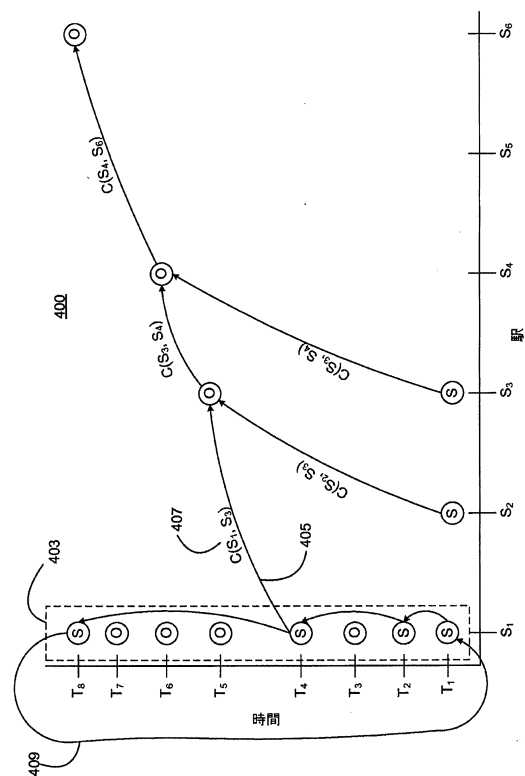
【図 2】



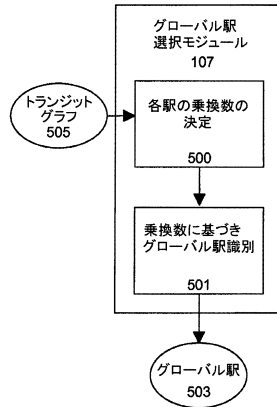
【図 3】



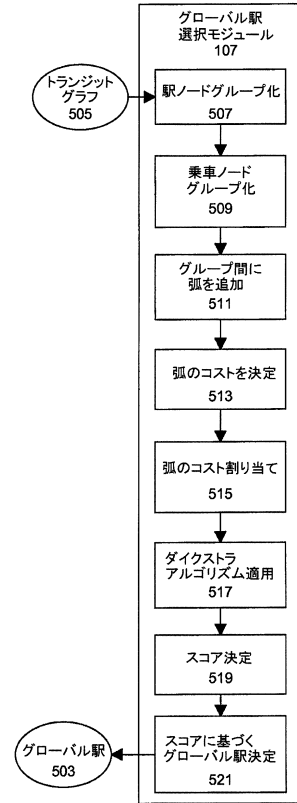
【図 4】



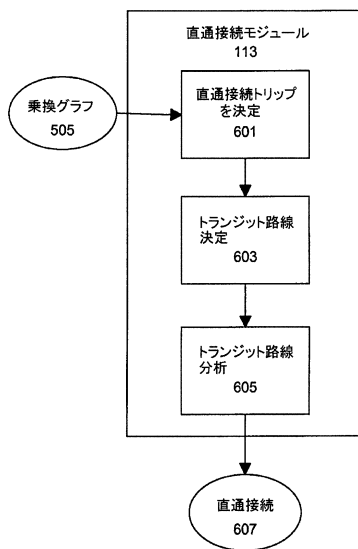
【図 5 A】



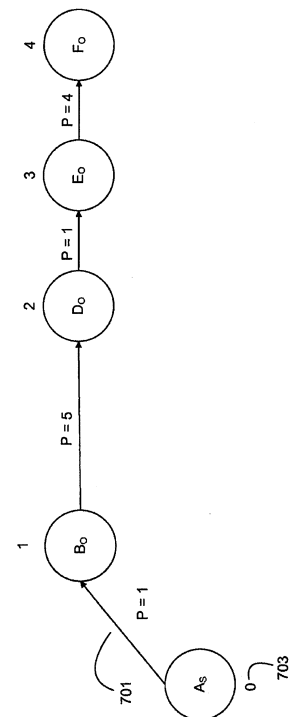
【図 5 B】



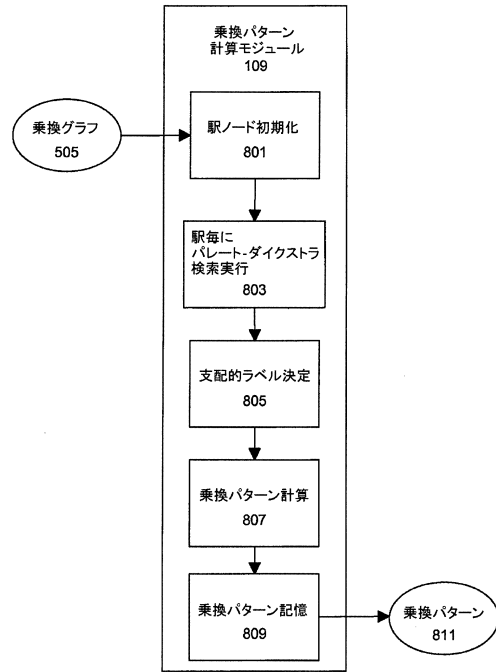
【図 6】



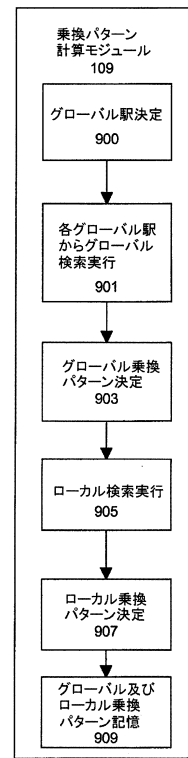
【図 7】



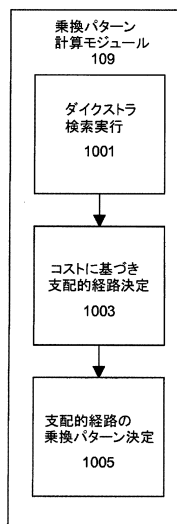
【図 8】



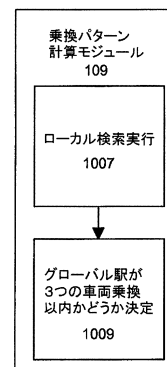
【図 9】



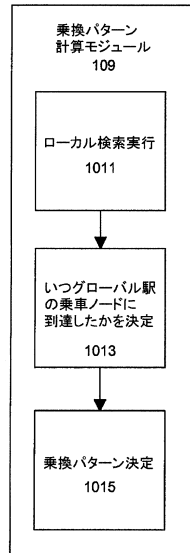
【図 10 A】



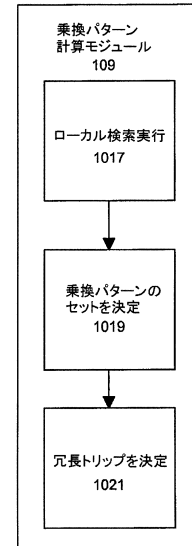
【図 10 B】



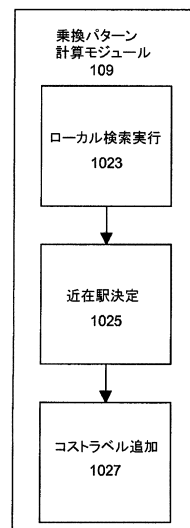
【図 10 C】



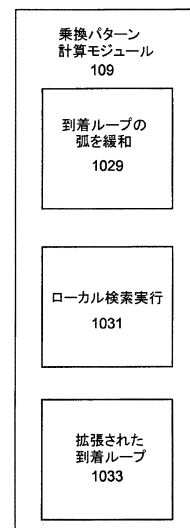
【図 10 D】



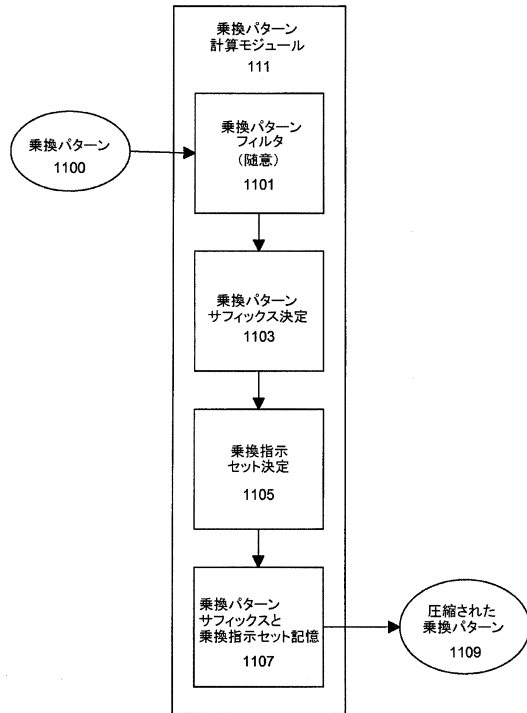
【図 10 E】



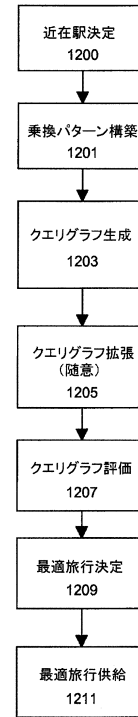
【図 10 F】



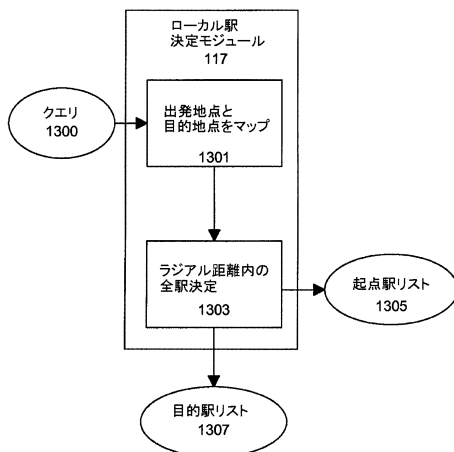
【図 1 1】



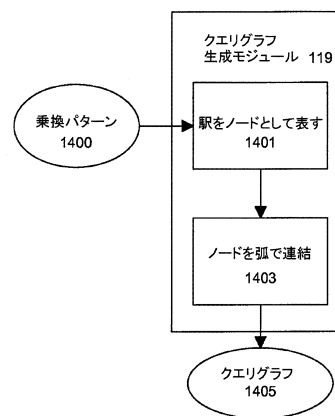
【図 1 2】



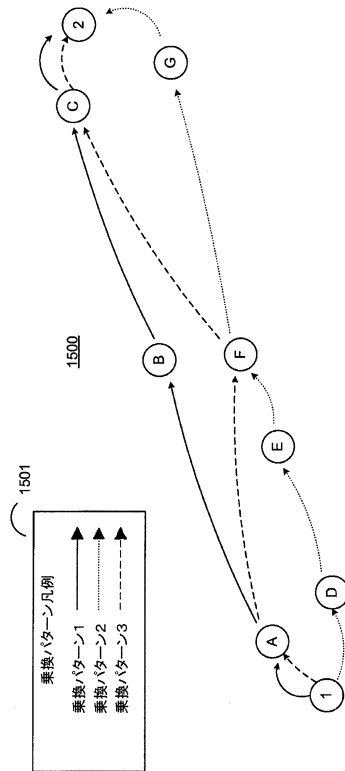
【図 1 3】



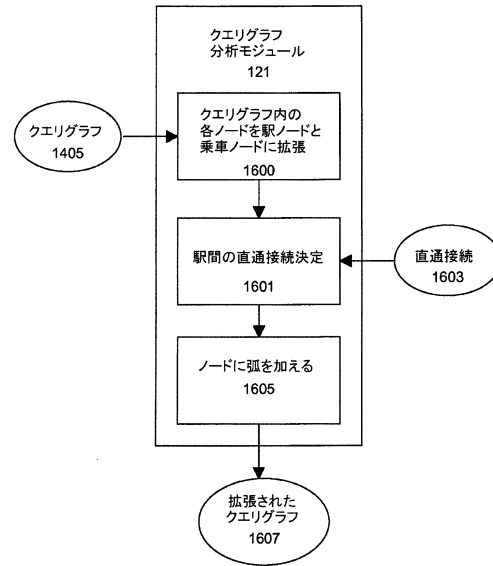
【図 1 4】



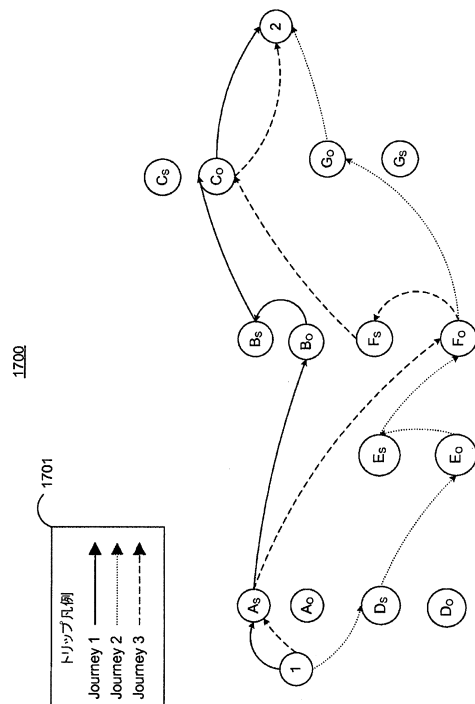
【図 15】



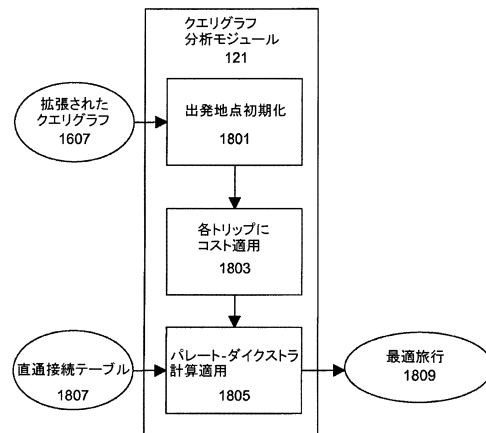
【図 16】



【図 17】



【図 18】



【 19 】

1900

Get Directions: My Maps

1901

801 California St. Mt. View, CA  
1555 California Street, San Francisco, CA  
Add Destination - Show options  
By public transit  
Get Directions

Also available: By car

Transit directions to 1555 California St, San Francisco, CA 94104

1903

Suggested routes - Show options

8:51am - 10:23am  
1 hour 32 mins  
9:21am - 10:56am  
1 hour 35 mins  
9:51am - 11:26am  
1 hour 34 mins

801 California St  
Mountain View, CA 94041

Showing Trip 1 Travel time: About 1 hour 32 mins  
\$7.50 (or \$22.54 driving)

1905

Walk to Mountain View Caltrain  
About 6 mins

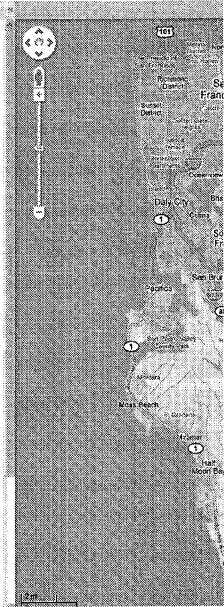
Show details

Train - Limited - 233 - Direction: Tamien to San Francisco  
Service run by Caltrain  
8:50am Depart Mountain View Caltrain  
1 hour 2 mins  
10:02am Arrive San Francisco Caltrain

Bus - 10 - Townsend - Direction: Van Ness & North Point  
7 mins to make transfer  
Service run by San Francisco Municipal Transportation Agency  
10:09am Depart Townsend St & 4th St  
15 mins  
10:26am Arrive Sansome St & California St

Walk to 1555 California St  
About 4 mins

Show details



## フロントページの続き

- (72)発明者 バスト, ハンナ  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 カールソン, エリク  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 アイゲンブリク, アルノー  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 ガイスベルガー, ローベルト  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 ハレルソン, クリス  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 レイチェフ, ベセリン  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内
- (72)発明者 ビゲル, ファビアン  
アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600, グーグル インコーポレイテッド内

審査官 青柳 光代

- (56)参考文献 特開2006-193020(JP, A)  
特開2012-012014(JP, A)  
特開2009-014545(JP, A)  
特開2006-047274(JP, A)  
特開2004-212056(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06Q 10/00-50/34  
B61L 25/02