(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2008-157698
(P2008-157698A)
(43)公開日　平成20年7月10日(2008.7.10)

(51)Int.Cl.		F I			テーマコード（参考）
G01C 21/00	(2006.01)	G01C	21/00	G	2C032
G08G 1/0969	(2006.01)	G08G	1/0969		2F129
G09B 29/00	(2006.01)	G09B	29/00	A	5H180

審査請求 未請求　請求項の数 19　ＯＬ　（全 22 頁）

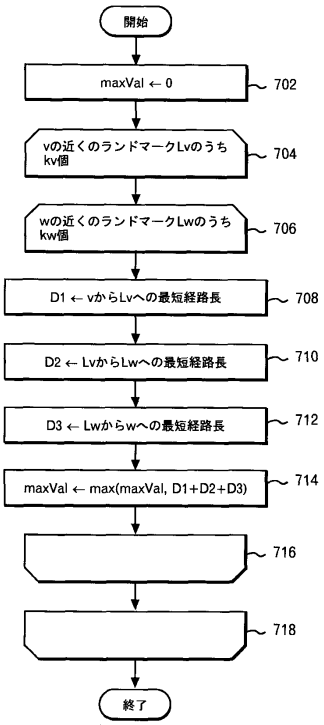
(21)出願番号　特願2006-345224 (P2006-345224)
(22)出願日　平成18年12月22日 (2006.12.22)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】経路探索方法、プログラム及びシステム

(57)【要約】
【課題】　Ａ＊探索技法において、上限値と下限値の精度を向上させることにより、探索速度を向上させること。
【解決手段】　経路探索される、もととなる重み付きグラフに対して、複数のランドマークを選択し、ランドマーク間の最短経路長、及び各頂点から、近くのランドマークへの最短経路長を計算し、後で参照できるように、記憶装置に記録する。
また、２つの頂点v,wとそれにそれぞれ近接する２つのランドマークの２つの頂点からなる四角不等式から導かれる式に基づき、v,wに対応する最短経路長の上下限値を計算するルーチンが用意される。これらのルーチンは、Ａ＊探索プログラムからの呼び出しに応じて、予め記憶装置に記録されている、ランドマーク間の最短経路長及び、各頂点から近くのランドマークへの最短経路長のデータを参照して、v,wに対応する最短経路長の上限値または下限値を返す。
【選択図】　図７



【特許請求の範囲】
【請求項１】
記憶手段をもつコンピュータにより、複数の頂点をもつ重み付きグラフ上で、頂点v,w間の最短経路長d(v,w)の上限値と下限値を利用して経路探索するための、グラフ上でのデータ処理プログラムであって、
コンピュータにより、
前記グラフの頂点のうちの複数の頂点をランドマークとして選択する機能と、
前記ランドマークとして選択された頂点Lv, Lw間の最短経路長d(Lv, Lw)の値を計算して、前記記憶手段に記録する機能と、
前記グラフの頂点vの、最短経路長に関して近くの１つまたは複数のランドマークLvを選択して、前記頂点vから前記ランドマークLvへ最短経路長d(v, Lv)の値を計算して、前記記憶手段に記録する機能と、
前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算し、d(v,w)の上限値として提供する機能と、
前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算し、d(v,w)の下限値として提供する機能を実現する、
プログラム。
【請求項２】
前記グラフの頂点の総数をＮとしたとき、前記ランドマークの個数が、$N^{1/2}$個以上選ばれる、請求項１のプログラム。
【請求項３】
前記グラフの頂点vの、最短経路長に関して近くのランドマークとして、４個のランドマークが選ばれる、請求項１のプログラム。
【請求項４】
前記上限値または下限値の計算ステップにおいて、前記グラフの頂点v,wに関して、該頂点vに関連して４個のランドマークが選ばれ、該頂点wに関連して１個のランドマークが選ばれる、請求項１のプログラム。
【請求項５】
コンピュータにより、重み付きグラフ上で、頂点v,w間の最短経路長d(v,w)の上限値と下限値を利用して経路探索するための、グラフのデータ処理システムであって、
データを読み出し可能に記録可能な記憶手段と、
グラフの全頂点データを読み出し可能に保持する頂点データ保持手段と、
前記頂点データ保持手段から頂点データを読み取り、前記グラフの頂点のうちの複数の頂点をランドマークとして選択する手段と、
前記選択されたランドマークの情報に基づき、前記ランドマークとして選択された頂点Lv, Lw間の最短経路長d(Lv, Lw)の値を計算し、前記記憶手段に記録する手段と、
前記全頂点データと、前記選択されたランドマークの情報に基づき、前記グラフの頂点vの、最短経路長に関して近くの１つまたは複数のランドマークLvを選択して、前記頂点vから前記ランドマークLvへ最短経路長d(v, Lv)の値を計算し、前記記憶手段に記録するする手段と、
前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算する手段と、
前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前

記記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算する手段とを有する、
データ処理システム。
【請求項６】
前記グラフの頂点の総数をＮとしたとき、前記ランドマークの個数が、$N^{1/2}$個以上選ばれる、請求項５のデータ処理システム。
【請求項７】
前記グラフの頂点vの、最短経路長に関して近くのランドマークとして、４個のランドマークが選ばれる、請求項５のデータ処理システム。
【請求項８】
前記上限値または下限値の計算ステップにおいて、前記グラフの頂点v,wに関して、該頂点vに関連して４個のランドマークが選ばれ、該頂点wに関連して１個のランドマークが選ばれる、請求項５のデータ処理システム。
【請求項９】
記憶手段をもつコンピュータにより、重み付きグラフ上で、頂点v,w間の最短経路長d(v,w)の上限値と下限値を利用して経路探索するための、グラフ上でのデータ処理方法であって、
前記コンピュータの処理により、前記グラフの頂点のうちの複数の頂点をランドマークとして選択するステップと、
前記コンピュータの処理により、前記ランドマークとして選択された頂点Lv, Lw間の最短経路d(Lv, Lw)を計算して、前記記憶手段に記録するステップと、
前記コンピュータの処理により、前記グラフの頂点vの、最短経路長に関して近くの１つまたは複数のランドマークLvを選択して、前記頂点vから前記ランドマークLvへ最短経路d(v, Lv)を計算して、前記記憶手段に記録するステップと、
前記コンピュータの処理により、前記グラフの頂点v,wにつき、前記頂点v関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算し、d(v,w)の上限値として提供するステップと、
前記コンピュータの処理により、前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された、１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算し、d(v,w)の下限値として提供するステップとを有する、
データ処理方法。
【請求項１０】
前記グラフの頂点の総数をＮとしたとき、前記ランドマークの個数が、$N^{1/2}$個以上選ばれる、請求項９の方法。
【請求項１１】
前記グラフの頂点vの、最短経路長に関して近くのランドマークとして、４個のランドマークが選ばれる、請求項９の方法。
【請求項１２】
前記上限値または下限値の計算ステップにおいて、前記グラフの頂点v,wに関して、該頂点vに関連して４個のランドマークが選ばれ、該頂点wに関連して１個のランドマークが選ばれる、請求項９の方法。
【請求項１３】
前記複数の頂点をランドマークとして選択するステップが、
グラフに配置された頂点を全て含む最小の矩形を取るステップと、
矩形を縦横それぞれの辺に平行な、矩形の中心を通る２本の直線で田の字に４等分するステップと、
分割されたそれぞれの領域が規定数以上の頂点を含んでいれば、再度４等分するステッ

プを繰り返し、前記ステップの手続きが終了後、各小領域の中心部に最も近い頂点をランドマークとして選択するステップを有する、請求項９の方法。

【請求項１４】

コンピュータにより、重み付きグラフ上で、経路探索するためのシステムにおいて、

前記グラフの全頂点のデータを読み出し可能に記録するグラフ頂点記憶手段と、

前記グラフの全頂点のうち、ランドマークとして選択された頂点Lv, Lw間の最短経路長d(Lv, Lw)の値を読み出し可能に記録するランドマーク記憶手段と、

前記グラフの頂点vと、該頂点vの、１つまたは複数の近傍のランドマークLwの間の最短経路長d(v, Lw)の値を読み出し可能に記録する近傍ランドマーク記憶手段と、

前記グラフの頂点v,wにつき、前記頂点v関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記ランドマーク記憶手段と前記近傍ランドマーク記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算しその値を返す上限値計算手段と、

前記グラフの頂点v,wにつき、前記頂点vに関して選択された１つまたは複数のランドマークLvと、前記頂点wに関して選択された１つまたは複数のランドマークLwに関して、前記ランドマーク記憶手段と前記近傍ランドマーク記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算しその値を返す下限値計算手段と、

グラフの始点と終点を受け取り、前記上限値計算手段と前記下限値計算手段を呼び出すことにより、頂点v,w間の最短経路長d(v,w)の上限値と下限値を利用してＡ＊探索を行い、該始点と終点の間の最短経路を含む探索結果を出力する経路探索手段とを有する、

システム。

【請求項１５】

前記グラフの始点と終点を受け取り、前記下限値計算手段を呼び出すことにより、前記グラフの始点から終点への下限値と、前記グラフの終点から始点への下限値を計算して比較し、前記グラフの始点から終点への下限値が、前記グラフの終点から始点への下限値より大きいことに応答して、前記始点と終点とを入れ替えて前記経路探索手段に渡す探索方向決定手段をさらに有する、請求項１４のシステム。

【請求項１６】

複数のランドマークを選択された重み付きグラフ上で、頂点と前記ランドマークの間の最短経路長を記録する記憶手段をもつコンピュータにより、経路探索するためのプログラムであって、

コンピュータにより、

グラフの頂点v,wにつき、前記頂点vの近傍として選択された１つまたは複数のランドマークLvと、前記頂点wの近傍として選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算しその値を返す機能と、

前記グラフの頂点v,wにつき、前記頂点vの近傍として選択された１つまたは複数のランドマークLvと、前記頂点wの近傍として選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算しその値を返す機能と、

グラフの始点と終点を受け取り、前記上限値を計算しその値を返す機能と、前記下限値を計算しその値を返す機能を呼び出すことにより、頂点v,w間の最短経路長d(v,w)の上限値と下限値を利用してＡ＊探索を行い、該始点と終点の間の最短経路を含む探索結果を出力する経路探索機能とを実現する、

プログラム。

【請求項１７】

前記コンピュータにより、

前記グラフの始点と終点を受け取り、前記下限値計算手段を呼び出すことにより、前記グラフの始点から終点への下限値と、前記グラフの終点から始点への下限値を計算して比較し、前記グラフの始点から終点への下限値が、前記グラフの終点から始点への下限値よ

り大きいことに応答して、前記始点と終点とを入れ替えて前記経路探索機能に渡す探索方向決定機能をさらに実現する、請求項１６のプログラム。
【請求項１８】
複数のランドマークを選択された重み付きグラフ上で、頂点と前記ランドマークの間の最短経路長を記録する記憶手段をもつコンピュータにより、経路探索するための経路探索方法であって、
コンピュータの処理により、グラフの頂点v,wにつき、前記頂点vの近傍として選択された１つまたは複数のランドマークLvと、前記頂点wの近傍として選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限値を計算しその値を返す上限値計算ステップと、
コンピュータの処理により、前記グラフの頂点v,wにつき、前記頂点vの近傍として選択された１つまたは複数のランドマークLvと、前記頂点wの近傍として選択された１つまたは複数のランドマークLwに関して、前記記憶手段から値を読み出すことにより、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限値を計算しその値を返す下限値計算ステップと、
コンピュータの処理により、グラフの始点と終点を受け取り、前記上限値計算ステップと前記下限値計算ステップにより、頂点v,w間の最短経路長d(v,w)の上限値と下限値を計算してＡ＊探索を行い、該始点と終点の間の最短経路を含む探索結果を出力するステップとを有する、
方法。
【請求項１９】
前記グラフの始点と終点を受け取り、前記下限値計算手段を呼び出すことにより、前記グラフの始点から終点への下限値と、前記グラフの終点から始点への下限値を計算して比較し、前記グラフの始点から終点への下限値が、前記グラフの終点から始点への下限値より大きいことに応答して、前記始点と終点とを入れ替えて前記経路探索を出力するステップに渡すステップをさらに有する、請求項１８の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明は、カーナビゲーション・システムなどで利用される経路探索技法に関し、より詳しくは、経路探索を高速化するための方法、プログラム及びシステムに関するものである。
【背景技術】
【０００２】
近年、カーナビゲーション・システムや、ウェブサイトにおける最適交通路探索サービスの増加により、コンピュータにより経路を探索する、より高速化された技法への要望が高まっている。このことは、コンピュータ・サイエンス的には、重み付きグラフにおける、最短経路の決定アルゴリズムに帰着される。
【０００３】
ところで、重み付きグラフにおける、最短経路の決定アルゴリズムの、古典的とも言える技法として、ダイクストラ法と呼ばれるものが知られている。これは要するに、重み付きグラフＧにおいて、頂点ａから頂点ｚへの最短経路を決定するにあたって、ａからある他の頂点への最短経路を決定し、次に、そうして見つかった他の頂点を対象から外して、残りの頂点に対して同様のことを行い、残りの頂点がなくなったとき、手続を完了する、というものである。このアルゴリズムは、必ず解が存在することを保証するが、グラフＧの頂点の数が増えてくると、計算量の点で、現実的なものでなくなる。
そこで、計算量の点でより改善された、さまざまな技法が提案されてきた。そのようなものの１つとして、Ａ＊探索と呼ばれる技法がある。これは、予め計算された任意の２頂点間の最短経路長の上下限値を利用して、最短経路の計算を高速化するものであり、具体的には、下記のような技法である。
【０００４】

以下で、頂点 v, w について d(v,w) は、vからwへの最短経路長を表すものとする。また、w(v,w)を、vとwを結ぶ枝の長さとする。

先ず、初期設定として、始点sから終点tまでの最短経路を求めたい場合、各頂点vについて以下のように定める。

k(v): v=s のとき k(s):=π(s)、v≠sのときk(v):=∞;
d(v): v=s のとき d(v):=0、v≠sのときd(v):=∞;
prev(v) = NULL;
頂点集合 S = {s};

そうして、次のようにして探索処理を行う。

```
while( Sが空でない ) {
    S から k(v)の値が最も小さい頂点vを除去;
    for(vから出ている全ての枝(v,w) について以下の操作を行う) {
        if( k(w) > k(v) + w(v,w) - π(v) + π(w)) {
            k(w) := k(v) + w(v,w) -π(v)+π(w);
            d(w) := d(v) + w(v,w);
            prev(w) := v;
            if (d(w)+π(w) <= min(d(s,t) の上限値, d(t)))
              { S ← S ∪{w}; }
        }
    }
}
```

最後に、d(t) に入っている値が最短経路長になる。

また、prev(t) に入っている頂点を t'、prev(t')に入っている頂点を t'' ・・・とすると、 t←t'←t''←・・・←s が最短経路になる。

【０００５】

任意の２頂点 v, w について、d(v,w) の上限値と下限値が正しく与えられるとき、上記のＡ＊探索が正しく最短経路を求めることが知られている。また、Ａ＊探索に与える上下限値の精度が良ければ良いほど終点まで速く到達できることが知られている。直感的には、アルゴリズム中のk(v) には、始点sからvを経由して終点tに到達する経路の最短経路長の下限値、d(v)には始点 s からvまでの最短経路長が格納される。

【０００６】

そこで、上記のＡ＊探索アルゴリズムでブラックボックスになっている部分（d(v,w)の上下限値を計算する方法）が、従来より課題になっている。

【０００７】

これに関連して先ず、Computing the Shortest Path: A* Search Meets Graph Theory by A. V. Goldberg and C. Harrelson, in Proceedings of 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05), Vancouver, Canada, 2005.という文献に記載された手法がある。

この手法では、グラフ上の少数（十数個）の頂点を「ランドマーク」として選択し、ランドマークから各頂点への最短経路長を記憶しておく。そして、記憶しておいた最短経路長を利用して任意の２頂点間の最短経路長の下限値を算出し、Ａ＊探索による探索を高速化する手法である。

【０００８】

また、Goal Directed Shortest Path Queries Using Precomputed Cluster Distances by Jens Maue, Peter Sanders, Domagoj Matijevic, in Proceedings of the 5th International Workshop on Experimental Algorithms (WEA-06), LNCS 4007, 316-327, Springer, 2006.も知られている。

この手法では、グラフを多数のクラスターに分割し、各クラスター間の最短経路長を定義し、記憶しておく。そして、記憶しておいた最短経路長を利用して、任意の２頂点間の

最短経路長の上限値と下限値を算出し、最短経路長の上限値による枝刈りとＡ＊探索を組み合わせて高速化する手法である。

【０００９】

いずれの手法も、Ａ＊探索で利用する「任意の２頂点間の最短経路長の上下限値」の精度がアルゴリズムの高速化に大きく寄与している。[1]の手法では、下限値計算時に複数のランドマーク情報を利用するが、ランドマーク自体の数が少ないことが欠点である。[2]の手法では、クラスターの数自体は沢山あるが、上下限値計算時に利用するクラスター情報の数が少ないことが欠点である。

【００１０】

この他にも、従来技術として特開２００６－２０１１７４という特許公開公報に開示されている技術があるが、その発明者の一人は、上記文献[1]にも著者としてリストされており、その開示内容も、上記文献[1]とほぼ重なるものである。

【非特許文献１】[1] Computing the Shortest Path: A* Search Meets Graph Theory by A. V. Goldberg and C. Harrelson, in Proceedings of 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05), Vancouver, Canada, 2005.

【非特許文献２】[2] Goal Directed Shortest Path Queries Using Precomputed Cluster Distances by Jens Maue, Peter Sanders, Domagoj Matijevic, in Proceedings of the 5th International Workshop on Experimental Algorithms (WEA-06), LNCS 4007, 316-327, Springer, 2006.

【特許文献１】特開２００６－２０１１７４

【発明の開示】

【発明が解決しようとする課題】

【００１１】

この発明の目的は、経路探索を高速化するための技法を提供することにある。

より詳しくは、この発明の目的は、Ａ＊探索を用いた経路探索において、任意の２頂点間の最短経路長の上下限値の精度を向上させることによって、経路探索を高速化することにある。

【課題を解決するための手段】

【００１２】

本願発明者は、研究を重ねた結果、下記のような手法により、経路探索が高速化されることに想到した。

すなわち、先ず、経路探索される、もととなる重み付きグラフに対して、ランドマークを生成し、上下限値の計算に必要な情報が予め、計算される。

次に、最短経路長の上下限値が計算される。特に、本発明は、この計算のために、頂点ｖの近くのランドマークＬｖと、頂点ｗの近くのランドマークＬｗの４点で構成される四角不等式に着目するものであり、下記のような式が、上下限値の計算に利用される。

d(v,w)の下限値：d(v,w)>=d(Lv,Lw)-d(Lv,v)-d(w,Lw)

d(v,w)の上限値：d(v,w)<=d(v,Lv)+d(Lv,Lw)+d(Lw,w)

これらの不等式の右辺の値は全て、上記のランドマークの生成に関連して、予め計算され保存されていることに留意されたい。

この上限値と、下限値の計算ルーチンは、後のＡ＊探索に適宜呼び出されて利用される。

【発明の効果】

【００１３】

この発明によれば、最短経路長の上下限値の精度の向上が達成され、この結果、経路探索の時間が短縮される。発明者の実験によれば、特定のグラフの探索に関して、上記文献[2]の手法よりも、４倍以上の探索性能の向上が確認された。

【発明を実施するための最良の形態】

【００１４】

以下、図面に基づき、この発明の実施例を説明する。特に断わらない限り、同一の参照

番号は、図面を通して、同一の対象を指すものとする。尚、以下で説明するのは、本発明の一実施形態であり、この発明を、この実施例で説明する内容に限定する意図はないことを理解されたい。

【００１５】

図１は、本発明の手法を実施するためのハードウェア構成のブロック図である。図１において、コンピュータ１００は、ＣＰＵ１０２と、メイン・メモリ１０４をもち、これらは、バス１０６に接続されている。メイン・メモリ１０４は、少なくとも５１２ＭＢ、好適には、１ＧＢ以上の記憶容量をもつものである。後述するグラフの全頂点データなどをロードするためには、メイン・メモリ１０４の容量は、大きい程好ましい。ＣＰＵは、好適には、３２ビットまたは６４ビットのアーキテクチャに基づくものであり、例えば、インテル社のＰｅｎｔｉｕｍ（Ｒ）　４、ＡＭＤ社のＡｔｈｌｏｎ（Ｒ）などを使用することができる。バス１０６には、ディスプレイ・コントローラ１０８を介して、ＬＣＤモニタなどのディスプレイ１１０が接続される。ディスプレイ１１０は、本発明に係る、処理用に書かれたプログラムを眺めたり、編集したりするために使用される。また、本発明に従い、探索した経路を表示するためにも使用される。バス１０６にはまた、ＩＤＥコントローラ１１２を介して、ハードディスク１１４と、ＣＤ－ＲＯＭドライブ１１６が接続される。ハードディスク１１４には、オペレーティング・システム、コンパイラその他のプログラムが、メインメモリ１０４にロード可能に記憶されている。ハードディスク１１４は、少なくとも４０ＧＢ、好適には、１６０ＧＢ以上の記憶容量をもつものである。ＣＤ－ＲＯＭドライブ１１６は、必要に応じて、ＣＤ－ＲＯＭからプログラムをハードディスク１１４に追加導入するために使用される。バス１０６には更に、キーボード・マウスコントローラ１１８を介して、キーボード１２０と、マウス１２２が接続されている。キーボード１２０及びマウス１２２は、コンパイラを使用して、本発明に係る処理を実行させるプログラムを書いたり、実行させたりするために使用される。

【００１６】

コンピュータ１００のソフトウェア的な環境について更に説明すると、上記オペレーティング・システムとしては、Ｌｉｎｕｘ（Ｒ）、マイクロソフト社のＷｉｎｄｏｗｓ　ＸＰ（Ｒ）、Ｗｉｎｄｏｗｓ（Ｒ）２０００、アップルコンピュータのＭａｃ　ＯＳ（Ｒ）などを使用することができるが、ここに掲げたオペレーティング・システムには限定されない。本発明の処理用のプログラムを作成するためのコンパイラは、使用するオペレーティング・システムの上で、動作し得る、マイクロソフト社のＶｉｓｕａｌ　Ｂａｓｉｃ（Ｒ）、Ｖｉｓｕａｌ　Ｃ＋＋（Ｒ）、Ｖｉｓｕａｌ　Ｓｔｕｄｉｏ（Ｒ）、ボーランド社のＢｏｒｌａｎｄ（Ｒ）Ｃ＋＋　Ｃｏｍｐｉｌｅｒ　５．５、Ｄｅｌｐｈｉ（商標）、Ｃ＋＋　Ｂｕｉｌｄｅｒ（商標）、ＩＢＭ社のＶｉｓｕａｌＡｇｅ　ｆｏｒ　Ｊａｖａ（商標）などがあり、これらの任意のものを使うことができる。コンパイラは、ＧＵＩをサポートしておらず、ＣＵＩベースだけのものでも、本発明の機能を実現することができる。

【００１７】

図２は、この実施例の高レベル機能ブロック図である。図２において、グラフ全頂点データ２０２は、重み付きグラフを、コンピュータ可読な形式で、ハードディスク１１４に保存したものであり、各頂点に対応するデータを保持する。経路探索アプリケーションの場合、重み付きグラフの各頂点は、各々が、各地点に対応する。また、この実施例では、グラフは、無向グラフである。

【００１８】

グラフの各頂点には、好ましくは、１から連番で番号が付与され、そのデータ形式は、図３に示すように、各頂点毎に、隣接する頂点の番号のリストへのリンクと、隣接する頂点への距離のリストへのリンクとを含む。尚、このようなデータ形式は、可能ないくつかの選択肢のうちの１つであり、本発明が想定するグラフ表現のデータ形式は、これに限定されるものではないことを、当業者なら理解するはずである。

【００１９】

図４には、具体的なグラフの頂点の配置の例が示されている。すなわち、図４では、ｎ

番目の頂点は、ｎを丸で囲って表示されている。図４で、頂点１は、隣接する頂点として、頂点２、５及び８をもち、それぞれへの距離は、4.0, 5.0, 2.5である。頂点２は、隣接する頂点として、頂点１、４及び５をもち、それぞれへの距離は、4.0, 3.5, 3.0である。図３には、ハードディスク１１４から、メイン・メモリ１０４上に、前処理部２１２が処理可能にロードされた、頂点１及び頂点２のデータ構造が示されている。

これを、例えば、Ｃ言語の構造体で記述すると、下記のようになる。

```
struct Node  {
 int NodeID;  // 頂点番号
 int *adjNodeList[]; // 隣接する頂点の頂点番号を配列として記憶
 double *adjDistanceList[]; // 隣接する頂点までの距離を配列として記憶
};
```

図４の頂点１に、この構造体を適用すると、NodeID = 1, *adjNodeList = {2,5,8}, *adjDistanceList = {4.0, 5.0, 2.5} となる。

【００２０】

図２に戻って、前処理部２１２、上限値計算部２１４、下限値計算部２１６、探索方向決定部２１８、及びクエリー処理部２２０は、この実施例に処理を行うように、好適には、ＣまたはＣ＋＋で書かれ、ハードディスク１１４に記憶されているプログラムであり、キーボード１２０またはマウス１２２の操作により、オペレーティング・システムの動作で、ハードディスク１１４から、メイン・メモリ１０４にロードされる際に、ＣＰＵ１０２の働きで、実行されるものである。

【００２１】

また、好適には、前処理部２１２、上限値計算部２１４、下限値計算部２１６、探索方向決定部２１８、及びクエリー処理部２２０がメイン・メモリ１０４に際に、または、後のユーザーの操作に応答して、グラフ全頂点データ２０２が、ハードディスク１１４から、図３に示すようなグラフ全頂点データ２０２'として、例えば、Ｃ言語のmalloc()のような関数を呼び出してメモリ領域を確保しながら、メイン・メモリ１０４にロードされる。このとき、後の処理の高速化のために、グラフ全頂点データ２０２の全体をメイン・メモリ１０４にロードすることが望ましい。

【００２２】

しかし、メイン・メモリ１０４の容量が十分でない場合、あるいは、グラフ全頂点データ２０２がの頂点の数が非常に多い場合は、このようにグラフ全頂点データ２０２の全体をメイン・メモリ１０４にロードできないことがある。そのような場合、例えば、次のような方法が考えられる。すなわち、許される限りのメイン・メモリ１０４の領域を確保しておき、そこに可能な限りのグラフ全頂点データ２０２の頂点のデータをロードしておく。そうして、前処理部２１２が、ある頂点vのデータを、メイン・メモリ１０４にロードされた頂点データ２０２'中で探し、もし見つかれば、そのデータを前処理部２１２は利用する。もし見つからなければ、例えば、見つからなかった頂点のデータを含むあるまとまりの頂点データを、ハードディスク１１４に記録されているグラフ全頂点データ２０２からメイン・メモリ１０４にロードし、頂点データ２０２'の領域の一部に置換することにより、前処理部２１２が参照可能となるようにすることができる。

【００２３】

このようなグラフ全頂点データ２０２をメイン・メモリ１０４にロードするためのメモリ管理ルーチンは、必要に応じて予め用意されるものであるが、オペレーティング・システムが提供するＡＰＩ関数やＣ言語の標準ライブラリ関数等を用いて慣用の技術で実現できるものであるため、ここでは、これ以上詳述しない。

【００２４】

後で詳しく説明するが、前処理部２１２は、好適には、メイン・メモリ１０４にロードされたグラフ全頂点データ２０２'から個々の頂点データを読み取って、全ランドマーク間の最短経路長のデータ２０４と、全頂点ｖの近くのｋ個のランドマークとの最短経路長のデータ２０６を生成し、後で読み出し可能に、ハードディスク１１４に記憶する機能を

もつ。なお、全ランドマーク間の最短経路長のデータ２０４と、全頂点ｖの近くのｋ個のランドマークとの最短経路長のデータ２０６は、好適には、上限値計算部２１４及び下限値計算部２１６が高速でアクセスできるように、データ２０４'及びデータ２０６'として、メイン・メモリ１０４にロードされる。

【００２５】

上限値計算部２１４は、メイン・メモリ１０４にロードされた、全ランドマーク間の最短経路長のデータ２０４'と、全頂点ｖの近くのｋ個のランドマークとの最短経路長のデータ２０６'を参照して、後述する手順により、頂点の上限値を計算して提供する機能をもつ。上限値計算部２１４は、他のプログラムまたはモジュールから呼び出され、好適には倍精度浮動小数点数として上限値の値を返す、サブルーチンまたは関数として実装される。

【００２６】

下限値計算部２１６は、メイン・メモリ１０４にロードされた、全ランドマーク間の最短経路長のデータ２０４'と、全頂点ｖの近くのｋ個のランドマークとの最短経路長のデータ２０６'を参照して、後述する手順により、頂点の下限値を計算して提供する機能をもつ。下限値計算部２１６は、他のプログラムまたはモジュールから呼び出され、好適には倍精度浮動小数点数として下限値の値を返す、サブルーチンまたは関数として実装される。

【００２７】

クエリー処理部２２０は、実際の出発頂点と、宛先頂点の指定を受けて、それらの頂点間の最短経路を探索し、その結果を返す機能をもつ。クエリー処理部２２０は、Ａ＊探索に基づき、必要に応じて上限値計算部２１４と、下限値計算部２１４を呼び出して上限値と下限値を利用するとともに、探索方向決定部２１８を利用して、探索方向を決定することにより、経路探索を行う機能をもつ。

【００２８】

以下、前記各処理モジュールの動作を、フローチャートを参照して、説明する。図５を参照すると、前処理部２１２の動作フローチャートが示されている。図５のステップ５０２では、グラフＧの頂点から、Ｌ個の頂点がランドマークとして選ばれる。Ｌ個のランドマークは、グラフＧにおいて、なるべく均一に分布するように配置すのが望ましい。本発明者の実験によれば、ランドマークの個数Ｌは、グラフＧの頂点の数をＮとすると、$L = N^{1/2}$個程度、あるいはそれ以上にすればよいことが分かった。

【００２９】

ランドマークを選択する１つの方法は、下記のようなものである。

・グラフに配置された頂点を全て含む最小の矩形を取る。
・矩形を縦横それぞれの辺に平行な、矩形の中心を通る２本の直線で田の字に４等分する。
・分割されたそれぞれの領域が規定数以上の頂点を含んでいれば、再度４等分する。
・以後これを繰り返す。
・以上の手続きが終了後、各小領域の中心部に最も近い頂点をランドマークとして選択する。

【００３０】

あるいは、より粗い方法ではあるが、ランドマークは、Ｃ言語でライブラリとして装備されている乱数関数を用いて、選んでもよい。例えば、random関数を用いて、異なるL個の頂点が選ばれるように、random(N)を少なくともL回繰り返して、その結果の整数の番号の頂点をランドマークとしてもよい。

【００３１】

こうしてステップ５０２で、Ｌ個のランドマークが選ばれると、次に、ステップ５０４では、選ばれた全Ｌ個のランドマーク間の最短経路長が計算される。この最短経路長の計算は、例えば、公知のダイクストラ法などの方法を用いて、行ってもよい。全Ｌ個のランドマーク間の最短経路長だと、実質的に、L^2回の計算となる。以下では、ランドマークL

vと、ランドマークLwにつき、こうして計算された最短経路長を、d(Lv,Lw)であらわすものとする。このようにして計算されたd(Lv,Lw)の値は、一旦、適当な配列変数に格納される。ステップ５０４の完了時点で、その配列変数に入れられ、メイン・メモリ１０４に展開されているデータの集まりが、全ランドマーク間の最短経路長のデータ２０４'となる。

【００３２】

ステップ５０６では、グラフの各頂点vについて、vの近くのk個のランドマークを選択し、頂点vと、そのk個の各ランドマークへの最短経路長が計算される。この、近傍のk個のランドマークの選択は、例えば、vから全てのランドマークへの最短経路長を計算し、そのうちの最短経路長が小さいものから順にk個選ぶという方法で行う。ここでも、最短経路長の計算は、ダイクストラ法などの周知の経路探索方法を用いることができる。本発明者の実験によれば、k=4程度が望ましいことが分かった。以下では、頂点vと、ランドマークLwにつき、こうして計算された最短経路長を、d(v,Lw)であらわすものとする。各頂点に対して、このようにしてk_v個のランドマークが対応づけられる。このようにして計算されたd(Lv,Lw)の値は、一旦、適当な配列変数に格納される。ステップ５０６の完了時点で、その配列変数に入れられ、メイン・メモリ１０４に展開されているデータの集まりが、全頂点vの近くのランドマークとの最短距離長のデータ２０６'となる。

尚、ここでグラフの各頂点vというとき、ランドマークとして選ばれた頂点自体も除外されないことに留意されたい。

【００３３】

図６は、ランドマークと、他の頂点との関係を模式的に示す図である。分かりやすくするために、実際のグラフよりも、ずっと頂点の数を少なく示してある。図６では、白丸がランドマークであり、ここでは、ランドマークの数L=4である。また、他の頂点を黒丸で示し、図示するように、1つの黒丸の頂点には、２個のランドマークが対応付けられているので、この図の例では、k=2である。

【００３４】

ステップ５０８では、ステップ５０４とステップ５０６で計算され、メイン・メモリ１０４上に、データ２０４'及びデータ２０６'として存在している計算結果が、例えば、C言語のread()、write()のようなライブラリ関数を用いて、ハードディスク１１４に、それぞれ、データ２０４及びデータ２０６として記録される。より具体的に説明すると、ステップ５０４での計算結果として、２つのランドマークLv,Lwの全ての組み合わせに対する最短経路長d(Lv,Lw)が、全ランドマーク間の最短経路長データ２０４として、ハードディスク１１４に記録され、このデータは、データ２０４'として、ハードディスク１１４からメイン・メモリ１０４にロードされたとき、上限値計算部２１４または下限値計算部２１６が、任意の２つのランドマークLv,Lwに対して、その最短経路長d(Lv,Lw)を取り出すことができるようになっている。

【００３５】

同様に、ステップ５０６での計算結果として、グラフの任意の頂点vに対応づけられたk個のランドマークLv_kに対する、最短経路長d(v,Lv_k)が、全頂点vの近くのk個のランドマークとの距離データ２０６として、ハードディスク１１４に記録され、このデータは、データ２０６'として、ハードディスク１１４からメイン・メモリ１０４にロードされたとき、任意の頂点vに対して、データ２０６から、上限値計算部２１４または下限値計算部２１６が、その頂点に近傍のランドマークとして関連付けられたk個のランドマークへの最短経路長d(v,Lv_k)の各々を取り出すことができるようになっている。

【００３６】

尚、図５のフローチャートでは、ステップ５０４とステップ５０６の終了後、ステップ５０８で、ステップ５０４とステップ５０６での計算結果を一気にハードディスク１１４に記録するように示されているが、ステップ５０４での計算の終了後直ちに、ステップ５０４での計算結果をハードディスク１１４に記録し、ステップ５０６での計算の終了後直ちに、ステップ５０６での計算結果をハードディスク１１４に記録するようにしてもよく

、そのようにしても、図５のフローチャートの処理と、本質的な差異はないことを当業者なら、理解するであろう。

【００３７】

さて、図７と図８を参照して、上限値計算部２１６と、下限値計算部２１８の計算処理について説明する前提として、本発明の特徴である四角不等式に基づく計算の原理について、図９を参照して説明する。図９には、頂点vと、それに近接するランドマークLvと、頂点wと、それに近接するランドマークLwの４つの頂点が描かれている。その４つの頂点を線で結ぶと、必ずしも長方形ではないにしても、何らかの四角形を形成することは確かである。もしこの四角形の頂点が全て同一平面上にあれば、任意の四角形において、一辺は、必ず残りの三辺の長さの和よりも小さい、という命題は無条件で成立する。しかし、四角形の頂点が同一平面上にないと、一辺は、残りの三辺の長さの和よりも小さいとは、必ずしも言えなくなる。ところが、辺の間の距離を測る指標を、最短経路長にすることによって、四角形の頂点が同一平面上になくても、一辺は、必ず残りの三辺の長さの和よりも小さい、という命題が復活する。そこで、まず、頂点vと頂点wを結ぶ線をその一辺と考えると、次の関係式が成立する。

d(Lv,v) + d(Lv,Lw) + d(w,Lw) >= d(v,w)

この式から、d(Lv,v) + d(Lv,Lw) + d(w,Lw)は、常にd(v,w)より大きいと言えるので、d(Lv,v) + d(Lv,Lw) + d(w,Lw)の上限を以って、d(v,w)の上限とすることができる。図７のフローチャートでは、この式を用いて、d(v,w)の上限値が計算される。

【００３８】

次に、ランドマークLvとランドマークLwを結ぶ線をその一辺と考えると、

d(Lv,v) + d(v,w) + d(w,Lw) >= d(Lv,Lw)

これから、d(v,w) >= d(Lv,Lw) - d(Lv,v) - d(w,Lw)

この式から、d(Lv,Lw) - d(Lv,v) - d(w,Lw)は常にd(v,w)より小さいと言えるので、d(Lv,Lw) - d(Lv,v) - d(w,Lw)の下限を以って、d(v,w)の下限とすることができる。図８のフローチャートでは、この式を用いて、d(v,w)の下限値が計算される。

【００３９】

ここで、図７のフローチャートを参照して、上限値計算部２１４の処理について、説明する。

図７のステップ７０２では、MaxValという変数が、初期的に0にセットされる。この変数は、例えば、Ｃ言語だと、倍精度浮動小数点のdoubleとして宣言されるものである。

【００４０】

ステップ７０４では、頂点vにつき、その近くとして関連付けられたランドマークLvのうちのk_v個が選ばれる。このような対応付けは、図２に示す、全頂点vの近くのk個のランドマークとの距離データ２０６'として、既に前処理部２１２の処理によって予め用意されており、頂点vに対応するものとして、データ２０６'から取り出される。

【００４１】

ステップ７０６では、同様に、頂点wにつき、その近くとして関連付けられたランドマークLwのうちのk_w個が選ばれる。１つの実施例では、$k_v = k_w = 4$である。

【００４２】

さて、ステップ７０８では、頂点vからランドマークLvへの最短経路長が、データ２０６'から取り出されて、その値が、倍精度浮動小数点の変数D1に格納される。ステップ７１０では、LvからLwへの最短経路長が、データ２０４'から取り出されて、その値が、倍精度浮動小数点の変数D2に格納される。ステップ７１２では、ランドマークLwから頂点wへの最短経路長が、データ２０６'から取り出されて、その値が、倍精度浮動小数点の変数D3に格納される。

【００４３】

ステップ７１４では、D1 + D2 + D3が計算されて、それが、MaxValという値と比較され、その大きい方が、MaxValにセットされる。

【００４４】

このようにして、データ２０６'から、頂点vに対応して取り出されたk_v個のランドマークLvと、wに対応して取り出されたk_w個のランドマークLwにつき、ステップ７０８～７１４を巡らせ、その結果MaxValに与えられた値が、d(v,w)の上限値ということになる。すなわち、ステップ７０６から７１６までが、Lwについての内側のループで、ステップ７０４からステップ７１８の外側のループである。$k_v = k_w = 4$だと、4 * 4 = 16で、ステップ７０８～７１４は、１６回実行される。

【００４５】

但し、頂点v,wに対して、k^2回の計算を行うと計算量が大きくなりすぎる場合があるので、頂点vの近くのk'_v個のランドマークと、頂点wの近くのk'_w個のランドマークだけ抜き出して、計算のループの回数を$k'_v * k'_w$に削減する、という工夫を行うことができる。

このとき、$k'_v <= k_v$, $k'_w <= k_w$である。

本発明者のさまざまな試行により、$k'_v = k$, $k'_w = 1$の設定が良好な結果をもたらすことが分かった。

【００４６】

このようにして計算された値は、頂点v,wに対応するd(v,w)の上限値として、上限値計算部２１４を呼び出すプログラムに返される。上限値計算部２１４を呼び出すプログラムとは、典型的には、クエリー処理部２２０である。

【００４７】

次に、図８のフローチャートを参照して、下限値計算部２１６の処理について、説明する。

図８のステップ８０２では、MinValという変数が、初期的に∞にセットされる。この変数は、例えば、Ｃ言語だと、倍精度浮動小数点のdoubleとして宣言されるものである。コンピュータの変数の値で、∞ということは現実的にはあり得ないが、ここでは、グラフの経路長としてとり得る値よりも十分大きい値を予めセットしておくものとする。

【００４８】

ステップ８０４では、頂点vにつき、その近くとして関連付けられたランドマークLvのうちのk_v個が選ばれる。このような対応付けは、図２に示す、全頂点vの近くのk個のランドマークとの距離データ２０６'として、既に前処理部２１２の処理によって予め用意されており、頂点vに対応するものとして、データ２０６'から取り出される。

【００４９】

ステップ８０６では、同様に、頂点wにつき、その近くとして関連付けられたランドマークLwのうちのk_w個が選ばれる。好適な実施例では、$k_v = k_w = 4$である。

【００５０】

ステップ８０８では、ランドマークLvからランドマークLwへの最短経路長が、データ２０４'から取り出されて、その値が、倍精度浮動小数点の変数D1に格納される。ステップ８１０では、Lvからvへの最短経路長が、データ２０６'から取り出されて、その値が、倍精度浮動小数点の変数D2に格納される。ステップ８１２では、頂点wからランドマークLwへの最短経路長が、データ２０６'から取り出されて、その値が、倍精度浮動小数点の変数D3に格納される。

【００５１】

ステップ８１４では、D1 - D2 - D3が計算されて、それが、MinValという値と比較され、その小さい方が、MinValにセットされる。

【００５２】

このようにして、データ２０６'から、頂点vに対応して取り出されたk_v個のランドマークLvと、wに対応して取り出されたk_w個のランドマークLwにつき、ステップ８０８～８１４を巡らせ、その結果MinValに与えられた値が、d(v,w)の下限値ということになる。すなわち、ステップ８０６から８１６までが、Lwについての内側のループで、ステップ８０４からステップ８１８の外側のループである。$k_v = k_w = 4$だと、4 * 4 = 16で、ステップ８０８～８１４は、１６回実行される。

【００５３】

上限値の計算の場合と同様に、頂点v,wに対して、k^2回の計算を行うと計算量が大きくなりすぎることがあるので、頂点vの近くのk'_v個のランドマークと、頂点wの近くのk'_w個のランドマークだけ抜き出して、計算のループの回数を$k'_v * k'_w$に削減する、という工夫を行うことができる。このとき、$k'_v <= k_v$, $k'_w <= k_w$である。$k'_v = k$, $k'_w = 1$の設定が良好な結果をもたらすことも同様である。

【００５４】

このようにして計算された値が、頂点v,wに対応するd(v,w)の下限値として、下限値計算部２１６を呼び出すプログラムに返される。下限値計算部２１６を呼び出すプログラムとは、典型的には、探索方法決定部２１８及びクエリー処理部２２０である。

【００５５】

尚、上限値、下限値の計算ループで、$k'_v = k$, $k'_w = 1$のような工夫を行うことができることを上記で説明したが、このような設定を行うならば、上限値、下限値の計算を行う前に予め、固定的に設定し、後は変更しないようにすることに留意されたい。例えば、頂点p,q,r について、d(p,q) の上下限を計算するときは、pの近くの k'_v 個のランドマークと q の近くのk'_w個のランドマークを用いて計算し、d(q,r) の上下限を計算するときは、q の近くの k'_v 個のランドマークと r の近くのk'_w個のランドマークを用いて計算する。

【００５６】

また、図７と図８の計算ルーチンを、グラフの頂点v,wのすべてに亘って予め計算して、v,wの上限値と下限値をハードディスク１１４に記録しておいて、実行時にメイン・メモリ１１４にロードし、Ａ＊探索においては、図７と図８の計算ルーチンを呼び出す代わりに、v,wの値で、メイン・メモリ１１４にある値をルックアップすることにより、データの計算が高速化されると思われるかもしれない。実際、多くの場合、ルックアップは、その場で計算するよりは短い時間で済む。しかし、このような方法は、ごく少ない頂点のグラフにしか適用できない。すなわち、一般的に、あまり精細でない地図でも、グラフの頂点としてあらわされる地点の数は、少なくともN = 3 × 10^5程度である。このようなデータをv,wの組について用意するには、組み合わせの数は少なくとも、$N^2 = 9 \times 10^{10}$で、それに上記kを掛け、また、倍精度浮動小数点数が8バイトであらわされることを考慮すると、あまり精細でない地図でも、上限値と下限値を記録するには、数テラバイトを下らない記憶容量を要することが理解されよう。

【００５７】

次に、図１０のフローチャートを参照して、探索方向決定部２１８の処理を説明する。探索方向決定部２１８に与えられるのは、グラフの２つの頂点、sとtである。探索方向決定部２１８は、クエリー処理部２２０から呼び出されるものであり、典型的には、sとtは、探索すべき始点と終点である。

【００５８】

さて、図１０のステップ１００２では、頂点sから頂点tへの最短経路長の下限値が、下限値計算部２１６を呼び出すことによって、求められる。同様に、ステップ１００４では、頂点tから頂点sへの最短経路長の下限値が、下限値計算部２１６を呼び出すことによって、求められる。

【００５９】

ステップ１００６では、このようにして計算された、頂点sから頂点tへの最短経路長の下限値と、頂点tから頂点sへの最短経路長の下限値とが比較され、頂点sから頂点tへの最短経路長の下限値が、頂点tから頂点sへの最短経路長の下限値より大きいと、ステップ１００８で、sとtが入れ替えられる。このように、最短経路長の下限値の大きさに応じて、sとtを入れ替える理由は、頂点sから頂点tへの最短経路長の下限値が、頂点tから頂点sへの最短経路長の下限値より大きいと、sからtへ向かう経路の上下限値の精度よりも、tからsへ向かう経路の上下限値の精度の方がよいことが多い、という発明者の経験則によるものである。すなわち、より高い精度の上下限値を用いることで、Ａ＊探索の探索速度の向上がはかられる。

【００６０】
図１１は、ここまでで用意された結果を以って、最終的に経路探索するためのクエリー処理部２２０の処理を示すフローチャートである。
【００６１】
クエリー処理部２２０に与えられる入力は、始点としてのグラフの頂点sと、終点としてのグラフの頂点tであり、クエリー処理部２２０は、これらの頂点の間の最短経路を探索して出力として与えるものである。
【００６２】
クエリー処理部２２０を実行するに際して、グラフ全頂点データ２０２、全ランドマーク間の最短経路長のデータ２０４、及び全頂点vの近くのランドマークとの最短距離長のデータ２０６は、予め、データ２０２'と、データ２０４'と、データ２０６'として、ハードディスク１１４からメイン・メモリ１０４にロードされているものとする。
【００６３】
ステップ１１０２では、頂点s及びtのデータを以って、図１０に関連して説明した、探索方向決定部１０が呼ばれる。この結果、頂点s及びtの条件に応じて、sとtの入れ替わりが行われ、その頂点s及びtが、ステップ１１０４のＡ＊探索ルーチンに引き渡される。
【００６４】
次に、ステップ１１０４のＡ＊探索ルーチンを、Ｃ言語的な擬似コードを用いて説明いる。
下記の擬似コードで、頂点 v, w について d(v,w) は、vからwへの最短経路長を表すものとする。また、w(v,w)を、vとwを結ぶ枝の長さとする。
先ず、始点sから終点tまでの最短経路を求めるとして、各頂点vについて以下のように定める。
ここで、π(v)をd(v,t) の下限値とする。その下限値は、図１１のフローチャートにおいて、ステップ１１０８で、v,tに関して下限値計算部２１６を呼び出すことにより、求められる。
更に、初期設定として、始点sから終点tまでの最短経路を求めたい場合、各頂点vについて以下のように定める。

```
k(v): v=s のとき k(s):=π(s)、v≠sのときk(v):=∞;
d(v): v=s のとき d(v):=0、v≠sのときd(v):=∞;
prev(v) = NULL;
頂点集合 S = {s};
```

そうして、次のようにして探索処理を行う。

```
while( Sが空でない ) {
    S から k(v)の値が最も小さい頂点vを除去;
    for(vから出ている全ての枝(v,w) について以下の操作を行う) {
        if( k(w) > k(v) + w(v,w) - π(v) + π(w)) {
            k(w) := k(v) + w(v,w) -π(v)+π(w);
            d(w) := d(v) + w(v,w);
            prev(w) := v;
            if(d(w)+π(w) <= min(d(s,t) の上限値, d(t)))
              { S ← S ∪{w}; }
        }
    }
}
```

最後に、d(t) に入っている値が最短経路長になる。また、この擬似コードで示す処理において、d(s,t) の上限値を求める際に、図１１のステップ１１０６に示すように、s,tに関して上限値計算部２１４が呼び出される。
このとき、prev(t) に入っている頂点を t'、prev(t')に入っている頂点を t'' ・・・

とすると、 t←t'←t''←・・・←s が最短経路になる。

【００６５】

本発明者は、本発明の効果を確認するために、次のような実験を行った。先ず、下記の表に示すような、７つの重み付きグラフを用意した。また、パラメータは、k=4，k_w=4，k_v=1を用いた。また、ランドマークは、既に説明したが、下記のような方法で、配置した。

・グラフに配置された頂点を全て含む最小の矩形を取る。
・矩形を縦横それぞれの辺に平行な、矩形の中心を通る２本の直線で田の字に４等分する。
・分割されたそれぞれの領域が規定数以上の頂点を含んでいれば、再度４等分する。
・以後これを繰り返す。
・以上の手続きが終了後、各小領域の中心部に最も近い頂点をランドマークとして選択する。

【表１】

	頂点数	枝数
グラフ１	340 万	1000 万
グラフ２	33 万	86 万
グラフ３	58 万	135 万
グラフ４	40 万	100 万
グラフ５	161 万	398 万
グラフ６	195 万	545 万
グラフ７	207 万	516 万

【００６６】

グラフ１に対して、文献[1]の方法と、本実施例の方法を、下記の表のような条件で、それぞれ適用した。特に、文献[1]の方法と本実施例の方法にあっては、クラスタは用いないので、「ランドマークまたはクラスタ数」の欄は、ランドマーク数と解釈されたい。すると、探索時間と探索頂点数に関して、本実施例の方法は、明らかに優れた効果を奏したことが見て取れる。尚、この表及び以下の表で、「探索時間」の欄は、ダイクストラ法の性能を１とした場合の性能の程度を表している。すなわち、実際にかかった時間は、この「探索時間」に示した数値の逆数に比例する。同様に、「探索頂点数」の欄も、ダイクストラ法で探索される頂点数の１とした場合の、探索される頂点数の少なさの度合いを示すものである。よって、実際に探索された頂点数という意味では、ここの「探索頂点数」に示した数値の逆数に比例する。さらに、前処理後の必要記憶量という意味でも、この実施例が、文献[1]の方法より優れていることが示されている。

【表２】

グラフ１

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[1]の手法	16	256MB	13 倍	43 倍
実施例の手法	7000	177MB	69 倍	37 倍

【００６７】

グラフ２及び３に対しても、下記のように、本実施例の方法が、文献[1]よりも、探索性能において優れていることが示されている。

【表３】
グラフ２

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[1]の手法	16	21M	9.1倍	31倍
実施例の手法	2000	19M	22倍	28倍

【表４】
グラフ３

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[1]の手法	16	38MB	13倍	45倍
実施例の手法	4000	46MB	23倍	26倍

【００６８】
次に、以下では、文献[2]の方法と、本実施例の方法を比較する実験を行った。なお、本発明の方法は、クラスタは用いないので、「ランドマークまたはクラスタ数」の欄は、ランドマークの数と解釈する。一方、文献[2]の方法では、ランドマークは用いず、その代わりにクラスタを用いるので、「ランドマークまたはクラスタ数」の欄は、クラスタの数と解釈する。
下記のようにグラフ４に、文献[2]の方法と、本実施例の方法を適用した結果は、やはり、本実施例の方法が実質的に優れた効果を奏することを示している。この実施例は、前処理後の必要記憶量という意味でも、文献[2]の方法より優れていることが示されている。

【表５】
グラフ４

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[2]の手法	4000	94 MB	4.55倍	9.68倍
実施例の手法	4000	39 MB	30.4倍	41.3倍

【００６９】
グラフ５～７に、文献[2]の方法と、本実施例の方法を適用した比較結果は、下記のとおりである。これらの実験例でも、本実施例の方法が、文献[2]の方法よりも顕著な効果を奏することが見て取れる。

【表６】
グラフ５

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[2]の手法	7000	282 MB	5.6倍	10.9倍
実施例の手法	7000	136 MB	30.1倍	37.9倍

【表７】
グラフ６

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[2]の手法	4000	95 MB	4.96 倍	6.43 倍
実施例の手法	4000	78 MB	19.6 倍	25.7 倍

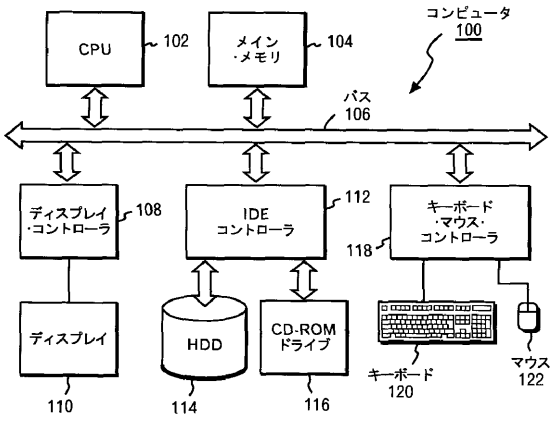
【表８】
グラフ７

	ランドマークまたはクラスタ数	前処理後必要記憶量	探索時間	探索頂点数
文献[2]の手法	7000	284 MB	8.11 倍	14.1 倍
実施例の手法	7000	148 MB	60.5 倍	49.3 倍

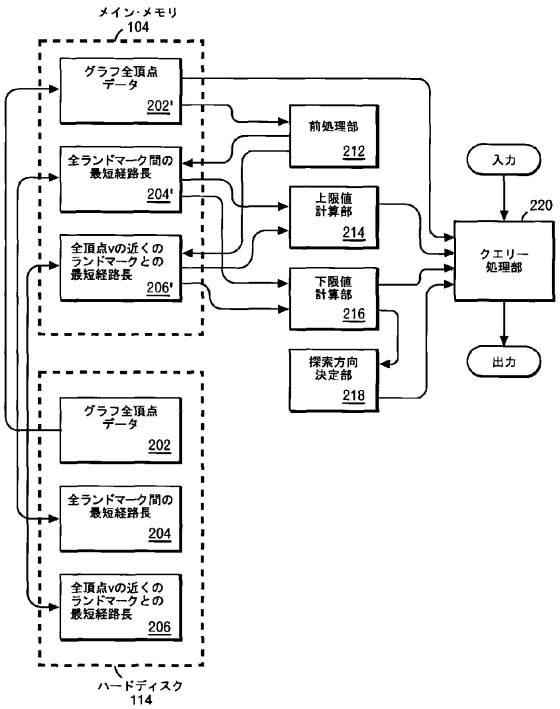
【００７０】
尚、上記の実施例では、無向グラフとして説明してきたが、本発明は、有向グラフにも適用できる。その場合、探索しようとする経路が、有向グラフの枝の向きと逆であると、その枝は避けて、別の枝を探索する、というような処理を行う。
【００７１】
また、上記実施例では、スタンドアロンの環境で、グラフのデータを探索する処理を説明したが、Ｗｅｂサーバ上で、上記で説明した処理を実行して、探索可能な環境を用意し、図２に示すクエリー処理部２２０の入力のところで、クライアント・コンピュータからの探索問い合わせを受け付け、クエリー処理部２２０の出力のところで、探索問い合わせを受け付けたクライアント・コンピュータに、探索結果を送信するようにすることもできる。
【００７２】
また、最近の携帯電話の性能の向上により、１ＧＢ以上のディスク容量と、高速のＣＰＵをもつ携帯電話もあらわれてきている。このような携帯電話には、図２に示すグラフ全頂点データ２０２、全ランドマーク間の最短経路長のデータ２０４、全頂点vのその近傍のランドマークとの最短経路長のデータ２０６、上限値計算部２１４、下限値計算部２１６、探索方向決定部２１８及びクエリー処理部２２０の全体をそのディスクに格納して、携帯電話だけでスタンドアロンに探索を行い、その結果を携帯電話の画面に表示するようにすることができる。
【００７３】
さらに、本発明に係るグラフの探索技法は、カーナビ、地図探索、鉄道路線探索など、頂点が重み付きグラフとして表現される対象における、任意の対象物上での経路探索に適用し得るものであることを理解されたい。
【図面の簡単な説明】
【００７４】
【図１】本発明を実施するためのハードウェアのブロック図である。
【図２】本発明を実施するための、データと処理モジュールのブロック図である。
【図３】保存されているグラフ頂点データの論理構造を示す図である。
【図４】重み付きグラフの一部を示す図である。
【図５】グラフの頂点データの前処理部の処理のフローチャートを示す図である。
【図６】ランドマークと他の点の関連付けを示す図である。
【図７】上限値計算部の処理のフローチャートを示す図である。
【図８】下限値計算部の処理のフローチャートを示す図である。
【図９】四角不等式におけるランドマークとその他の点の位置関係を示す図である。

【図１０】探索方向決定部の処理のフローチャートを示す図である。
【図１１】クエリー処理部の処理のフローチャートを示す図である。

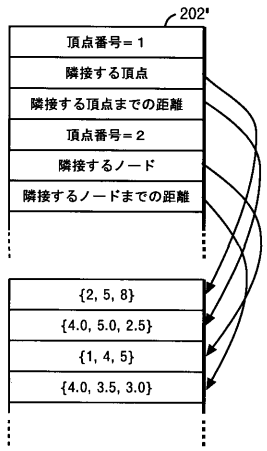
【図１】



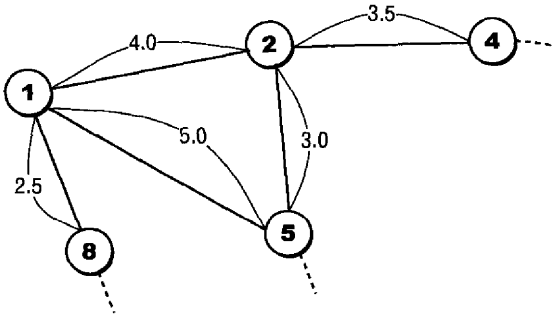
【図２】



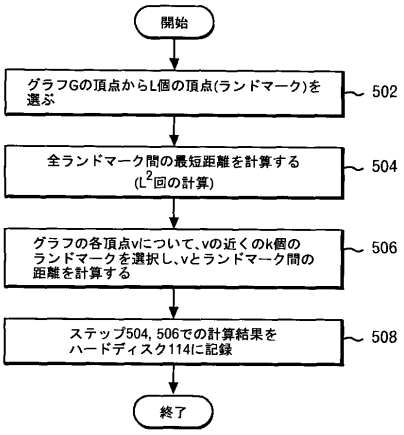
【図３】



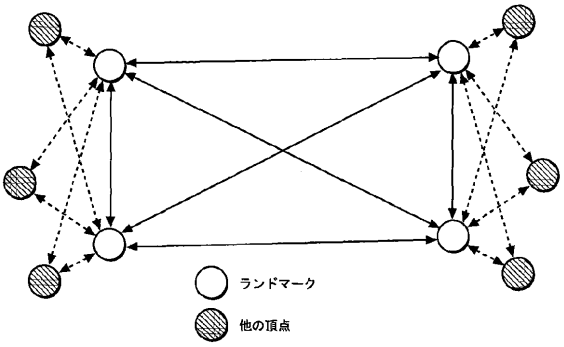
【図４】



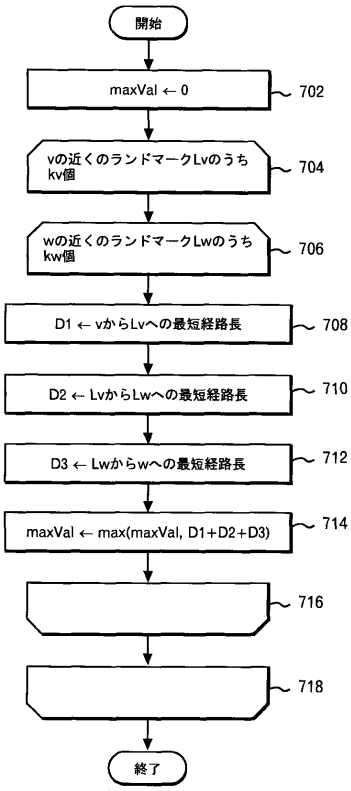
【図５】



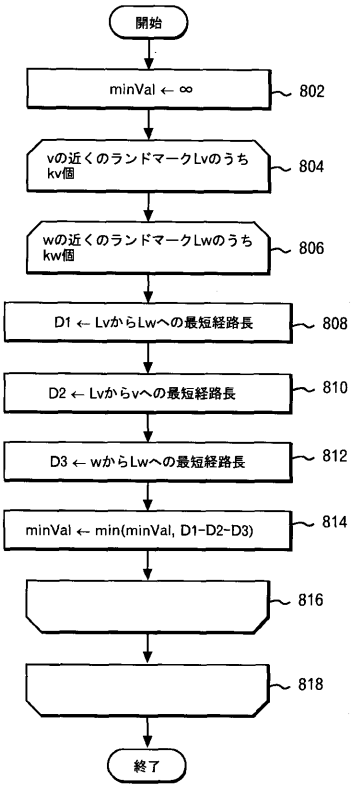
【図６】



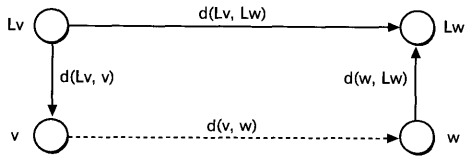
【図７】



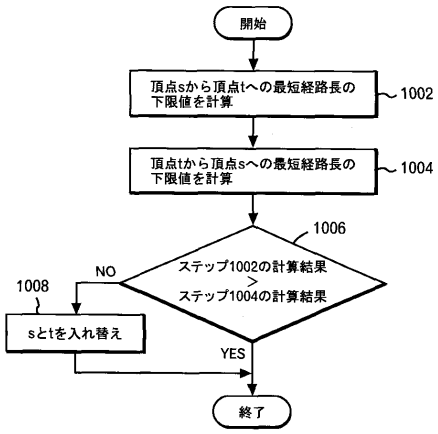
【図８】



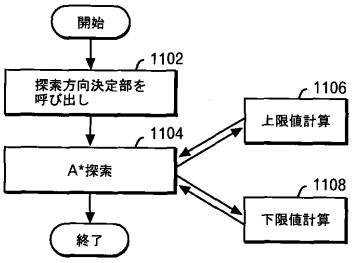
【図９】


Lv
d(Lv, Lw)
Lw
d(Lv, v)
d(w, Lw)
v
d(v, w)
w

【図１０】


開始
頂点sから頂点tへの最短経路長の下限値を計算
1002
頂点tから頂点sへの最短経路長の下限値を計算
1004
1006
ステップ1002の計算結果 > ステップ1004の計算結果
NO
1008
sとtを入れ替え
YES
終了

【図１１】


開始
1102
探索方向決定部を呼び出し
1106
上限値計算
1104
A*探索
1108
下限値計算
終了

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 ▲柳▼澤 弘揮
神奈川県大和市下鶴間１６２３番地１４ 日本アイ・ビー・エム株式会社 東京基礎研究所内
(72)発明者 今村 友和
神奈川県大和市下鶴間１６２３番地１４ 日本アイ・ビー・エム株式会社 東京基礎研究所内

Ｆターム(参考) 2C032 HB25 HC11 HD16
2F129 DD02 DD03 DD18 DD62 HH04 HH12 HH18 HH20 HH25
5H180 AA01 EE02 FF01 FF10 FF14