

AVENUE シミュレーションモデル解説書

by i-Transport Lab.

1. シミュレーションモデルの概要

1.1 サブモデルの構成

シミュレーションモデルは次の 3 つのサブモデルから構成される。

- 1) 交通流再現サブモデル
- 2) 経路選択サブモデル
- 3) 交通制御サブモデル

モデルは最小時間単位を 1 秒とする時間スキャン(periodic-scanning)方式で実装されている。スキャン時間の長さは任意だが、通常は 1秒にとられる。

交通流再現サブモデルには、後述するハイブリッドブロック密度法が採用され、スキャンインタバルごとに交通状況を再現する。

経路選択サブモデルはシミュレーション開始時に必要な経路を生成し、ユーザが指定した時間ごとに経路コストを更新する機能と、ドライバーが適切なタイミングで提示された経路コストに従って、各自の判断基準で次にとるべき経路を選択する機能を備える。経路コストは、距離、旅行時間、右左折回数などの関数として表現され、複数の利用者層の特性に応じた設定が可能である。

交通制御サブモデルは決められたタイミングで交通流を制御するもので、信号制御や通行規制の時間指定などの機能が含まれる。

1.2 モデルへの入力項目

シミュレーションモデルへの基本的な入力項目は、次の 3種類である。

- 1) ネットワークデータ
- 2) 交通制御データ
- 3) 時間帯別 OD 交通量データ

ネットワークデータには、ノードとリンクの接続関係だけではなく、飽和交通流率や自由旅行速度などの交通流に関する情報、車線数、右左折専用付加車線の長さ、進行方向規制といった車線構成に関する情報も含まれる。

信号データにはサイクルとスプリットだけでなくオフセット情報も含まれ、系統制御やエリア制御を表現できるようになっている。また信号現示も右折専用青矢印など、ユーザが任意の信号現示を指定できるようになってい

る。

時間帯別 OD 交通量データにはユーザが指定した一定時間ごとの、目的地別の発生交通量を与える。ここでは OD ペアはパーソナル調査でのようなゾーン間の OD ペアではなく、ネットワークの端点のペアを意味する。OD 交通量は車両タイプ別、旅行目的別など、利用者層ごとに設定することができる。

1.3 モデルからの出力項目

シミュレーションモデルは、再現されたネットワーク上の交通状況を観測し、必要なデータを集計・加工して、一定時間ごとに出力するものである。基本的な出力項目は、

- 1) リンクの方向別流出交通量
- 2) リンクの方向別平均旅行時間
- 3) リンク上の一定区間ごとの地点平均速度
- 4) プローブ車両の走行軌跡
- 5) 感知器情報

などである。リンクでの平均遅れや平均停止回数などのデータは、これらの基本出力データから2次的に求められる。

2. ハイブリッドブロック密度法

ハイブリッドブロック密度法は、交通流を流体近似したブロック密度法と、個別の車両属性を扱うための離散的な待ち行列モデルを組み合わせた手法である。ここではまずブロック密度法について説明し、ついでハイブリッドブロック密度法への拡張、またこの拡張により可能になった交通規制や信号制御の記述方法について述べる。さらに計算量を軽減するための改良を施したマルチスキャンハイブリッドブロック密度法についても説明する。

2.1 ブロック密度法

ブロック密度法はスキャンインタバルごとにリンク上の交通密度を時間更新する手法である。密度の更新は流量保存則と交通量 - 密度 (Q-K) 曲線に基づいているため、疎密波の伝播を含めた過飽和状態の交通流を再現することができる。

この手法ではリンクに交通容量 Q^c 、ジャム密度 K^j 、および自由流速度 V^f で決定される図 1 のような三角関数形の Q-K 曲線が設定され、臨界密度 K^c が求められている。リンクは一定の単位距離 dL で区切られたブロックという単位に分割される。単位距離 dL はスキャンインタバル dt の間に自由流速度 V^f で進む距離に等しい。

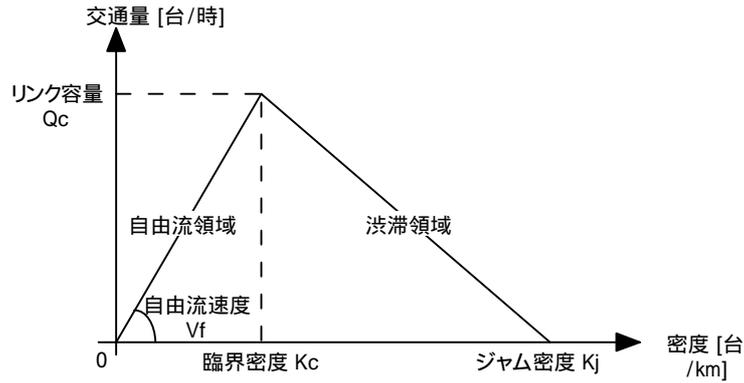


図 1 : ブロック密度法が仮定する交通量 - 密度関数

ひとつのブロックは車線をまたがっても、車線ごとに分割されていてもよく、上流と下流で隣接するブロックの間で交通量をやりとりする。とくに車線ごとに分割された場合には、あるブロックは斜め上流 / 斜め下流に位置するものも含め、すべての隣接するブロックとの間で交通量の移動の計算をおこなう。

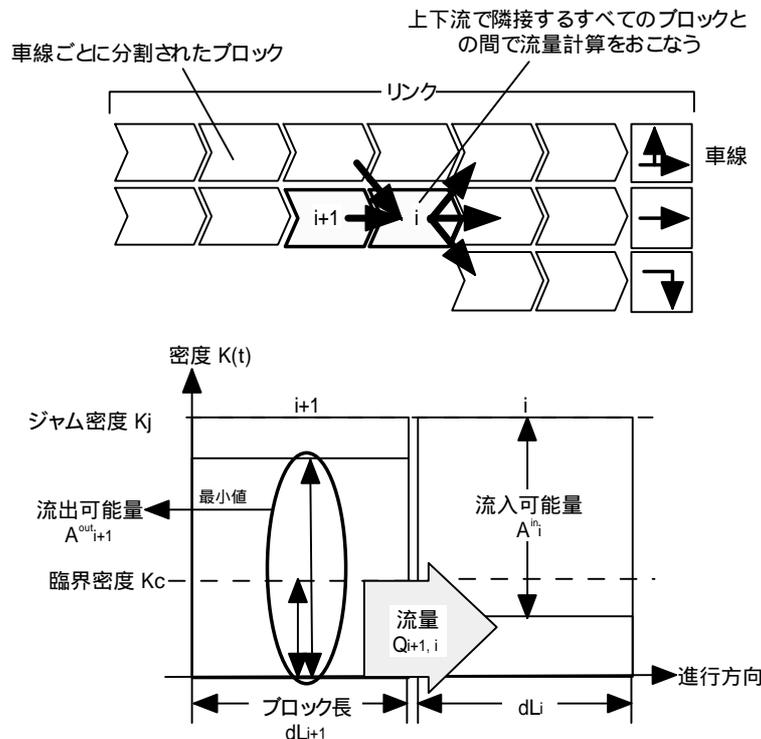


図 2 : ブロック密度法の概念図

以下では一つのブロックに着目し、それに隣接するブロックとの間を移動する交通量の計算について説明する。

ブロックにはリンク下流端から $i = 1, 2, \dots$ と添字がつけられる。またリンクの臨界密度とジャム密度に対応したブロック臨界密度 K_i^c とブロックジャム密度 K_i^j が与えられる。このとき、ある時刻 t におけるブロック密度の時間更新は(式 1) ~ (式 4)で示される。

$$A_i^{out}(t) = \min(K_i^c, K_i(t)) \frac{dL}{dt} \tag{式 1}$$

$$A_i^{in}(t) = \begin{cases} (K_i^j - K_i(t)) \frac{dL}{dt} & \dots (K_i(t) \leq K_i^c) \\ \frac{K_i^c (K_i^j - K_i(t))}{K_i^j - K_i^c} \frac{dL}{dt} & \dots (K_i^c < K_i(t) \leq K_i^j) \end{cases} \quad (\text{式 2})$$

$$Q_{i+1,i}(t) = \min(A_{i+1}^{out}(t), A_i^{in}(t)) \quad (\text{式 3})$$

$$K_i(t+1) = K_i(t)dL + Q_{i+1,i}(t)dt - Q_{i,i-1}(t)dt \quad (\text{式 4})$$

ここで、 $A_i^{out}(t)$ と $A_i^{in}(t)$ はそれぞれ時刻 t における単位時間当たりの流出可能量と流入可能量を、 K_i^c と K_i^j は臨界密度とジャム密度を、 $K_i(t)$ は時刻 t におけるブロック内車両密度を、また $Q_{i+1,i}(t)$ は時刻 t におけるブロック $i+1$ から i への単位時間当たりの流量を表している。

(式 1) ~ (式 4)は実際には次のようなアルゴリズムとして記述される。まず時刻 t においてブロックの流出可能量 $A_i^{out}(t)$ と流入可能量 $A_i^{in}(t)$ が求められる。流出可能量は(式 1)において、現在のブロック内の車両台数と最大流量の最小値として表されている。一方、(式 2)で表される流入可能量は、ブロックの状態が自由流領域にあれば、ブロック内の空き容量に等しく、渋滞領域であれば Q-K 曲線の渋滞側の直線から定まる流量となる。

全てのブロックについて流出可能量と流入可能量が計算された後、(式 3)によって隣接するブロック間の流量が計算される。流量は上流ブロックの流出可能量と下流ブロックの流入可能量の最小値として表される。この流量を用いて次の時刻でのブロック内車両密度を(式 4)によって更新する。(式 4)は交通量の保存則を表している。

上述のアルゴリズムはリンク上の密度管理に関するものだが、これをネットワーク上の密度管理に自然に拡張することができる。なぜならば、(式 1) ~ (式 4)にはブロックを下流側から順にスキャンするといったような制約がなく、1 スキャン前の状態を用いて、一旦移動させる交通量を計算してからブロックの密度を更新するので、ループを含むネットワークに適用した場合でも、どこを最下流のリンクとするかという問題が生じないからである。

2.2 ハイブリッドブロック密度法

上述のブロック密度法では、ブロック間を移動する流量が連続値として算出される。しかしながら、交通を流体近似したままでは経路選択や交通規制のモデル化に不利であるため、同時に離散車両を移動させるように拡張したハイブリッドブロック密度法に拡張する。

拡張の基本的な考え方は、ブロック密度法で求められた流量を切り上げた整数に等しい台数の車両を、上流ブロックから下流ブロックに移動させるものであるが、その際に生じる誤差を補正交通量として保存しておき、次のスキャン以降で優先的に移動させるというもので、数スキャンを通して連続量と離散量の整合性を確保するものである。

離散車両の移動量は次式において求められる。まず、ブロック間の車両移動量 $M_{i+1,i}(t)$ を、ブロック間流量を切り上げて求める。

$$M_{i+1,i}(t) = \text{RoundUp}(Q_{i+1,i}(t)dt) \quad (\text{式 5})$$

ただし、 $\text{RoundUp}()$ は切り上げ整数化の関数

同時に各ブロックでの交通密度の整合性を保つために、切り上げによって増加した小数点以下の補正交通量 $E_{i+1,i}(t)$ を次式 (6)で求める。

$$E_{i+1,i}(t) = M_{i+1,i}(t) - Q_{i+1,i}(t)dt$$

(式 6)

ただし,前スキャンに生じた補正交通量は離散車両の移動に優先して処理されるため,離散車両の移動台数を求める(式 5)(式 6)は 1スキャン前の補正交通量を考慮した形で記述されなければならない。したがってそれぞれ(式 5)(式 6)におきかえられる。

$$M_{i+1,i}(t) = \text{RoundUp}(\max(0, Q_{i+1,i}(t)dt - E_{i+1,i}(t-1))) \quad (\text{式 5})$$

$$E_{i+1,i}(t) = M_{i+1,i}(t) + E_{i+1,i}(t-1) - Q_{i+1,i}(t)dt \quad (\text{式 6})$$

また,車両に乗用車換算係数の情報を持たせ,流量計算と密度の改訂に利用することで,大型車などの影響を考慮することもできるようになる。

図 3 にハイブリッドブロック密度法による車両移動計算の例を示す。図の左半分には連続量の密度の変化と流量の移動が,右半分にはブロック内に存在する離散車両が移動する様子を示している。初期状態 $t = t_0$ では上流端のブロック $i+1$ に 2 台の車両が存在し,次スキャン以降でリンク容量で下流方向に流れていく。リンク容量から求められるブロック間の最大流量は,0.4 [台/秒] としてある。

最初のスキャンでは $i+1$ と i の間の移動台数は

$$M_{i+1,i}(t_0) = \text{RoundUp}(\max(0, 0.4)) = 1 \text{ [台]}$$

となり,先頭の車両が下流ブロック i に移動するが,その際に

$$E_{i+1,i}(t_0) = 1 - 0.4 = 0.6 \text{ [台]}$$

の補正交通量が生じる。次のスキャン t_{0+1} ではこの補正交通量を優先して移動させなければならないので, $i+1$ と i の間の移動台数は

$$\text{RoundUp}(\max(0, 0.4 - 0.6)) = 0 \text{ [台]}$$

となるため 2 台目の車両は移動できず,もとのブロックに残る。このときの補正交通量は

$$E_{i+1,i}(t_{0+1}) = 0 + 0.6 - 0.4 = 0.2 \text{ [台]}$$

となる。以下,同様の計算をおこなって車両を移動させることで,数スキャンにわたって連続量と離散量の整合性が保たれるのがわかる。

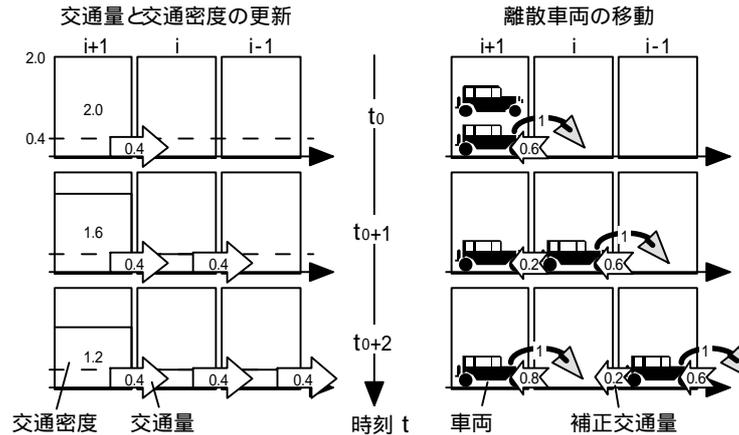


図 3 :ハイブリッドブロック密度法の概念図

2.3 車線選択 交通規制 信号交差点などのモデル化

ハイブリッドブロック密度法の導入によって、車両の属性情報を利用してさまざまな交通制御、交通規制をモデルに組み込むことが可能になる。また、車線ごとにブロックに分割し、直上流 / 直下流の方向だけでなく斜め上流 / 斜め下流のブロック間の移動も考慮することができる。ここではドライバーの車線選択や車線変更などの挙動、交差点での信号による制御や右折車両のギャップ待ちなどを例として説明する。

2.3.1 車線選択

ブロックには車両流入時の規制に関する情報を属性として記述することができる。あるブロックがその上流ブロックから車両を移動させてくる際に、その車両の属性と流入時の規制情報とを参照し、規制対象となる車両については流入させないことで、さまざまな車線に対する規制を表現することができる。

は車線の進行方向規制を実現している例である。各ブロックには、それが属する車線の進行方向規制に基づいて、同じような進行方向規制の情報が記述されている。一方、後述の経路選択モデルにより、リンク下流の交差点での各車両の進行方向がすでに決定されているので、各ブロックはその進行方向規制と車両の進行方向が適合するもののみ移動させることで、各車両は適切な車線上を走行することになる。

同様の規制は車種や特定の利用者層に対して設定することができる。これにより、バス車線や、駐車場入庫待ち車両専用のバッファ車線などが実現できる。

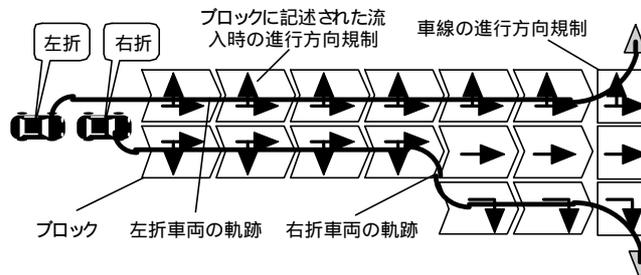


図 4 :ハイブリッドブロック密度法での車線選択

2.3.2 車線変更

街路では右折専用ポケットから溢れた車両が後続する直進車両の進行を妨げている状況がよく見られる。ハイ

ブリッドブロック密度法ではこのような現象もモデル化することができる。

図5は右折車両のブロッキングにより、直進車が車線変更するケースを示している。ハイブリッドブロック密度法では、まず直下流方向への車両移動が斜め方向の移動に優先して計算される。このとき第2車線に進入した直進車両は右折車両で一杯になっているブロックの手前で、これ以上直下流方向に進行できなくなるので、斜め下流のブロックに流出しようとする。斜め下流のブロックで直上流ブロックからの流入がなければ、その直進車両を受け入れる余裕がまだあるので、結果として直進車両は第1車線に移動する。第1車線の通行量が多ければ、このような斜め下流方向への移動ができる機会が少なくなり、第2車線での待ち行列が長くなる。

同様の仕組みで、路上駐車や道路工事による容量低下をモデル化することも可能である。

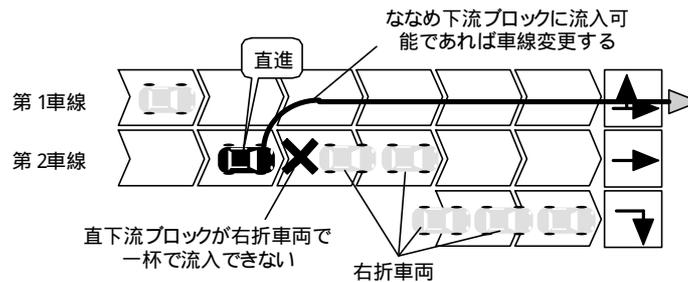


図5: ハイブリッドブロック密度法での車線変更の組み込み

2.3.3 信号による交通制御

ブロック流入時だけでなく、流出時の規制情報を記述することで、さまざまな機能をモデルに与えることができる。

図6には信号交差点での交通流の制御をモデル化したものを示してある。リンク下流端のブロックには車両の進行方向にたいする流出のパーミッションの情報が与えられ、信号現示に従って動的に変更される。信号現示が青の場合は全ての方向の流出が許可されているので、どの車両もブロックから流出することができるが、右折専用の青矢印のような現示のときは、右折車両のみが当該ブロックから流出できる。

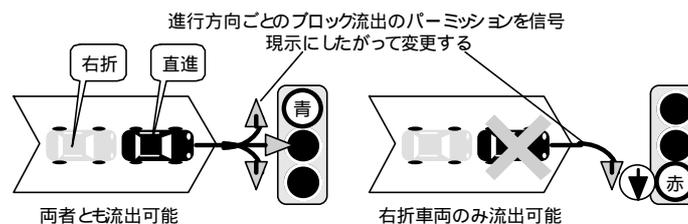
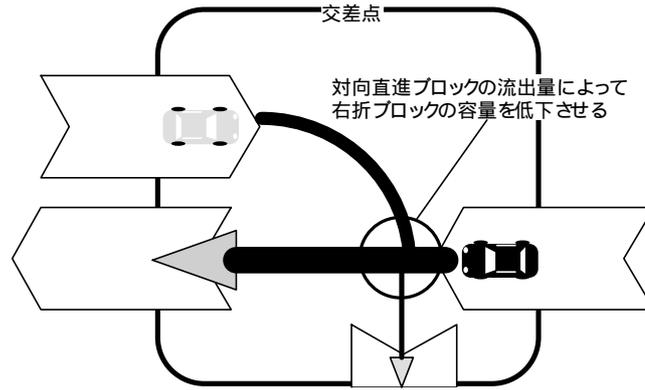


図6: 信号により流出が制御されるブロック

2.3.4 対向直進交通による右折容量の逓減

交差点での右折容量は対向直進交通の影響を受けて逓減する。ハイブリッドブロック密度法では交差点部の右折ブロックの流出容量に、対向直進ブロックの流出量によって決められる逓減率を乗じている。逓減率には、対向直進の1スキャンでの流出量を1時間当たりの交通量に換算し、それに応じたギャップアクセプタンス確率 f の値を用いている。これにより、対向直進交通量が増加すると、右折の捌け量が低下する現象をモデル化することができる。



q [台/時]	0	200	400	600	800	1000	1000超
f	1.00	0.81	0.65	0.54	0.45	0.37	0.00

図 7 : 右折車両のギャップアクセプタンス組み込み

2.4 マルチスキャンハイブリッドブロック密度法

ここまでのハイブリッドブロック密度法の説明では、ブロックの長さは一定長に固定されていた。このアルゴリズムの計算量は明らかにブロック数に比例するので、ブロック長が一定であれば、すなわちリンクの総延長に比例する。スキャン間隔を大きくとればブロック長も長くなるため、ブロック数が減少して計算量が軽減されるが、信号交差点でのモデル化を考えると、数秒のスキャン間隔でも流出量が正確に再現されず、精度の面で大きな問題が生じる。

信号現示を考えるとスキャン間隔は1秒程度が望ましい。この程度だとブロック長はおおよそ10~20mとなる。しかしながら、ハイブリッドブロック密度法では車両が存在していないブロックも計算対象となるため、ネットワークを短いブロックで細かく分割すると、計算効率が悪くなる。

そこで計算量を軽減するために、流量の再現性の精度が要求されるリンク下流端のブロックは短いスキャン間隔で、それ以外のあまり精度を要求されない区間では長いスキャン間隔のブロックを配置する、マルチスキャンハイブリッドブロック密度法を提案する。すなわち、図8に示されるように、スキャン間隔をリンク下流端ブロックでは1秒とし、上流に向かって徐々に整数倍していくものである。実用上はスキャン間隔を2倍ずつ長くしていき、最大2~16秒という適当な上限を決めてある。

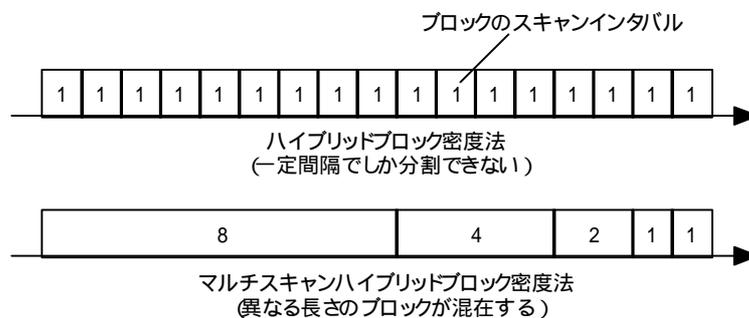


図 8 : マルチスキャンハイブリッドブロック密度法でのブロック構成

このようなブロック長の設定に応じて、密度改訂の計算アルゴリズムも変更される。すなわち各ブロック i は時刻 t がそのスキャン間隔 dt_i の倍数となるときだけ、(式1)の流出可能量計算と式2)の流入可能量計算の対象となる。それぞれの値は次に計算されるまで変数に保持される。

同様に、各ブロック i は時刻 t がそのスキャンインタバル dt_i の倍数となるときだけ、(式 3)による上流ブロックからの流入量計算の対象となる。このとき求められた流入量を用いて、自らの密度を更新する。

このとき、上流ブロックの密度と変数に保持していた流出可能量から、この流入量を差し引いておく。またリンクとリンクが接続する境界では、1秒スキャンのブロックが長いブロックの上流になるため、1秒ごとに上流ブロックが下流ブロックに交通量を「押し込む」ような例外的処理をおこなう必要がある。

このようにアルゴリズムを変更した場合、長いスキャンのブロックでは密度分布が平滑化されるものの、以前と同様に設定された交通量 - 密度曲線に基づく密度管理が可能であることが特徴としてあげられる。計算量は、一定長で分割した場合と比べてブロック数が減少するだけではなく、毎スキャンにおいて全てのブロックを計算対象とする必要がなくなるため、大幅に軽減される。豊田市ネットワークにおけるモデル検証の例では、ノード数 84、リンク数 193、リンク総延長 73km という条件で、ブロック数が 20% に、1スキャンあたりの計算量が 15% にまで軽減された。また実際のシミュレーション実行時間も 1/4 程度に短縮された。

ただしブロックの分割を粗くすることによる誤差も発生する。リンクの接続部分では 1秒ごとに上流リンクの下流端ブロックが下流リンクの上流端ブロックに車両を押し込むのに対し、そこから車両が流出するのは、さらに下流のブロックのスキャンインタバルごととなる。そのため、たとえばリンク上流端ブロックのスキャンインタバルが 16 秒の場合、車両がそのブロックに流入してくるタイミングによっては、本来 16 秒後に下流ブロックに流出するはずのところを、1 秒だけブロック内に滞在しただけで流出してしまうとも考えられる。このように車両の旅行時間が各リンクを超えるごとに、最大で (上流端ブロックのスキャンインタバル - 1) 秒短縮される可能性がある。

3. 経路選択モデル

AVENUE は確率的経路選択挙動を取り込み、複数の利用者層を設定することができるようになっている。以下では AVENUE の経路選択モデルについて述べる。

3.1 交通の発生・集中

時間帯別 OD 交通量データは、発生点ごとに目的地別の発生交通量として与えられる。発生交通の時間変動到着パターンはランダム到着を仮定し、1秒ごとの到着台数が計算される。発生した車両は、ネットワーク端点から流出するリンクの最上流ブロックに押し込まれ、ネットワーク上を移動していく。

目的地の端点に到達した車両は、ネットワークから取り出され消滅する。

3.2 経路の列挙

AVENUE ではシミュレーション実行前に、各 OD ペアについて選択候補となるすべての合理的な経路 (efficient path) が列挙される。デフォルトではリンク長を用いて efficient path を設定しているが、適当な外部プログラムで作成した経路を読み込ませることも可能である。

また、実際のシステムでは実用性を考慮して、ユーザ指定の経路を追加・削除することもできる。これは距離だけを基準として efficient path を探索しても、そのなかには必ずしもとりうるすべての経路が含まれていない可能性もあるためである。ネットワークの規模が大きくなるにつれ、ユーザが経路を確認・修正するのに要する労力も大きくなるが、現実的にはたかだか 100 交差点程度の規模のネットワークへの適用を想定しているため、操作性のよいインタフェースを用意すれば、大きな問題にはならないと思われる。

3.3 経路の表現

経路の探索は、図3-10に示されるようなリンク-ターングラフ上でおこなわれる。これは道路ネットワークを表現するノード-リンクグラフにおいて、リンクを点に縮退させ、そのリンクから進行することができる別のリンクへの枝をターンとしたものである。

ターンには左折/直進/右折/Uターンのラベルが付けられる。これはノード-リンクグラフでは、右折禁止やUターン禁止のような進行方向規制を陽に表現することができず、経路探索アルゴリズムだけでこの規制を考慮することが困難であるためである。また実際のネットワークでは同じリンクでも直進と右折とでは所要時間が異なる場合もあり、このような状況もノード-リンクグラフでは表現することができない。リンク-ターングラフではターンにコストを持たせることができるので、より実際的な経路選択が可能である。

リンク-ターングラフ上で探索された経路は、ターンのリストとして表現される。したがって経路が決定されると、自動的に各リンク下流端での進行方向も決定され、前出の車線選択のモデル化のところで述べたように、適切な車線を選択することができる。

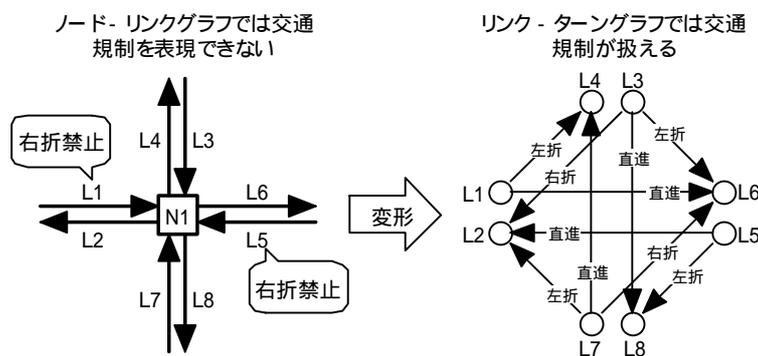


図9:リンク-ターングラフ上での経路選択

3.4 経路の選択

AVENUEではネットワーク利用者層を、まず次の2つのグループに大別している。

- 1) 経路指定利用者層
- 2) 経路選択利用者層

経路指定利用者層に属する車両は、交通状況にかかわらずユーザが指定した経路を通る。これは路線バスや、案内誘導に従う車両などを表現したものである。またシミュレーションで再現される旅行時間を測定するプローブ車両もこのグループに属する。

経路選択利用者層は提示された経路コストをもとに経路を選択する。経路選択挙動はロジット式の確率選択モデルとして表現されている。ロジットモデルにはコスト差に対する選択確率の感度をコントロールするパラメータが用意されており、 θ が大きいほどわずかのコスト差でも、よりコストが小さい経路候補を選択するようになる。よって、 θ を十分大きくとれば、最小コスト経路を選択するモデルにほぼ等しくなる。

経路選択利用者層は、旅行目的や経路選択挙動に応じて、異なった経路コスト関数、感度パラメータを持った複数の設定が可能である。コスト関数は経路長、旅行時間、右左折回数などの関数として表現される。

図10は経路選択のモデル化を模式化したものである。リンクには目的地ごと、利用者層ごとの経路コスト情報を格納した情報表示盤のような属性が定義されている。経路情報更新部は交通流再現部によるネットワーク上の交通状況を観測し、一定時間ごとにリンクの情報表示盤の内容を更新する。車両はリンク流入時に自分の目的地と属する利用者層に応じた経路コスト情報を参照し、ロジットモデルに従って経路を選択する。

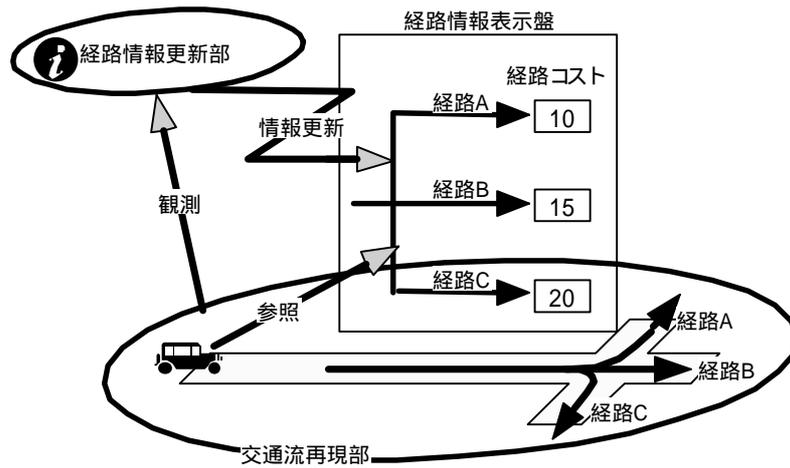


図 10 経路選択モデルの実装イメージ

3.5 経路コストの質が経路選択挙動に与える影響

前述の経路選択挙動のモデル化は、与えられた経路コストのもとで、ロジットモデルに従って経路選択をするというのだが、経路コストの内容によってモデルの挙動が大きく異なる。

広く認知されている利用者均衡状態は、ドライバーが実際経験する旅行時間を基準に経路選択を行った結果であるが、将来の交通状況を完全に予測しなければならない。このためシミュレーションのような動的な配分手法では取り扱いが困難である。

AVENUE では現在の旅行時間をコストとした、リアクティブな動的均衡配分を仮定している。これはシミュレーションでの取り扱いが容易であるに加え、可変情報表示盤や VICS、ATIS などの新たな情報提供の枠組みが近年整備が進められていることから、さまざまな要求に応えるモデルである。

図 11 に示されるネットワークを用いて、両者の挙動の違いを理論値で確認する。ネットワークは 2 経路からなり、自由速度で通過した場合 3 分の時間差がある。与えられた需要に対して、リンク 0~2 は十分な容量を持っているが、リンク 3 はボトルネックとなる。簡単のためロジットのパラメータは θ とし、渋滞時のボトルネックでの合流比は 1:1 とする。

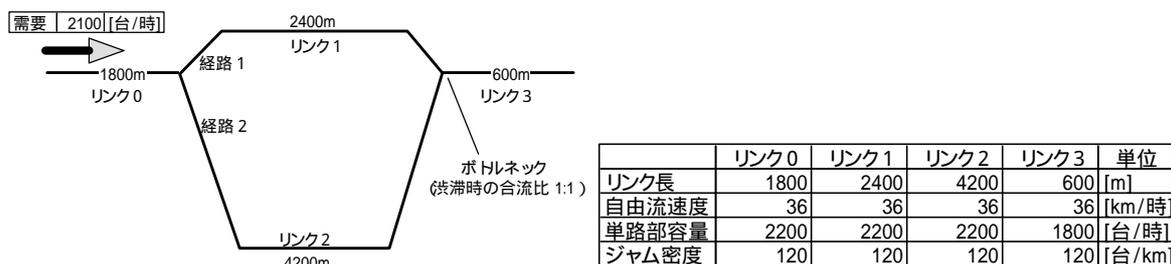


図 11 2 経路からなる単純なネットワーク

図 12 は実際にドライバーが経験する旅行時間をコストとして経路選択を行った場合のリンク 1 とリンク 2 のそれぞれの累積通過交通量とリンク旅行時間のグラフである。当初は旅行時間の短い経路 1 をすべての車両が選択するが、21 分経過した時点で両者の将来の旅行時間が等しくなるため、両経路に交通は等分されるようになる。経路コストも 21 分を経過してからは両経路とも等しく増加していき、利用者均衡状態になることを示す。

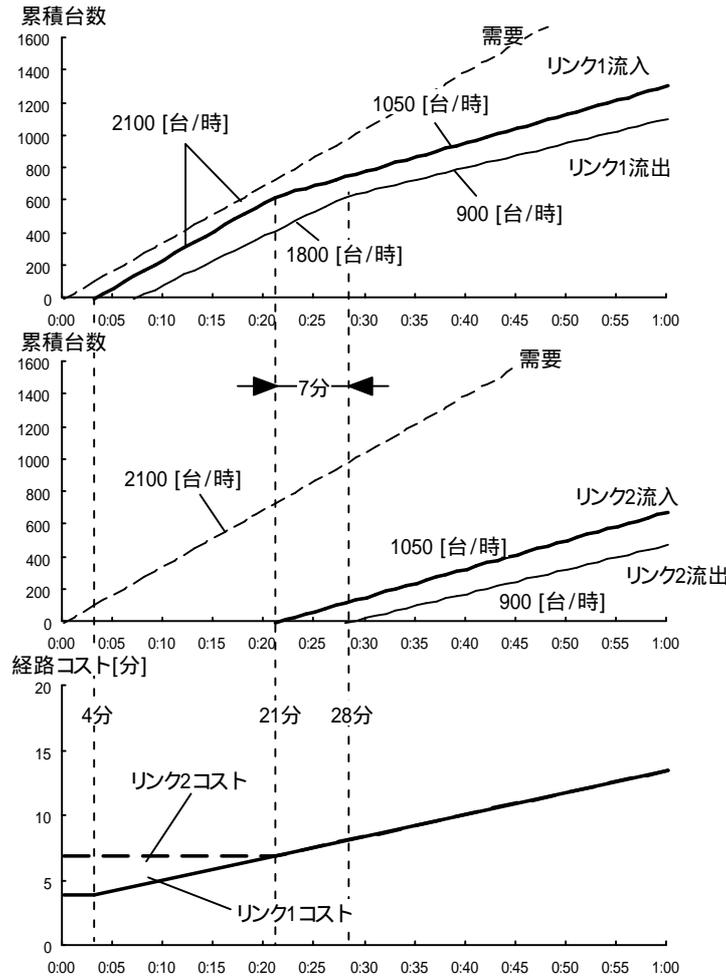


図 12 動的な利用者均衡状態(DUE)での配分結果

一方、図 13 は現在の旅行時間をコストとして経路選択するときの、同様のグラフであるが、この場合は 21 分を経過した時点でも、その瞬間のリンク 1 の旅行時間は 7 分より短く、依然としてすべての車両は経路 1 を選択する。28 分を経過して初めて、両経路の旅行時間は等しくなるが、次の瞬間にはすでにリンク 1 に流入してしまった余分な車両があるため、リンク 1 の旅行時間が 7 分を超えてしまう。そのため利用者均衡の場合のように交通は等分されず、28 分からはすべての車両が経路 2 を選択するようになる。

初めてリンク 2 に流入した交通が流出するまでの間、リンク 2 の旅行時間は 7 分のままであるが、その間もリンク 1 の旅行時間は増加し続けるため、両者の旅行時間差も広がっていく。リンク 2 から交通が流出し始める 35 分後からも、合流比を 1:1 と仮定しているのでそれぞれの流出交通量の流率は等しく、旅行時間差は縮まらない。結局、リンク 1 とリンク 2 の旅行時間が逆転するのは、余分に「リンク 1 に流入してしまった車両がすべて流出して、リンク 1 上に車両が存在しなくなった 38 分後となる。

以下、同様の状況が繰り返され、結果としてリンク 1 とリンク 2 を選択する交通量は周期的に振動する。この振動の周期は、経路の距離の差や需要のボトルネック容量に対する超過の程度によって変化する。

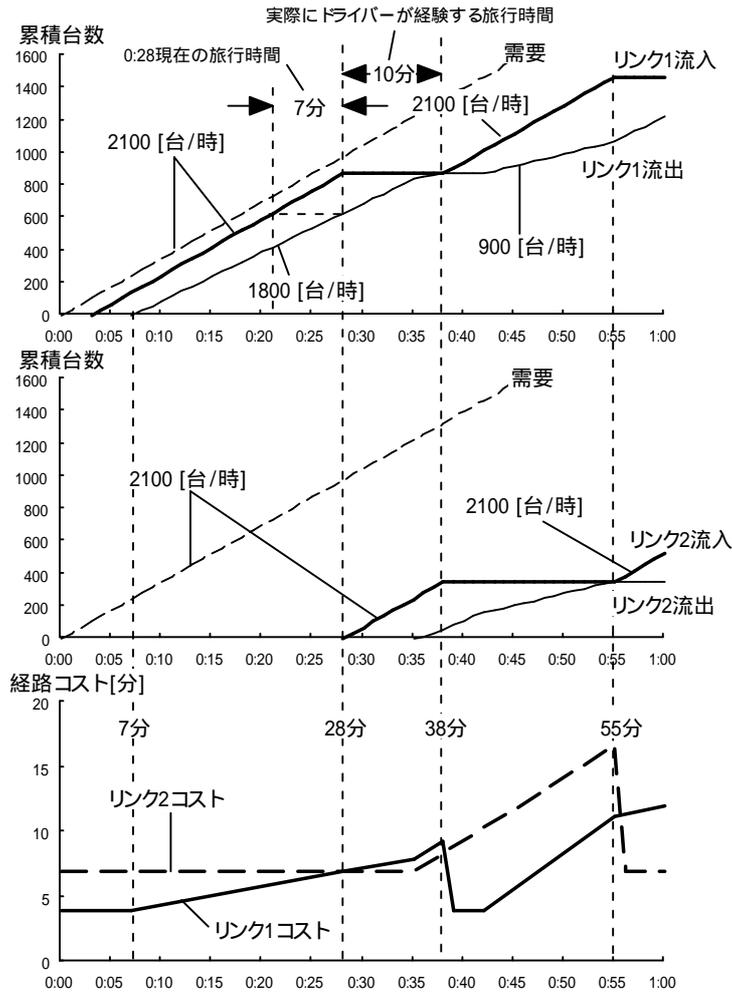


図 13 動的な利用者最適戦略(DUO)に基づく配分結果

現在の旅行時間をもとに経路選択を行う場合、ドライバーが実際に経験する旅行時間は、渋滞の成長過程では現在のものより長くなり、逆に解消過程では短くなる。同様の現象が首都高速道路などで観測されており、この仮定が現実的であることを裏付けている。

AVENUE ではロジット式の確率選択を仮定しているが、感度パラメータ が大きい場合はここで示した の挙動に近づき、 が小さくなるにつれ長い振動周期を示すようになる。また、経路コストを一定の間隔で更新しているが、上記の理論的な挙動を再現するためには、この更新間隔をリンクの旅行時間に比べて十分短くする必要があるのである。

仮に更新間隔を長くすると、現在の旅行時間が短い経路に「余分に」流入する交通量が多くなってしまつため、旅行時間が再び逆転するまでの時間が長くなり、振動の周期が増加する割合が増えると考えられる。逆に十分短い更新間隔をさらに短くしても、モデルの示す挙動はかわらない。

一般に街路では、交通は1~2分程度のサイクル長の信号で制御されるため、数10秒~1分程度の更新間隔で十分短いといえる。またリンクの旅行時間として、直前に通過した車両の旅行時間を探ると信号現示のタイミングによって大幅に変化してしまうので、更新間隔の間に通過した車両の旅行時間の平均値をリンク旅行時間としている。

4. オブジェクト指向による交通モデルの記述

図 14 に AVENUE のネットワーク交通モデルを構成する主要なオブジェクトクラスの階層図を示す。実線の矢印はクラス階層の継承を表し、点線の矢印はクラス間に定義された関係を意味している。関係名や関係の基数は省略した。太い枠で囲まれているクラスはインスタンスオブジェクトを持つクラスである。以下では、各主要クラスの属性、機能などについて概説する。

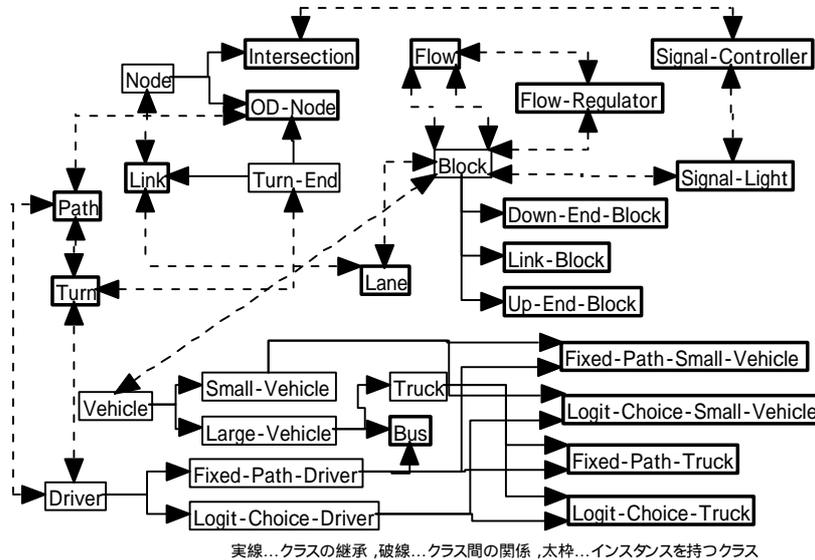


図 14 AVENUE の主要なオブジェクトクラス階層図

4.1 ノード・リンク (Node, Link)

道路ネットワークをグラフ表現したときの点と有向枝は、それぞれノードとリンクとしてクラス定義される。

ノードはそれに流出入するリンクとの関係を持つ。ノードのサブクラスには交差点 (Intersection) と OD ノード (OD-Node) の 2つが定義される。OD ノードはネットワーク端点やセントロイド、駐車場などの交通が発生・集中するノードで、交通量の発生パターンを属性として持つ。

リンクは上流ノードと下流ノードとの関係を持ち、長さ、車線数、右左折専用ポケット長、容量、停止線部飽和交通流率、自由流速度、ジャム密度などの属性を持つ。また後述する車線との関係も記述される。

リンクと OD ノードは次に述べるターン端点のサブクラスでもある。これはリンク・ターングラフに変形したときに、どちらもターンの始点・終点になりうるためである。

4.2 経路・ターン・ターン端点 (Path, Turn, Turn-End)

AVENUE の交通モデルでは経路がオブジェクトとして作成される。経路はターンのリストとして表現され、その起終点となる OD ノードと関係を持つ。

ターンとターン端点はリンク・ターングラフにおける枝と点に相当する。ターンはその始点と終点になるターン端点と関係を持ち、ターン方向のラベルを属性として持つ。ターン端点はノード・リンクグラフでの OD ノードとリンクに対応する。

4.3 車線 (Lane)

車線は実際の道路上での車線と同じもので、右左折専用ポケットも一つの車線オブジェクトとして表現される。車線はそれが属するリンクおよびその車線上のブロックとの関係を持ち、長さや車線規制に関する情報の属性

を持つ。

4.4 ブロック・フロー・フロー整流器 (Block, Flow, Flow-Regulator)

ブロックはハイブリッドブロック密度法の計算をおこなう主体となるオブジェクトである。その属性には計算で用いられるスキャンインタバル、ブロック長、ブロック臨界密度、ブロックジャム密度といったパラメータに関するもの、ブロック内車両密度のような状態量、ブロックに存在する車両のリストなどが定義される。

あるブロックの上流と下流ブロックの間にはフローオブジェクトが生成され、隣接するブロックどうしはフローを仲介して関係づけられる。フローには直上流 / 直下流か、斜め上流 / 斜め下流といったラベルが属性として与えられ、車線変更の挙動の再現に利用される。またブロック間交通量や補正交通量もフローの属性として表現される。

フロー整流器は、あるフローの状態を観測し、制御の対象となるブロックの流出容量を調整するオブジェクトである。対向直進交通による右折容量の低下や無信号交差点での合流、織り込み区間の合流などはこのオブジェクトを用いて表現する。

4.5 車両・ドライバー (Vehicle, Driver)

車両は目的地の情報を持ち、ブロック間を移動することでネットワーク上を流れるオブジェクトである。サブクラスには小型車両 (Small-Vehicle)と大型車両 (Large-Vehicle)があり、さらに大型車両はトラック (Truck)とバス (Bus)に分かれる。車両の属性には、出発地、出発時間、目的地、車種、乗用車換算係数などが定義されている。

ドライバーは経路選択をおこなうためのオブジェクトで、経路選択基準によって固定経路ドライバー (Fixed-Path-Driver)と確率選択ドライバー (Logit-Choice-Driver)にわけられる。それぞれのドライバーがコストや経路選択基準に関する情報を持ち、選択した経路に関係づけられている。

実際にインスタンスを生成するクラスは車両とドライバーからの多重継承を受けており、確率選択乗用車 (Logit-Choice-Small-Vehicle)や固定経路乗用車 (Fixed-Path-Small-Vehicle)など細分化されている。

4.6 信号制御機・信号灯器 (Signal-Controller, Signal-Light)

信号制御機はそれが設置される交差点、およびその交差点に流入するリンクの下流端に設置される信号灯器との関係を持つ。信号制御機の属性にはサイクル長、絶対オフセット、相対オフセットの基準交差点のほか、信号現示 (Signal-Phase)オブジェクトのリストも定義される。この中にある信号現示の種類に従って、スプリット時間ごとに信号灯器の表示を変更する。

信号灯器は交通流を制御する断面をふくむ位置にあるブロックと関係を持つ。自信の表示状態が信号制御機によって変更されると、それに従ってブロックの流出方向のパーミッションを変更する。