

鉄道運行計画におけるグラフ理論の適用

--理論の基礎と運転整理問題への適用可能性

長崎 祐作*, 江口 誠, 古関 隆章 (東京大学)

Application of graph theory to scheduling of train operation
----fundamental theory and possibility of its application to realtime
train rescheduling

NAGASAKI Yusaku, EGUCHI Makoto, KOSEKI Takafumi (The University of Tokyo)

Abstract

In train operation management system, it is important to simulate and estimate futural train operation. Furthermore, we have proposed that it is also important to simulate and estimate passengers' behavior. In this paper, we introduce a method to simulate train operation and passengers' behavior with graph theory, which is very fast, hence suitable for realtime train rescheduling.

1. はじめに

鉄道の運行管理システム(列車運行計画、運行監視、運転整理システムなど)においては、将来の列車の運行を予測することがしばしば重要となる。列車の運行は、車両の性能と線路の条件に加え、信号システムの制約など複雑な条件に基づいて決まってくる。こうした条件を正確に反映して将来の列車運行を推測するには走行曲線を描くなどの複雑な計算をする必要が出てくる。しかし実際には駅間の基準走行時分、駅の基準停車時分などを基にした簡単なシミュレーションで十分な場合が多い。こうした場合に用いることが出来るPERTダイアグラムに基づく列車運行シミュレーションの研究^[1]を最初に紹介する。

また、筆者らはかねてから乗客の立場からみた運転整理案の評価が重要であると主張してきた^{[2][3]}。筆者らの提案する乗客の立場からみた運転整理案の評価をするためには、まず最初に乗客行動仮定に基づいて乗客の行動を計算する必要がある。この計算を迅速に行うことが出来る、最短経路探索法に基づく乗客行動シミュレーションの手法^[4]を紹介する。

2. PERTダイアグラムによる列車運行シミュレーション

グラフ理論でいうグラフ構造のうち、重み付きのリンクを持った有向非循環グラフ(Directed Acyclic Graph: DAG)を用いた、互いに依存関係のある仕事の日程計画などを行うために使われるPERT (Program Evaluation and Review Technique)ダイアグラムというものがある。このPERTダイアグラムの考え方を列車運行シミュレーションに応用した先行研究^[1]があり、我々の研究でもこれを用いているのでここで簡単にまとめて紹介する。

2.1. PERTダイアグラムの考え方

図1にPERTダイアグラムの例を示す。互いに依存関係の

ある仕事の日程計画を行うためのPERTダイアグラムにおいては、グラフのリンクが何らかの作業を表し、リンクの重みはその作業を完了するのにかかる時間を意味する。あるノードから出ているリンクと、そのノードに入ってくるリンクは、入ってくるリンクが表す仕事が全て完了してからでないとして出て行くリンクが表す仕事に着手することができないという関係を表すように配置されている。図1で、ノードを表す の中に書き込まれている数字がその位置に達するまでにかかる時間を意味している。出て行くリンクしかないノードはそこが始点になるので、かかる時間は当然0である。そこから、リンクの重みに従って各ノードに到達するまでにかかる時間を書き込んでいくが、複数のリンクが入り込むノードは、それらのリンクが表している仕事全てが終わる時間を書き込むことになる。このようにPERTダイアグラムに時間を書き込むことで、全ての仕事が終わるのにかかる時間がどの程度か、各作業工程のうちどの部分を効率化すれば全体の時間を短縮できるか、などを簡単に掴むことができる。

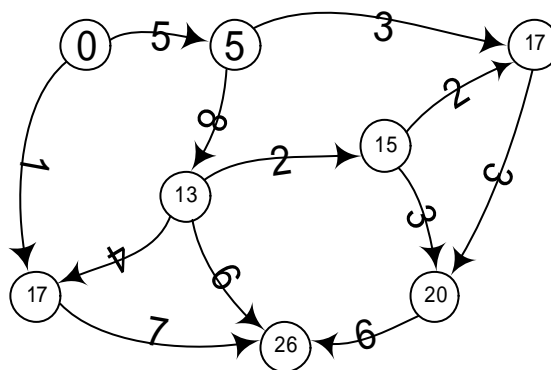


図1. PERTダイアグラムの例

Fig 1. An example of PERT diagram.

2.2. PERTダイアグラム計算アルゴリズム

2.1.に示したPERTダイアグラムを計算するアルゴリズムを紹介する。まず、各ノードにトポロジカルオーダと呼ばれる値を割り当てる。トポロジカルオーダとは、全てのリンクに対して、(リンク元のノードのオーダ) < (リンク先のノードのオーダ)の関係が満たされるように、各ノードに与えられた数値である。このような条件を満たす数値の組み合わせは無限にあるので、一意に定めることは出来ない。PERTダイアグラムにおいては、リンクの向きにしたがってたどることができるノード間でしかオーダを比較することができないので、お互いにオーダを比較することができないノードが含まれていることがある。すなわち、各ノードには半順序関係があり、ノードにトポロジカルオーダを割り当てることは、各ノードの半順序関係に従ってノードを並べることに相当し、トポロジカルソートと呼ばれる。トポロジカルソートのアルゴリズムは次のようになる。

1. 全てのノードに未到達を示すフラグを立てる。
2. 未到達フラグの立っているノードがあれば、その中から適当に1つを着目する。なければトポロジカルソート完了。
3. 着目しているノードから出ている全てのリンクを順にたどって次のノードに着目し、4.に進む。そのノードからたどるべきリンクが全てなくなったときには、そのノードを出力して到達フラグをたて、1つ手前のノードに戻る(最初に着目したノードの時は2.に戻る)。
4. 着目したノードが既到達ノードであった時は何もせず3.に戻って次のリンクをたどる。未到達ノードであった時は、そのノードから再帰的に3.を実行する。

このアルゴリズムによって、ノードが順番に出力されてくるが、このときトポロジカルオーダが大きい順に整列されてくる。従ってこの順に適当な値を振れば、トポロジカルオーダを与えることが出来る。

一度トポロジカルオーダが与えられてしまえば、オーダが小さい順番にノードをたどって、図1に示したようなPERTダイアグラムの所要時間の数値を計算していくことは簡単である。

2.3. 列車運行シミュレーション

2.2.までに示してきたPERTダイアグラムの考え方とアルゴリズムは、信号制約などの複雑な条件を考慮しないで簡便な列車運行シミュレーションを行う時にも適用できる。この場合、各ノードは列車の駅への到着や出発などの事象を表し、ノード間のリンクは各事象の間の遷移関係を表し、リンクの重みは時間となる。

このグラフを作成するためにはまず、全ての列車の駅への到着と出発を表すノードを作成する。続いてそれらのノードを結ぶリンクとその重みを設定する。リンクは次のようなものを設定する。

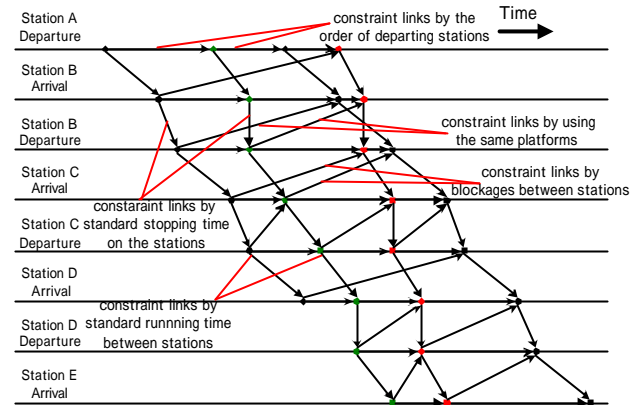


図2. 列車運行の制約を表したグラフ

Fig 2. A graph representation of train operation constraint

1. 各列車の各駅出発ノードとその次の駅の到着ノードの間に張られる、駅間走行を表すリンク。リンクの重みはその列車の駅間走行基準時分。
2. 各列車の各駅到着ノードとその駅の出発ノードの間に張られる、駅への停車を表すリンク。リンクの重みはその列車のその駅における標準停車時分。
3. 各列車の駅出発ノード・到着ノードと、その次の列車の駅出発ノード・到着ノードとの間に張られる、駅出発・到着順序制約を表すリンク。リンクの重みは最小出発・到着時隔。
4. 各列車の駅出発ノードと、同一ホームを次に利用する列車のその駅への到着ノードの間に張られる、同一ホーム利用制約を表すリンク。リンクの重みは最小同一ホーム利用時隔。
5. 列車の駅到着ノードから、n本後の列車の前駅出発ノードの間に張られる、閉塞制約を表すリンク。ただし、nは駅間の閉塞数。リンクの重みは0。
6. 基準となるノードから、各列車の各駅出発・到着ノードとの間に張られる、ダイヤ制約を表すリンク。リンクの重みは正常運転時の出発、到着時刻と基準時刻の差。

このうち、6は正常なダイヤより早く走らないという目的のために加えられる制約のリンクである。6以外の各制約を表すリンクを付け加えたグラフの例を図2に示す。

このようにして作ったグラフに対して、PERTダイアグラムの計算をすることで各列車の各駅における到着時刻、出発時刻が得られることになる。

2.4. 運転整理時の列車運行シミュレーション

2.3.に示した方式で正常運転時の列車運行シミュレーションを行うことが出来る。これに対して、事故が発生するなどして列車の運行が遅れたときに、列車の運行順序を変更したり、時間調整をしたりしてダイヤを変更する、「運転

整理」が行われるが、このときにも再度列車の運行をシミュレーションする必要がある。運転整理時の列車運行シミュレーションにはまず、最初に遅れた列車を表すノードに対して基準ノードから遅れの時分だけの重みを持ったリンクを張って遅れを表現する。それから、運転整理の対象となった列車に対して、時隔調整や順序変更などの条件を表すよう適宜ノードとリンクの付け替え、削除、追加などを行う。修正グラフが完成した後再度PERTダイヤグラムの計算を行って修正後の列車運行時刻を得るが、この際に修正があったノードのうち一番小さなトポロジカルオーダーを持つものより、小さなオーダーを持つノードには変更の影響が及ばないという性質を利用して、単純に再計算するより計算量を顕著に縮減することが可能である。

3. 最短経路探索法による乗客行動シミュレーション

2.では列車の運行をシミュレーションする方法を紹介したが、これに対して3.では乗客の行動をシミュレーションする方法を紹介する。これにはグラフ理論のアルゴリズムのうち、ダイクストラ法という方法を用いる。ダイクストラ法自体は有向グラフでないグラフに対しても用いることが可能であるが、ここでは有向グラフに対して用いている。

3.1. ダイクストラ法の考え方

ダイクストラ法はリンクに重み付けされたグラフ上で、始点とするノードから任意のノードまで、経路上のリンクの重みの合計が最短となるような経路を探索するのに用いられるアルゴリズムである。一般にリンクの重みの合計は距離を表していることが多い。乗換案内プログラムのような最短経路探索プログラムに良く用いられるアルゴリズムである。そのアルゴリズムは次の通りである。

1. 始点のノードに確定した重み合計値として0を入れる。始点のノードを着目しているノードとする。
2. 着目しているノードから全ての隣接ノードに対して、そのノード間のリンクの重みを自分の重み合計値と合計した値を暫定の重み合計値として記入する。ただし既に確定しているノードは変更しない。暫定の合計値が記入されていたノードの場合、より小さな重みを選択して記入する。
3. 全ての暫定値が記入されたノードのうち、一番小さな暫定値が記入されているノードを選んでその暫定値を確定値に変え、今着目しているノードとする。全てのノードが確定値に変われば終了し、そうでなければ2.に戻る。

非有向グラフに対して適用した場合はこのようになるが、今回の乗客行動シミュレーションに関しては有向グラフに対して適用することになる。この場合、2.でリンクをたどる時、リンクの向きにしかたどらないということになる。

3.2. ダイヤの評価値

乗客の行動シミュレーションをすることは、最終的に乗客の立場から見たダイヤの評価値を算出することが目的である。評価値は乗客にとっての損失の積み上げとして表すものとする。従って、評価値が大きいほど悪いダイヤということになる。評価値が最も良くなるのは、乗客にとっての損失が全く無くなるときであり、それはどういう状態であるかを考えると

1. 乗客が駅にやってくると待ち時間無く乗車する列車がやってくる。
2. 乗車した列車は乗客の目的地の駅までノンストップで(最小の所要時間で)走る。
3. 乗車した列車には他の乗客は乗っていない(混雑度が0である)。

という3点が考えられる。3については混雑度0が理想的な状態であるが、厳密には自分自身が乗車した時点で混雑度は0とはならない。

実際にはこの条件を全ての乗客について満たすことは不可能であるので、この状態からの損失を計算してダイヤの評価値とする。評価項目は以下の3点である。

3.2.1. 損失時間

損失時間は、出発駅から到着駅まで直行する列車があったと仮定した場合の所要時間に比べてどれだけ余計に時間がかかったかを表すものである。従って、正常運転時であっても乗換や待ち時間があるため損失は発生している。損失時間の評価式は式(1)に示した通りである。

$$L_1 = w \times \sum_{i=0}^N (t_i' - t_i) \quad (1)$$

N : 乗客の総数

t_i : i 番目の乗客の直行時の所要時間

t_i' : i 番目の乗客の運転整理時の所要時間

: 時間価値(円/分)

時間価値 は、乗客の時間あたりの賃金をもとにした所得接近法により、都市圏ではおよそ50[円/分]と求められている^[2]。

3.2.2. 乗換回数

乗換は、実際に乗換にかかる時間以上に乗客に負担となるため、乗換にかかった時間を損失時間として評価する以外に、その回数自体を評価する。乗換1回を2分に換算して、それを金額に換算する。

$$L_3 = 2 \times w \times \sum_{i=1}^N r_i \quad (2)$$

N : 乗客の総数

: 時間価値(円/分)

r_i : 乗客 i の乗換回数

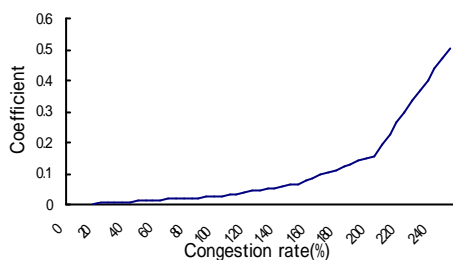


図3 混雑度と係数の関係

Fig. 3. Characteristic curve of the human feeling factor against congestion rate

3.2.2. 混雑度

混雑度の評価については、混雑時に1分を何分に感じるか、人間の感覚の非線形性を表した係数^{[5][6]}を用いて時間に換算し、それにその混雑を経験した人の人数と時間をかけ、これに時間価値をかけて金額に換算する。

$$L_2 = \mathbf{w} \times \sum_{j=1}^{n-1} \sum_{k=1}^{H_j} \{f_c(c_{jk}) \cdot q_{jk} \cdot t_{jk}\} \quad (3)$$

n : 全駅数

\mathbf{w} : 時間価値

H_j : 駅 j に停車する列車本数

c_{jh} : 駅 j に h 番目に停車する列車の、次の駅までの区間における混雑度

q_{jh} : その乗客数(人)

t_{jh} : その所要時間(分)

f_c : 図3に示す混雑に対する人間の非線形な感覚を表す係数

3.3. 乗客行動仮定

3.2.に示した評価値を元に、乗客は自分の損失が最も少なくなるような経路を選択して乗車するものとする。ただし、3.2.3.の混雑度については、乗客すべての行動が決定してから計算できるものなので、それを事前に予測して行動するというのは時系列の逆転が起きてしまう。また、そのような行動を計算しようとすると均衡計算が必要となって計算量が増大するため、乗客は3.2.3.以外の評価値の合計が最低となるような経路を選ぶものとする。これをまとめると以下の通りである。

1. (規則上認められていないので)乗客は逆方向乗車をしないものとする。
2. 損失時間の評価値と乗換回数の評価値の合計がもっとも少なくなる経路を選択する。
3. その経路が複数ある場合は、所要時間が最も少ない経路を選択する。
4. その経路が複数ある場合は、目的駅に最も近い駅で乗り換える経路を選択する。
5. その経路が複数ある場合は、乗換駅に最も早く到着する経路を選択する。

1の逆方向乗車禁止の条件は、初期グラフ作成の段階で自動的に制約として入る。2の評価値の合計が最も少なくなる経路を選択する条件は、ダイクストラ法を用いて探索することによって得られる。3以降の条件は、最後に出てきた複数の選択肢に対してプログラムで条件判定することによって選び出す。

ただし、この仮定で乗客行動を一意に決定できるのは直線的な路線を対象とする場合のみであり、環状線などのループを含む場合には別な仮定が必要となる。

3.4. 乗客行動シミュレーション

3.1.で示したダイクストラ法の考え方と、3.2、3.3で示した評価値の計算方法、乗客行動仮定の考え方を組み合わせることで、乗客行動をシミュレーションして評価値を算出することができる。このとき、各ノードは列車の各駅への到着、出発を表し、リンクはその間での乗客の行動を表す。またリンクの重みはその行動に対する評価値を表す。

このグラフを作成するためにはまず、全ての列車の駅への到着と出発を表すノードを作成する。続いてそれらのノードを結ぶリンクとその重みを設定する。リンクは次のようなものを設定する。

1. 各列車の駅出発ノードから次の駅への到着ノードへのリンク。乗客がその列車に乗って移動したことを表す。リンクの重みは損失時間の金額換算。
2. 各列車の駅到着ノードから出発ノードへのリンク。乗客がその駅で乗降せず乗り続けていることを表す。リンクの重みは損失時間の金額換算。
3. 全ての乗り換え可能駅で、ある列車の到着ノードから、その列車から乗換可能な列車の出発ノードへのリンク。乗客がその駅で乗り換えたことを表す。リンクの重みは損失時間の金額換算と乗換損失の金額換算の合計。
4. 列車の終着駅で、到着ノードから後続の同種別列車の出発ノードへのリンク。終着駅からまだ先に行く乗客が続行の列車に乗り換えたことを表す。リンクの重みは損失時間の金額換算と乗換損失の金額換算の合計。

3.において設定するリンクはできるだけ減らすことが計算量の縮減に重要であるため、異種別列車でしか乗換は起こらないと考えてよいことを利用して、種別ごとに直後に来る列車1本に対してのみリンクを張る。4.は終着駅におけるこの制限の例外である。

可能性のある乗り換えリンクを漏れなく張るためには、運転区間が異なる列車はたとえ停車駅パターンが同じであっても異種別の列車として取り扱う必要がある。

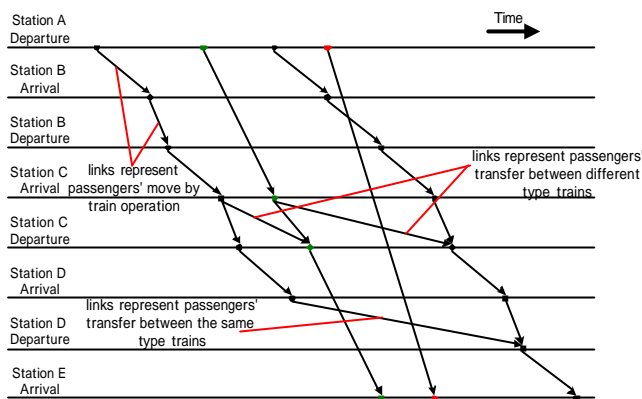


図4 乗客行動を表したグラフ

Fig 4. A graph represents passengers' behavior

この方法により作成した乗客行動を表すグラフを図4に示す。このグラフに対してダイクストラ法を適用すると、ある始点からの評価値の一覧が出来上がることになる。そのうち、各駅への到着ノードの評価値の中から一番低いものを選んでいくと、その始点から全ての駅への経路が定まることになる。任意のODに対して全ての可能な始点でこの作業を繰り返すと、全ての駅間での乗客経路を決定することができる。一部同一評価値を持った複数の経路が出てくることが考えられるが、3.3に示した乗客行動仮定に基づいて後ほどプログラムの処理で唯一に決定する。

乗客経路が決定すると、都市近郊鉄道においては乗客は時刻表を気にすることなく、乗車駅に時間的に均等に現れるものと考えられるので、同一ODに対して考えられる複数の経路に対して、その経路の時間間隔に比例してODを配分することで、各経路の乗客数を決定することができる。

3.5. 積み残しに関する考慮

3.4.の計算をして乗客行動を計算すると、一部の列車に乗客が集中して混雑度が異常に高くなることもある。一部の列車の混雑度が異常に高くなるようなダイヤはもともと良くないダイヤであるので、表1に示したような混雑度と評価値の換算係数を混雑度が高い場合に極端に高く設定することで、そうしたダイヤの評価値を悪くするという考え方もある。しかし、ある一定以上の混雑度は非現実的なため、一定の混雑度で乗客の乗車を差し止めて次の列車に回すという考慮を含めることも出来る。そのためには、3.4.の計算が終わった時点で閾値以上に混雑している列車とその区間を選び出す。そして、積み残しが出た場合には続行の同一種別列車に対する乗換も生じると考えられるので、関係する列車の間に乗換リンクを新たに設定する。それから、積み残しの対象となる乗客を選び出し、それぞれの目的地に応じて新たな経路を割り当てる。

積み残しの対象となる乗客は図5のような考え方で選ぶ。

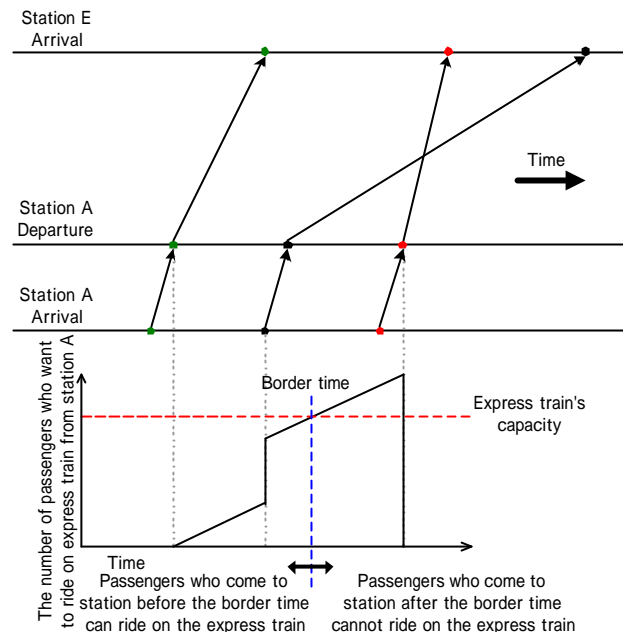


図5 積み残し対象乗客選択法の考え方

Fig 5. An idea of method to select bumped passengers

考え方の基本は、最初から当該列車に乗っている乗客は対象とせず、積み残しが発生する駅から当該列車に乗車しようとする乗客のみを対象とする。そして、乗客がその駅に現れた順番にホームに並んで待っているものとみなして、閾値を超えるまで順番に選んでいく。

図6に積み残し考慮前と考慮後の乗客行動の変化を示す。リンクに添えられている数字はそのリンクの表す列車に乗っている乗客数を表し、ノードに添えられている数字はそのノードが表すのが到着イベントの時はその駅で降りた乗客数、そのノードが表すのが出発イベントの時はその駅から乗った乗客数を表している。積み残し考慮前は乗客が限度を超えて乗っている列車があるが、考慮するとその列車に乗るはずだった乗客が後続の列車に乗るように変化しており、その乗換のリンクが加えられている。またこのとき後続の列車で再度混雑度が限界を超えたため、積み残しの考慮が再び行われてその後続列車に乗客が移っている。

3.6. 運転整理時の乗客行動シミュレーション

運転整理時には、列車運行シミュレーションの結果に応じて2.4.と同様に再度グラフを構築しなおす。列車の順序関係が変わっていないところでも、時間間隔が変われば乗客の行動経路が変化するので、そうした変化がおきたところからダイクストラ法による再探索を行って経路を繋ぎかえることになる。また、各経路間の時間間隔が変わると、各経路に割り当てられる乗客数は変化するので、時刻が変化したノードについては全て乗客数の再割り当てを行う必要がある。

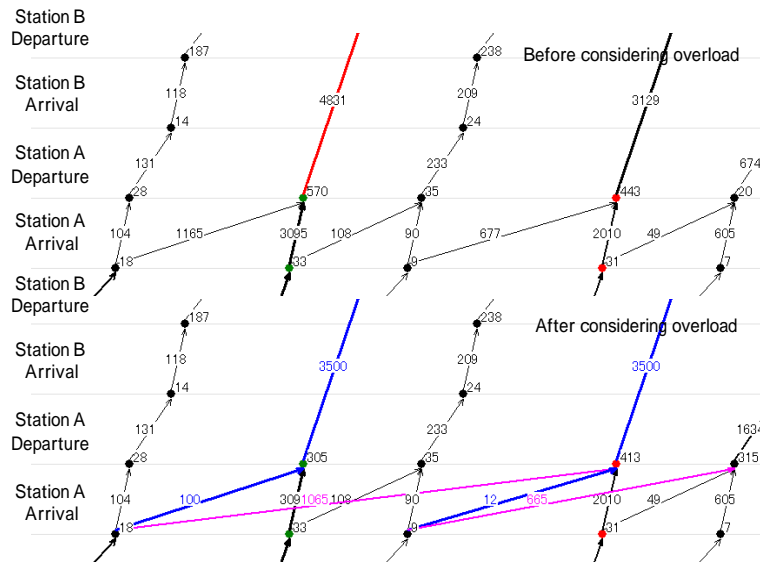


図6 積み残し考慮前と考慮後の乗客行動

Fig. 6. Passengers' behavior before and after consideration of overload

4. まとめ

本論文では、グラフ理論の考え方であるPERTダイアグラムとダイクストラ法を用いて、列車運行シミュレーションと乗客行動シミュレーションをする方法を紹介した。

グラフ理論を用いることの最大の利点は、これらのアルゴリズムをコンピュータプログラムとして実装する際に、既存のライブラリなどを簡単に流用することができるということが挙げられる。また自力で実装するとしても、既に計算量などの証明が専門家によってなされた効率の良いアルゴリズムを用いることができるので、一から自分で考えたアルゴリズムに比べて、バグも計算量も少なくできるという利点がある。さらに、グラフ理論の考え方に問題を帰着することはアルゴリズムとデータを分離することにつながり、バグの修正や新たな機能の追加といった点でプログラミング上の数々のメリットをもたらす。

さらにユーザにとっての利点として、グラフ理論を用いるとプログラム内部の動作・状態の可視化がしやすいため、対話型の運行管理支援システムを構築する際にユーザに分かりやすいものを作ることができる。また、時間順にイベントを処理していくような方式に比べるとずっと高速に処理することができるのでリアルタイム性を確保できる。

逆に、今回用いたグラフ理論のアルゴリズムではリンクの重みやリンクとノードの配列が動的に変化するような系を取り扱うのは不得手であり、それゆえに先行列車に追いついて信号の制約を受けて列車の速度が変化し、駅間走行時間が変化するというような状態を考慮に入れることができない。したがって精度を求めるシミュレーションでは用いることが出来ないといえるが、駅間閉塞数の条件や最低時隔の条件は含めることが出来るので、これだけでもある程度の精度でシミュレーションをすることができる。また

乗客行動シミュレーションについてはこれらの問題は無いため、列車運行シミュレーションだけを別の方法で行って、それにこのアルゴリズムでの乗客行動シミュレーションを組み合わせることも出来る。

このアルゴリズムを用いたプログラムは実際に実装されており、運転整理案の評価などに用いられている。これからはさらに複雑な路線条件に対応させるなどの改良作業を行っていきたく考えている。

参考文献

- [1] 安部, 荒屋 「最長経路法を用いた列車運行シミュレーション」, 情報処理学会論文誌, 1986
- [2] 林, 古関 「都市圏鉄道における運転整理の評価と効果的手法の提案」, 平成13年電気学会全国大会 4-234, 2001
- [3] 長崎, 古関, 村木, 館, 駒谷 「都市近郊鉄道における運転整理案の作成と評価」, 平成14年交通・電気鉄道 リニアドライブ合同研究会 pp.11-16, 2002
- [4] 江口, 長崎, 古関 「都市近郊鉄道における運転整理時の乗客行動経路再決定の高速化手法」, 平成15年電気学会全国大会 5-239, 2003
- [5] 「鉄道プロジェクトの費用対効果分析マニュアル99」, 運輸政策研究機構, 1999
- [6] 美谷, 家田, 畠中: 「乗車位置選択行動モデルを用いた混雑費用の定量的評価法」, 土木計画学研究・論文集, No.5, pp.139-146, 1987