Ⅳ-24　列車運行パターンの最適化について

神戸大学工学部　正　員　枝村 俊郎
神戸大学工学部　正　員　森津 秀夫
神戸大学大学院　学生員　○土井 元治

1. まえがき

鉄道の利便性は運行する列車の停車駅をどのように設定するかによって、大きく左右されるものと考えられる。このような停車駅の設定つまり列車運行パターンはこれまでにもいくつかあったが、必ずしも十分なものではなかった。われわれは列車運行パターンがネットワークとみなせることから、最適ネットワーク問題の解法を適用することを試みた。ここでは、最適ネットワーク問題としての定式化とその解法について述べる。

2. 列車運行パターンのネットワーク

駅をノードで表わし、すべての運行パターンを示すと図-1のようなネットワークができる。このネットワークでのリンクは、その両端の駅を直通で結ぶ列車の運行を表わしている。図-1のネットワークは図-2に示すネットワークと同値であり、完全連結網になっている。いま、列車運行パターンを各駅間でそれぞれ定めた運行系統数以下になるものを求めるとすると、図-1では(I)~(IV)のカットに含まれるリンク数を、計画運行系統数以下にすればよいことになる。この制約条件下で実行可能解を求めると、例えば駅数6、各駅間での計画運行系統数をすべて2とした場合3から4へは、3，2，5，4のような経路をとらねばならない。これは3，4間に各駅停車のない場合であるが、このような解も場合によっては検討の価値があると思われるので、ここでは実行可能解に含めることにする。ただしネットワークが非連結になってはならないことは言うまでもない。

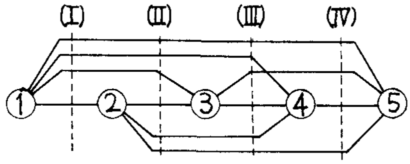


図-1　駅数5のときのネットワーク

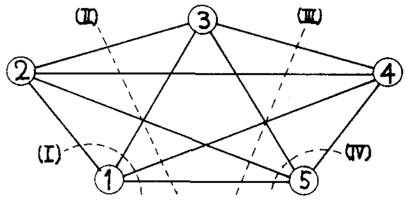


図-2　図-1と同値なネットワーク

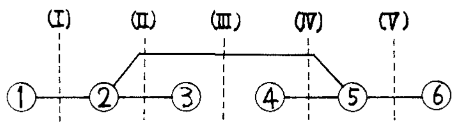


図-3　駅数6のときの実行可能解例

3. 問題の定式化

各駅間で計画運行系統数以下という制約下で、乗客のトリップ総所要時間を最小にするものとすると、次のように定式化できる。

$$\min Z = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} g_{ij}\, t_{ij}(x_1, x_2, \cdots, x_m) \qquad (1)$$

$$\text{s.t.} \sum_{k=1}^{m} C_{lk} x_k \leqq M_l^c \quad (l=1,2,\cdots,n-1) \qquad (2)$$

$$x_k = 0 \text{ or } 1 \quad (k=1,2,\cdots,m) \qquad (3)$$

ここに、Z；トリップ総所要時間，g_{ij}；i j ノード間の最短所要時間，x_k；kリンクがネットワークに含まれるとき $x_k=1$，含まれないとき $x_k=0$，C_{lk}；lカットにkリンクが含まれるとき $C_{lk}=1$，含まれないとき $C_{lk}=0$，M_l^c；lカットに含まれるリンク数の上限値，すなわちlカットでの計画運行系統数，m；リンク数，n；ノード数である。この計画運行系統数は、需要量や鉄道施設、採算性などを考慮して決められるものとする。

ここでは簡単のために通過待ちはないものとする。それは、どの駅でどの列車の通過待ちをするのが最適かを同時に決定しようとすれば、各系統の運行回数などダイヤそのものを作成する必要があるからである。ここで求めたいものは基本的な列車運行パターンなので、そこまで問題を複雑にしないことにする。

4. 問題の解法

この問題の特徴は、複数の制約式を持つことであるが、基本的な考え方は制約式が1式のみの場合と大差ない。そこで、枝村・森津の最適ネットワークアルゴリズム[1]を応用する。このアルゴリズムでは目的関数の下限値を使っているが、この問題では次のように定式化できる。

$$\min F = \sum_{k=1}^{m} f_k y_k \tag{4}$$

$$\text{s.t.} \sum_{k=1}^{m} C_{\ell k} y_k \geq \sum_{k=1}^{m} C_{\ell k} - M_\ell^c \quad (\ell = 1, 2, \cdots, n-1) \tag{5}$$

$$y_k = 0 \text{ or } 1 \quad (k = 1, 2, \cdots, m) \tag{6}$$

ここに、f_k；kリンクを除くことによる目的関数増加の下限値，y_k；kリンクがネットワークに含まれるとき $y_k = 0$，含まれないとき $y_k = 1$ である。式(5)は式(3)と同値であり、ある計算段階で各制約式を満たすためには、あとどれだけリンクを除かなければならないかを示す。

式(4)～(6)で示される問題は一般に多次元ナップザック問題と呼ばれるものである。これを解けば、目的関数の下限値 Z^ℓ は、

$$Z^\ell = Z_0 + F \tag{7}$$

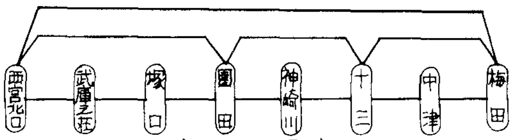
によって求めることができる。しかし、多次元ナップザック問題の最適解を短い計算時間で得ることは困難なので、ここでは千住・豊田[2]の近似解法を用いる。そのため求めた解の目的関数値 F' は、F よりも大きいことがある。したがって、F' を使って目的関数の下限値を定義することはできないのであるが、実用上はこれで十分なので、次式により $Z^{\ell'}$ を定義し、これを目的関数の下限値として扱うことにする。

$$Z^{\ell'} = Z_0 + F' \tag{8}$$

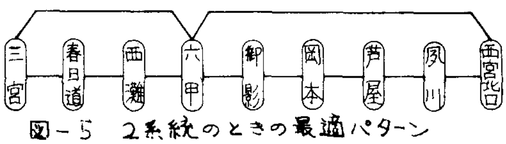
まとめると、ある節点で式(4)～(6)を解き $Z^{\ell'}$ を求める。暫定最適解に対する目的関数値 Z^u と $Z^{\ell'}$ を比較し、$Z^{\ell'} > Z^u$ ならバックトラックする。$Z^{\ell'} \leq Z^u$ なら、その解を使用して分枝し、目的関数 Z を求める。$Z < Z^u$ なら Z を新たに Z^u とし、バックトラックする。これをくりかえし、最終的な暫定解がこの問題の解となる。

5．適用例

阪急電鉄神戸線を例にとり計算を行った。データは、昭和50年の交通センサスから朝のラッシュ時のものを用いた。その結果、梅田－西宮北口間で各駅間の計画運行系統数を3にしたときの解が図－4、西宮北口－三宮間で計画運行系統数を2にしたときの解が図－5である。図－4において梅田－西宮北口間ノンストップの列車を十三に停車させ、十三－園田－西宮北口を塚口停車にすれば、実際に使用されているパターンになる。また図－5では六甲停車をやめ、西宮北口－三宮間ノンストップにすれば実際に使用されているパターンになる。



図－4　3系統のときの最適パターン



図－5　2系統のときの最適パターン

6．あとがき

ここでは、列車運行パターンをネットワークとして扱うことにより、これまでの手法よりも合理的なアルゴリズムを開発した。問題を単純化して扱っているが、適用例で明らかなように、かなりよい結果が得られた。

参考文献

1) 枝村俊郎・森津秀夫：最適交通ネットワーク問題の厳密解法と近似解法，土木学会論文報告集，第262号 P39～P53，1977年6月．

2) Shizuo Senju・Yoshiaki Toyoda： An Approach to Linear Programming with 0-1 variables, Management Science, vol.15, No4, Dec., 1968.