

組合せ最適化による東京オリンピックの問題解決

石川県立金沢泉丘高等学校 2年

北川 周哉 木下 颯馬 高橋 諒弥 谷口 和弥 堀 聡一郎

1. 要旨、概要

現在、わが国では2020年オリンピック・パラリンピック両大会の話題で持ちきりである。特に開催都市東京では、これを催すにあたって数々の事業が構想、或いは既に実行されており、世間でもその動向は注目されている。東京五輪は日本経済に大きな利益をもたらすと考えられているが、しかしながら、同時に様々な問題も生じるだろうと考える。私たちは、それらの問題のいくつかは、数学的な視点から解決策を導けるものなのではないかと考えた。そこで私たちが用いたのは、組合せ最適化と呼ばれる考え方である。この考え方では、離散値が用いられた複雑な組合せ構造をもつ問題に対して、最適解を求めることができる。現実問題は、数学的に見るとこのような複雑な組合せを持つことが多いため、この考え方をを用いるのは有効である。そのため、本研究では、特に地下鉄での交通困難の問題、宿泊施設の不足の問題の2つを取り上げ、組合せ最適化によりこの2つの問題の解決策を求めることにした。交通困難の問題では、1時間당りに運べる観客の人数の最大値を、宿泊施設不足の問題では、宿泊客の人数に対して部屋の過不足が最も小さくなるような部屋数の最適値を求めた。その結果、交通困難の問題に関しては、求める最大人数が32,500人となったが、これは解の精度を下げたことで求めたものであるため、今後は解の精度を追求していくことにする。宿泊施設不足の問題に関しては、立式の段階で複雑になり、計算を行うことができなかつたため、今後は式の簡略化を行うことによって解を求めることにする。

2. 研究目的

本研究の目的は組合せ最適化を用いて東京五輪開催にあたって生じる問題を解決することである。今回、私たちが目をつけた問題は、立地問題、交通困難、財政圧迫、治安悪化、宿泊施設の不足の5つである。この中でも、交通困難

と宿泊施設の不足については日本人だけでなく外国人にも問題となり、かつ数学的手法である組合せ最適化で解くことができる。東京五輪では外国人旅行者が多くなることが予想されるため、この2つの問題を取り上げた。

3. 研究方法

組合せ最適化とは変数が離散値をとる最適化問題を解く手法のことで様々な用途に使われている。離散値というのは、整数のように連続ではなくばらばらな値の事であり、最適化問題とはある物事を実行するときに、最も効率よく行うためにはどうしたら良いかというような問題のことである。身近な例で言うと、家から駅に行くときに、どのような道を通っていくと最も早く、安全にいけるのか、などのようなものがある。

また、今回全ての計算で計算ソフトPythonを用いている。このPythonというのは、プログラム言語のことなのだが、コードが簡単であり、そのため初心者でもとても扱いやすいものである。また、現実問題で出てくる数値は必ず整数のみとなっている。例えば、人の数を数える場合などは1人、2人と整数で数える。Pythonは整数についての計算に適している。また、今回はすべてについて総当たりで調べるため、式が正確であれば、出た答えは必ずその式のたった一つの正確な解となる。そのためPythonの計算についてはかなり正確といえる。よって、私たちはPythonを用い、計算した。

～交通における問題 Part 1～

本問題の研究では3つの会場に訪れる観客の人数の大小を考慮するために、訪れる観客の予想人数をまとめたデータをもとに、各会場に向かう観客の人数比を条件に取り入れた。このことより、観客を無制限に会場に運ぶのではなく1人1人が希望する会場に行けるという状況を再現する事ができ、より現実問題に近い条件で解を求められるようにした。これが私たちの追

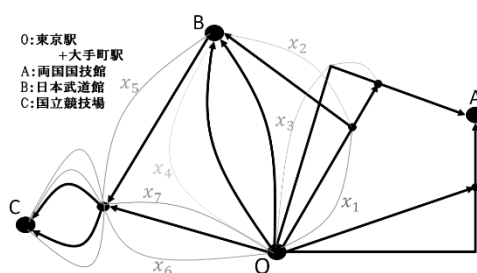
求してきた、より現実に近づけるという独創的思考に即するといえる。

<方法>

まず、東京五輪の主な目的地として、日本武道館、両国国技場、新国立競技場の3つを選び、東京駅からの路線図を調べ(図1)、それを基にグラフ化した(グラフ1)。



(図1)



(グラフ1)

このグラフ1における各点は駅と目的地、各線は路線を表しており、線についている矢印は人の移動方向を表している。これにより、問題が必要な情報のみで簡略化して表され、理解もしやすくなった。また、複雑な現在世界そのものを考えずとも、モデルをもとに考えて解決に近づく事ができ、東京五輪の観客輸送の問題にも可能であるという発見もできた。そして、駅と目的地と路線の関係がわかりやすくなった。

次に、このグラフ1をもとに定式化した。観客輸送の問題では輸送人数の最大値を求めるので地下鉄の各路線に乗る人数のうち、ほかの目的地に行く路線に分岐していたり、合流していたりしている路線に乗る人数を変数とし、それぞれ $x_1 \sim x_7$ とおいた。次に、それぞれの路線で、12時から13時の間に目的地に到着できる列車の本数、また、一本の列車に乗ることができる人数を調べ、その条件を元に立式した。

次に、それぞれの目的地へ行く一日当たりの予想人数を調べ、その人数の比をとりその式も

加え、最大値を求めるための式とした(式1)。

$$\max f(x_1, x_3, x_4, x_6, x_7) = x_1 + x_3 + x_4 + x_6 + x_7$$

$$\begin{cases} x_1 \leq 15 \times 600 \\ x_2 \leq 10 \times 600 \\ x_3 \leq 3 \times 600 \\ x_4 \leq 12 \times 600 \\ x_5 \leq 10 \times 600 \\ x_6 \leq 10 \times 600 \\ x_7 \leq 15 \times 600 \end{cases} \quad \begin{cases} x_1 - x_2 \leq 10 \times 600 \\ x_1 - x_2 + x_3 \leq 10 \times 600 \\ x_5 + x_6 \leq 10 \times 600 \\ x_6 + x_7 \leq 15 \times 600 \\ x_n \in \mathbb{N}(n = 1, \dots, 7) \end{cases}$$

$$(x_5 + x_6 + x_7) : (x_2 + x_4 - x_5 + 12 \times 600) : (x_1 - x_2 + x_3 + 10 \times 600) = 7 : 3 : 2$$

(式1)

最後にこの式1を計算ソフトPythonを用いて計算した。今回は最も単純な方法である、全ての場合を総当りで調べるという方法で計算させた。

<結果>

式1を計算すると丸一日かかっても終わらず、計算の途中経過を見ても進み具合から計算が終わるのに非常に時間がかかると考えた。

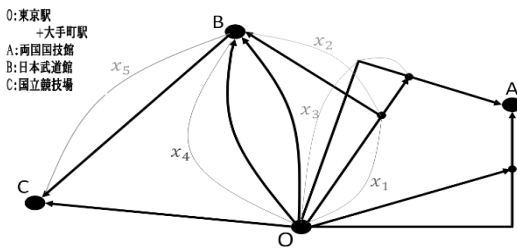
<考察>

まず計算が進まなかった理由を考えた。それは計算の演算処理数が多すぎたために、組合せ爆発が起こっていたからだと考える。計算量が多い理由は、変数が多いことと、変数の範囲が大きいことにあると考えた。Pythonではすべての変数の組み合わせを計算するので変数が多く変数の範囲が大きいほど計算時間が長くなるからだ。実際に、私たちが用いたコンピュータの演算速度が1秒当たり15000回であることを踏まえて考えると、計算を終えるのに約 4.80×10^{14} 年かかることが分かった。よって、現実的な時間では計算し終えることは不可能であると考えた。そこで計算を行うために、式を上記の観点から修正しなければならないと考えた。

～交通における問題 Part2～

<方法>

まず、一つ目の修正で変数を減らした。東京駅から国立競技場へのグラフが分かっている所を一つにまとめた。次に、100人を1つの塊とし、100人単位で人が移動すると考えることで変数の範囲を1/100まで減らした。このことにより、計算量も $1/10^{10}$ になった。そして再びグラフ化した(グラフ2)。



(グラフ 2)

このグラフ 2 を元に Part1 のときと同様にし
てそれぞれの変数のとりうる値とそれぞれの目
的地に行く予想人数を定め、今回はすべての変
数の範囲を 1/100 にして、立式した(式 2)。

$$\max f(x_1, x_3, x_4) = (x_1 + x_3 + x_4 + 90 + 72 + 60) \times 100$$

$$\begin{cases} x_1 \leq 15 \times 6 \\ x_2 \leq 10 \times 6 \\ x_3 \leq 3 \times 6 \\ x_4 \leq 12 \times 6 \\ x_5 \leq 10 \times 6 \end{cases} \quad \begin{cases} x_1 - x_2 \leq 10 \times 6 \\ x_1 - x_2 + x_3 \leq 10 \times 6 \\ x_5 \leq x_4 \\ x_i \in \mathbb{N} (i = 1, \dots, 5) \end{cases}$$

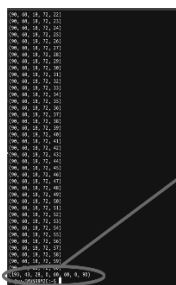
$$\frac{(x_5 + 15 \times 6)}{\text{Cに行く人}} : \frac{(x_2 + x_4 - x_5 + 12 \times 6)}{\text{Bに行く人}} : \frac{(x_1 - x_2 + x_3 + 10 \times 6)}{\text{Aに行く人}} = 7 : 3 : 2$$

(式 2)

最後に Part1 と同様に総当りで Python を用
いて計算した。

<結果>

求める式である、 $\max f(x_1, x_3, x_4) =$
 $(x_1 + x_3 + x_4 + 90 + 72 + 60) \times 100$ を計算
すると、 $x_1 = 45$ 、 $x_3 = 0$ 、 $x_4 = 60$ となり、
計 32,500 人運べることがわかった(図 2)。



$$\begin{aligned} \text{最大値} \\ \max f(43, 0, 60) \\ = (43 + 0 + 60 + 90 + 60 \\ + 72) \times 100 = 32500 \end{aligned}$$

(図 2)

<考察>

観客の最大輸送可能人数は求まったが、この
ときの x_3 の値は0となっている。これは x_3 の
区間の地下鉄に乗っている観客の人数が0人
であることを意味しているが、現実の場合を考
えると0人というのはいない。つまり、最大
値 32500 という値は最適解とはいえないので
ある。そのため、次に計算式の変数の範囲を再び

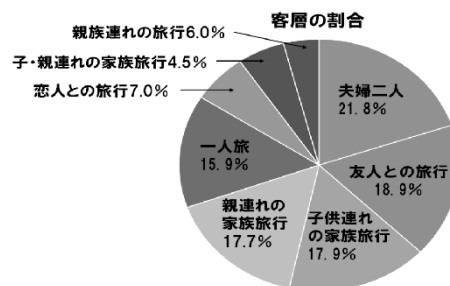
広げて解を求めることにした。計算を行う際、
計算式にある目的地に行く予想人数の比 7:3:2
という数値は、すべて±1まで動くことができ
るようにした。そして 1/100 にまで減らした変
数の範囲を、 x_1 は 4,250 から 4,350 まで、 x_2
は 2,750 から 2,850 まで、 x_3 は 0 から 50 ま
で、 x_4 と x_5 は 5,950 から 6,050 まで動かすこ
とにした。ところが、その場合でも解は
32,500 人となり、 x_3 も 0 のままであったため
結果的に精度が高まった解は得られなかった。

～宿泊における問題～

本問題は、宿泊施設の部屋数の最適化が目的
である。部屋数と宿泊客の組数の差の 2 乗平均
の最小値を求めることによって部屋数、組数の
ばらつきをその値の大小で評価した。したがっ
て、この最小値を求めることで部屋数の最適値
を求めたのである。このことにより、部屋数が
永遠に増えることで、最適値を出せても現実問
題に即さないといったことをなくした。これら
より私たちが意識してきた、より現実に近づく
という独創的思考に即するといえる。

<方法>

初めに、私たちはどのような客が宿泊施設に
泊まるのかを調べるために年間を通しての宿泊
旅行の同行形態に関するデータをインターネット
から集め、それをもとに円グラフを作成した
(グラフ 3)。



(グラフ 3)

次に、この同行形態の中で一概には人数を決
めることができないところの人数を場合分け
し、24 通りの同行人数による割合を出した。

そして、この 24 通りすべてについて、部屋
数の過不足が最も小さくなるように式を立て
た。このとき、過不足の評価をするためにそれ
ぞれの部屋に入る客の組数とその部屋数の差の
2 乗をとり、その平均の最小化をすることを考

えた。ここで、2乗平均をとったことにより、大きく外れれば外れるほど、値が大きくなり、近ければ近いほど、値は小さくなるというように過不足の正当な評価ができていたと考えた。

(式3)

$$\min f(x_1, x_2, x_3, a_i, b_i)$$

$$= \frac{1}{3 \times 24} \sum_{i=1}^{24} \{(p_{1i} + a_i - x_1)^2 + (b_i + p_{3i} - x_2)^2 + (p_{4i} + p_{5i} - x_3)^2\}$$

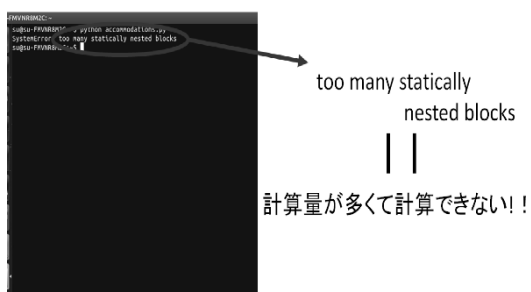
$$\left\{ \begin{array}{l} n = 1, \dots, 24 \\ x_1 + x_2 + x_3 = 1000 \\ x_i \in \mathbb{N} \quad (i = 1, 2, 3) \\ p_{mn}: m \text{人で泊まりに来る組の組数} \quad (m = 1, \dots, 5) \\ p_{1n} + p_{2n} + p_{3n} + p_{4n} + p_{5n} = 1000 \\ p_{2n} = a_n + b_n \\ a_n, b_n \in \mathbb{N} \end{array} \right.$$

(式3)

最後に、それぞれの部屋数を変数 $x_1 \sim x_3$ として動かし、この式3をPythonを用いて総当りで計算した。

<結果>

Pythonに計算させようとする、too many statically nested blocksと出力され、まったく計算されなかった(図3)。



(図3)

<考察>

結果に書いた英文の意味は、簡単に言うとコードが長すぎてPythonが読み取ることができない、ということである。つまり、場合分けが多すぎたために起こっており、特に、2人組の客は2種類の部屋に入ることができるためにループする場所が多くなりすぎていた。このままでは、そもそも計算してくれず、また、計算量を考えてみても最初の交通についての問題と比較してもはるかに多い。交通における問題Part1の式では、先に述べたように、計算結果が出るまでに、約 4.80×10^{14} 年かかると分かっている。このことから、これよりも計算量が多いこの式は例え計算を始めてくれても計算結果

は出ないということが予想できる。そのため、この式の答えを出すためには上記の問題点を解決しなければならないと考えた。そのためには、まず今回、もっとも大きく計算量の多さに関係している2人組の客が2種類の部屋に入ることができるということについての変数の範囲を小さくできればいいと考えた。さらに、ほかの解決方法としては、式の整数という条件をとりはずし、線形計画法などを用いて求めることができるのではないかと考えた。しかし、これは整数しか用いられないPythonでは計算できないため、ほかのプログラム言語を用いる必要がある。

4. 結果

東京五輪において、地下鉄を用いて運べる観客の人数は理論上最も効率よく乗ったときに32,500人である。しかし、この時には電車に乗る人数が0の車両が存在している。

宿泊施設の問題に関しては、現時点ではまだ計算式が複雑すぎるため、コンピュータを用いて計算することはできない。しかし、誤差が少しでも出ることを許して計算式を簡単にすれば、現実的な時間で結果は得られそうである。

交通の問題と宿泊の問題のどちらにしても、現実問題を解決するに当たって、数学は一つの手法になりそうである。

また、僕たちは今回先行研究として、市川周らによる論文である「組合せ最適化による並列数値シミュレーションの静的不可分散」と、情報処理学会の中力雅人らの研究報告である「多重解像度による組合せ最適化を用いた離散平面認識」を参考にした。多重解像度の問題では離散平面を少ない点集合の低い解像度で大まかに直線を求めることで計算時間を削減している。またこのときにはx軸、y軸方向それぞれの値のうち分散が小さい軸方向で誤差が大きくなっているため、幅を特定できるような手法を探す必要がある。並列数値シミュレーションの問題では細部を抽象化したモデルの上で最適解を求めているが、この抽象化が妥当であるか否かは、実機上での評価が必要である。しかしこれは特定の事のみにおいてできるので、一般的性質を解明するにはまた別の事をする必要がある。ここで、どちらの研究もモデルを抽象化するこ

とで計算時間を減らす、という意味では同じようなことをしている。しかしそこから厳密な値を求めるときに、調べるべき範囲を絞ったり、特定の場のみにおいて調べる、などといった違いが見られる。自分たちの研究でも計算時間を減らすために変数のとる範囲を大まかにすることでモデルを抽象化した。厳密な値を求めるとき、特に交通のほうでは一度大まかに求めた値の付近についてより細かく調べた。

5. 考察

まず東京駅から各目的地へ運べる人数の総和は32,500人という結果によると、実際に各目的地に行きたい人を運ぶにはかなりの時間がかかることがわかる。また1時間のうちに運べる観客の数も現実と比較すると少ないことがわかる。さらに乗る観客の人数が0人と値がでた部分があった。修正前の式は私たちが真の値に近い値を出せるように考えたものだが、修正をすることでもともと得ようとしていた値とは必ず異なってくる。つまり、総じて得られた値は現実問題に当てはめると不十分であるところが多く、最適化モデル、つまり、式の条件、変数などの修正はまだ必要であると考えた。またこの結果に沿うならば、運びきれない観客はほかの交通機関を使う必要があり、例えば、バスなどの本数を増やすなどの兼ね合いが必要であると考えた。

次に宿泊の問題について考える。今回の計算では結果は得られず、また交通のときと同じように式の修正が必要になってくると思う。今回の修正ではできるだけ条件の変更によって得られる値が変化しないようにしなければならないと考えた。

全体を通じて、組合せ最適化を用いて現実問題の解決に至るには、求めるものを記号で表し式を立てなければいけない。この式を立てることと、計算をすることにおいて、式は正しくても計算できないこともあり、それらをうまく兼ねあうことが必要である。

6. 結論

本研究において組合せ最適化を用いたことから、最終的に確立した結果を求めるには至らず、次から次へとより正確な解を求める必要が

あった。そのため、交通と宿泊においてのどちらでも最終的に結果を出すことができなかった。しかし前述にもあるように、本研究では確立した解ではなく、近似値を出すことが目的だったため、その目的に近い答えは出すことができた。また、2020年東京五輪に実際に来る外国人や日本人観光客の人数を知ることはできず、推測することしかできなかった。ゆえに、年齢層による人数の割合や、各会場に行くであろう人数を曖昧な値で入れることになり、結果として正確さは劣ることになってしまった。ほかに計算の値が正確にならなかった原因として、Pythonの計算が容量が大きすぎると計算の回数が多くなってしまい、計算にかかる時間が人間の考えられないほど大きくなってしまふことが挙げられる。このことに関しては、近畿サイエンスデイで他校の生徒や様々な大学の教授の方により、計算をより簡略化し、かつより正確な値を見つける方法を教わった。それらを参考に計算を進めていくことで、さらに最適解に近づくことができるだろう。

7. 参考文献

- 1) 梅谷 俊治, 「大規模な組合せ最適化問題に対する発見的解法 - SlideShare」, 2016/9/16 www.slideshare.net/shunjiometani/ss-32225124
- 2) 「観客と大会スタッフの輸送需要 (PDF)」 www.shochi-honbu.metro.tokyo.jp/TOKYO2016_15_9.pdf [Google Map] <https://www.google.co.jp/maps>
- 3) B. コルテ (著), J. フィーゲン (著), 浅野 孝夫 (翻訳), 浅野 泰仁 (翻訳), 小野 孝男 (翻訳) 「組合せ最適化 第2版 (理論とアルゴリズム) 単行本」 2012/2/29
- 4) 井上 純一, 「2004年度 グラフ理論講義ノート: HUSCAP」 2004年度, <http://eprints.lib.hokudai.ac.jp/dspace/handle/2115/371>

8. 謝辞

今回の研究を行うにあたって担当の川本先生、金沢大学の塚先生、京都大学の高橋先生をはじめとする多くの方から多大なお力添えをいただきました。本当にありがとうございました。