簡単そうで難しい組合せ最適化
— 身近な話題から —

柳浦 睦憲

はじめに

「しりとりを最も長く続けたらいったい何語続けることができるだろう?」そんな素朴な疑問があるテレビ番組でまじめに (?) 取り上げられていました．組合せ最適化にはこのような身近な問題がたくさん含まれています．以下では身近な問題を中心に組合せ最適化の様々な話題を紹介します．

組合せ最適化とは

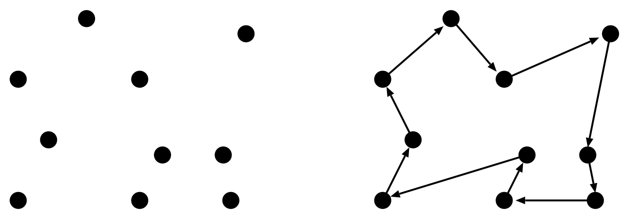
与えられた条件を満たす解の中で最も望ましいものを探す問題を一般に最適化問題と呼び, 組合せ的な条件が含まれている場合をとくに組合せ最適化問題といいます．いくつか例を見てみましょう．

最初の例は巡回セールスマン問題です．たとえば

郵便局, 銀行, 魚屋, 本屋, 八百屋, 肉屋,
パン屋, ケーキ屋, 花屋, たこ焼き屋

などの多くの場所に用事があるとき, できるだけ無駄な回り道をしないよう, 短いルートを考えてから行動するでしょう．これを一般的に書くと以下のようになります．n 個の目的地のそれぞれの間の距離が与えられたとき, すべての目的地をちょうど 1 度ずつ通って元の地点に戻るルート, すなわち巡回路の中で, 総距離最小のものを求めようという問題です．図 1-(a) に目的地数 $n = 10$ の例を示します．平面上の点が目的地で, 目的地間の距離はユークリッド距離[1]とします．図 1-(b) の回り方をはじめ, いろいろな回り方が有りますが, その中で最短のものを見つけたいわけです．実用的な例としては, 宅配やコンビニエンスストアの配送トラックのルートの決定が挙げられます．

次に, ナップサック問題を紹介しましょう．これは, n 個の商品それぞれの値段と満足度, および予算上限が与えられたとき, 予算の範囲内で買える商品の組合せの中で満足度の合計が最大のもの



(a) 目的地の配置　　(b) 巡回路の例

図 1. 巡回セールスマン問題の例

を求める問題です．（商品取合せ問題とでも呼ぶほうがイメージに合いますが, まあ, 買ったものをナップサックに入れると思って下さい.）小学校の遠足の時,「おやつは 300 円まで」というような規則があって, 何を持っていこうかあれこれ悩んだ, という経験を皆さんもお持ちでしょう．表 1 に商品数 $n = 6$ の例を示します．商品 2 と 5 を買うと満足度は 32, 商品 1 と 3 と 6 を買うと満足度は 23 となりますが, 商品 2 と 4 を買うおうとすると, 値段の合計が 11 円となり, 予算上限の 9 円を越えてしまいます．

表 1. ナップサック問題の例

商品番号	1	2	3	4	5	6
満足度	14	21	8	23	11	1
値段	3	5	2	6	4	1
予算上限			9			

簡単そうで難しい

上のふたつの例を見るかぎり, 組合せ最適化問題はそれほど難しくないもののように思えます．巡回セールスマン問題の可能な巡回路の数も, ナップサック問題の商品の組合せの数も, いずれも有限個なので, すべての可能性を列挙して一番よいものを選べば最適な解が得られるからです．しかしこのような列挙法は現実的でしょうか?

可能な巡回路の数は $n!$ 通り, 商品の組合せの数は 2^n 通りあります．これらは指数オーダーと呼ばれ, n とともに急激に増加するため, 列挙法は n が大きくなると（最近の高速な計算機を使ってもたかだか数十で）現実的でなくなります．

このタイプの関数が急激に増加する様子は,「鼠

[1] 平面上の 2 点 (x_1, y_1) と (x_2, y_2) の間のユークリッド距離は $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

算」や「曾呂利新左衛門の逸話」で実感できます. いずれもよく知られていて, 国語辞典や百科事典などに載っています. 後者を簡単に紹介しておきましょう. 曾呂利新左衛門が太閤秀吉から褒美をもらう際, 将棋盤の最初の目に米粒を 1 つ置き, 次の目に 2 つ, その次の目に 4 つと, 順次 2 倍ずつ置いた分だけ欲しいと言い, 秀吉ははじめは欲のないやつだと安請け合いしましたが途中で音をあげたというお話です. 約束を守った場合に秀吉が与える米粒の数は合計 $2^{81}-1$ 粒（約 4×10^{20} 合）となり, 1 億人がそれぞれ毎日 10 合（丼に 10 杯程度）のペースで 10 億年食べ続けても消費できない途方もない量になってしまいます.

これに対して, 計算時間が n や n^2 のように n^k（k は定数）に比例する時間で抑えられるとき, その計算時間は多項式オーダーであるといいます. これが指数オーダーよりも望ましいことは, 関数の形をグラフにしてみたり, 大きな n に対する具体的な値を計算してみたりすればすぐに実感できますので, 練習問題としておきましょう.

解きたい問題が多項式オーダーの時間（以下多項式時間）で解ければうれしいのですが, 残念ながら巡回セールスマン問題やナップサック問題をはじめとする多くの組合せ最適化問題は, NP 困難という難しい問題であることが示されていて, 多項式時間では解けないと考えられています. （NP 困難問題が多項式時間で解けないであろうことは $\mathrm{P}\neq\mathrm{NP}$ 予想と呼ばれ, 計算機科学における最大の未解決問題です.）

最適解は要らない?

お店をいくつかまわって買い物をするとき, 普通の人は最短巡回路を求めるのに時間と労力を費やすかわりに, そこそこの距離の巡回路で妥協してさっさと行動に移るでしょう. 問題を解くのにかかる時間も貴重な自分の時間だからです. 実際の応用においても, このように, 厳密な最適解でなくてもいいからよい解を速く求めたいという要求が多いのです. この目的を実現するための手法を近似解法と呼んでいます. その中の基本的なものとして欲張り法と局所探索法があります. それぞれ例を用いて紹介しましょう.

まず, ナップサック問題に対する欲張り法の例です. 商品の 1 円あたりの満足度を計算します. 表 1 の例では商品 1 は $14/3\simeq4.67$, 商品 2 は $21/5=4.2$, という具合いです. この値が大きいほど, 値段のわりに満足度が大きいので, お買い得ということです. この値の大きい順に商品を買っていき, 買うと予算を越えるものは飛ばして次にいきます. 表 1 の例では商品番号が 1 円あたりの満足度の順になっているので, 商品 1 から順に調べていきます. まず商品 1 を買うと満足度合計 14, 予算残高 $9-3=6$ 円, 次に商品 2 を買うと満足度合計 $14+21=35$, 予算残高 $6-5=1$ 円となります. 次に商品 3 を調べますが, 値段が予算残高の 1 円よりも高いので買えません. 商品 4 と 5 も同様です. 商品 6 は 1 円なので買います. 結局商品 1, 2, 6 を購入して予算をきっちり使い切り, 満足度合計 36 という解が得られます. 欲張り法は, このように簡単な規則に基づいて解を構築する方法です.

これに対し, 局所探索法は, 解に小さな修正を加える操作を反復する方法です. 巡回セールスマン問題に対する例を見てみましょう. 図 2 では, 交差している中央の 2 本の線をつなぎかえることで巡回路の距離が短くなっています. このような修正によって解を次々と改善していき, 改善ができなくなったところで終了するという方法です.

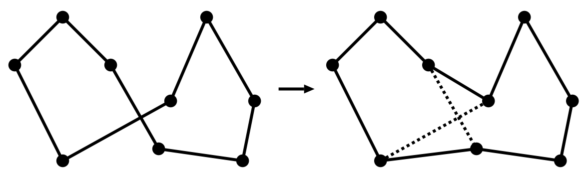


図 2. 巡回路の修正

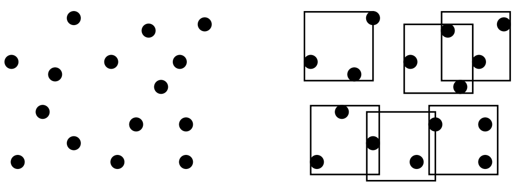
いずれも簡単な方法ですが, そんなに悪くなさそうな気がします. 実際, 通常は結構いい解が得られることが経験的に知られています. 最近では, これらを高度に発展させてよりよい解を求めようとするメタヒューリスティクス（あるいはメタ戦略, メタ解法）の研究が盛んです. 遺伝アルゴリズム, アニーリング法, タブー探索法などが含まれますが, これらは結構有名なので, 聞かれたことがあるかもしれません. 詳しくは [5] などをご覧下さい.

気になる近似精度

欲張り法や局所探索法による解は, 経験的にはよいと言いましたが, 理論的に精度を保証できないだろうかというのは気になるところです. たとえば, 最短の巡回路長 z^* と局所探索法によって得られる巡回路長 $\hat{z}$ に対して, 相対誤差 $(\hat{z}-z^*)/z^*$ は最悪の場合どの程度になるのでしょうか.

残念ながら, 上で紹介したナップサック問題に対する欲張り法や巡回セールスマン問題に対する局所探索法には, 相対誤差がいくらでも大きくなるような意地悪な例が存在することが知られています. 一方, これらふたつの問題に対しては, 相対誤差が十分小さくなることが理論的に保証できる方法もあるのですが, いずれも複雑なので, ここでは省略します. その代わり, 別の問題に対して, 簡単な論理で相対誤差 1 以下を保証できる方法を紹介します.

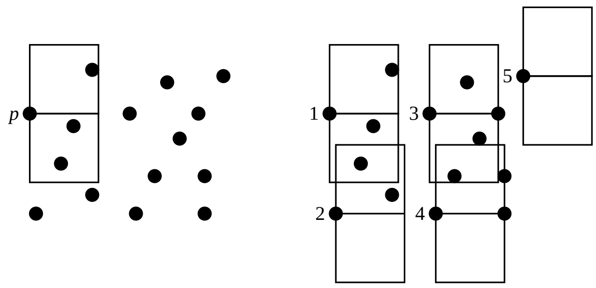
正方形カバー問題という問題を考えます. 平面上に n 個の点が与えられたとき, 各辺の長さが 1 の正方形を複数用いてすべての点をカバーします. このとき用いる正方形の個数を最小化するという問題です. なお, 正方形を平面に置くとき, 正方形内部あるいは辺上の点がすべてその正方形でカバーされます. 正方形は 1 辺が水平になるように置き, 回転は許されません. 図 3 に点数 $n=15$ の問題の例 (a) と, それらをカバーするために正方形を 6 個用いた例 (b) を示します.



(a) 点の配置　　(b) 正方形の配置例

図 3. 正方形カバー問題の例

これに対し以下の欲張り法を考えます. まず一番左の点を選び, これを p と呼びます. 2 枚の正方形を, 点 p が左下の角に重なるように 1 枚, 左上の角に重なるようにもう 1 枚を置きます (図 4-(a) 参照). 以上の操作を, カバーされずに残っている点に対して同様に繰り返し, すべての点がカバーされたら終了します. 図 4-(b) が得られた解です. 点 p として選ばれた点に, 選ばれた順に番号をつけています.



(a) 1ステップ目　　(b) 欲張り法による配置

図 4. 正方形カバー問題に対する欲張り法の実行例

この例では 5 個の点が p として選ばれていますが, この数を $\bar{z}$ としましょう. また, 欲張り法が用いた正方形の個数を $\hat{z}$, 最適解の正方形数を z^* と記します. まず, p として選ばれた点（図 4-(b) の例では点 1 から 5）のうちのふたつを 1 枚の正方形でカバーすることはできません. もしこれができてしまうと, 欲張り法の動作と矛盾するからです. このことから, p として選ばれた点の個数以上の正方形が必要であること, すなわち $\bar{z} \le z^*$ が成り立ちます. 一方, 欲張り法は p として選ばれた点のそれぞれに対して 2 枚の正方形を置いていくので, $\hat{z}=2\bar{z}$ です. 以上より $\hat{z} \le 2z^*$ となるわけですが, これを相対誤差で表すと, $(\hat{z}-z^*)/z^* \le 1$ となります.

さて, どんな点の配置に対しても欲張り法の解の相対誤差が 1 以下であることを示すことができました. しかし, 別の言い方をすると, 保証できたのは「相対誤差 100%以下」ですから, まだまだ大いに改善の余地がありそうに思えます. 実際, ここで紹介した欲張り法は, この理論的保証を得るために必要なルールを書き下しただけのものであって, 実際の性能を上げるには, もう一工夫して無駄を省く必要があります. たとえば図 4-(b) の解からは正方形を 3 枚取り除けます. しかし, 「相対誤差がいくらでも大きくなる意地悪な例がある」よりも, 「どんな意地悪な例を作っても相対誤差は 1 より大きくならない」ほうが安心感があります. 以上のような簡単な議論でこの程度の安心感を保証できるというのもなかなかおもしろいと思いませんか?

やはり最適解でなくては

実用上は近似解で十分であることが多いのですが,「やはり厳密な最適解でないと困る」ということもあります.

冒頭で紹介した最長しりとり問題は（実用性はさておき）近似解では満足できない一例です.「54,321 語のしりとりを見つけたよ」「もっと長いのないの?」「さあ, どうかな」ではすっきりしないでしょう. もちろん, この問題をきちんと議論するには, 対象となる単語集合を定めた上で,「バター」のように「ー」で終る単語などの扱いの約束を決める必要がありますが, これらを整理したうえで, この問題にまじめに取り組んだ人たちがいます [2]. それによると, ある国語辞典からすべての名詞を取り出した 192,687 語に対する最長しりとりの長さは 86,788 だそうです. これだけ多くの単語に対してすべての場合を列挙することはできません. 組合せ最適化における過去の成果をふまえていろいろな工夫を施すことで, これほどの規模の問題に対する厳密な最適解を求めることができるのです. 数式をなるべく使わずに分かりやすく解き方を解説した資料が公開されていますので, 是非ともご覧下さい.[2]

もうひとつおもしろい例を紹介しましょう. JR の片道切符を 1 枚だけ使って, 日本全国できるだけ長い距離を旅行するには, どのようなルートの片道切符を購入したらよいだろうか, という問題で, そのような切符は最長片道切符と呼ばれています. 鉄道に興味のない人には, まあ, どうでもいい問題ですが, 鉄道好きの人々の間ではかなり前からホットな話題だったようです. これもまた（実用性ははなはだ疑問ですが）近似解では満足できない一例です. この場合はとくに, 実際に旅行しようとすると, 旅費が 10 万円弱, 日程が 20 日程度と, 金銭的にも時間的にもかなりの負担になりますが, 鉄道好きとしては「これが最長片道切符に違いない」と信じて旅行したのに後日「もうちょっとだけ長いルートがあったよ」と言われたら, これほど悔しいことはないでしょう. それを避けるためにも厳密な最適解が必要です. こちらも, 組合せ最適化の手法を駆使して最適解をまじめに計算し, しかもそのルートを実際に旅行した人がいます. 図 5 に宮代と葛西 [3] が計算した最長片道切符全経路（2002 年 12 月現在）を示します.[3] なお, [3] には, 著者の一人が実際に最長片道切符を（学割で）購入し, そのルートに沿って旅行した体験談も記されています.

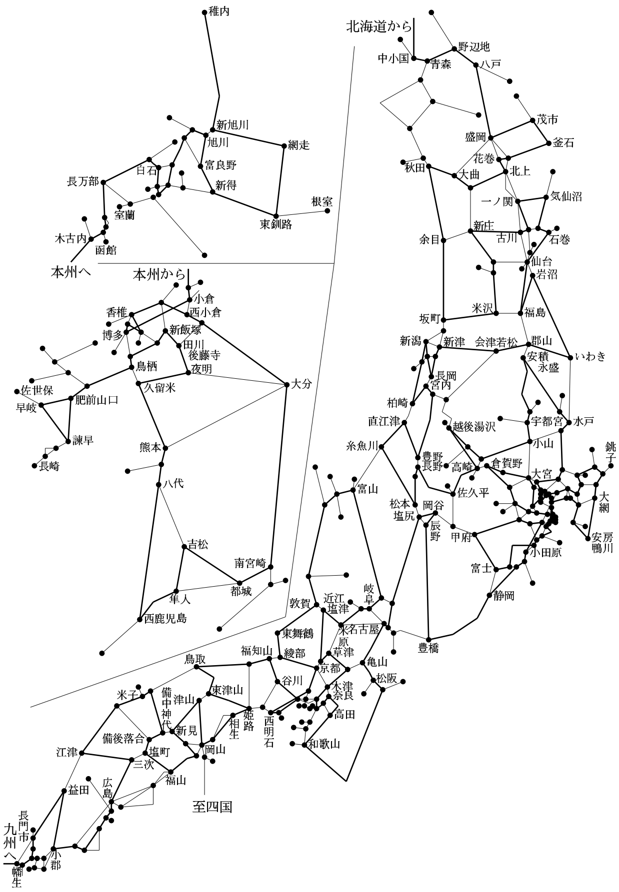


図 5. 最長片道切符全経路（宮代・葛西 [3] より）

例が少々マニアックすぎたかもしれませんが, 厳密な最適解が望まれるこのような状況に現実的に対処できるよう, 厳密解法の研究も古くから行われています. たとえば巡回セールスマン問題に対しては, 目的地数 n の大きな問題に対して厳密な最適解を求める競争が世界レベルで行われていて, 現在の世界記録は $n = 15{,}112$ です.[4] 図 6 にその最適巡回路を示します. 列挙法のような単純な方法では数十の目的地が限界であるのに対し, 1 万を越えるサイズの問題に対して最適解を計算できるというのはすごいと思いませんか?

これまで紹介した問題の厳密解はいずれも分枝

[2] http://al.cs.tuat.ac.jp/~yshinano/shiritori/

[3] http://www.swa.gr.jp/lop/にも詳しい情報がある.

[4] http://www.keck.caam.rice.edu/tsp/などを参照.

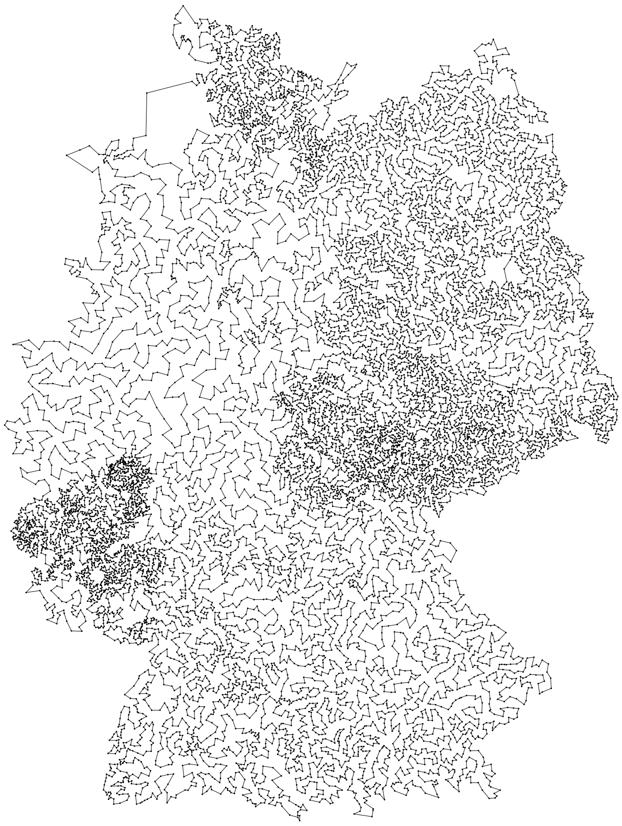


図 6. ドイツの 15,112 都市を巡回する最適巡回路

限定法という手法に基づいて得られたものですが, この方法は説明が少々大変なので, もう少し手短に説明できる動的計画法という厳密解法を, ナップサック問題の表 1 の例を用いて紹介しましょう. 計算の過程は表 2 のようになります (下部に表 1 の内容を再掲します). 表 2 の第 i 行 j 列は, 予算上限が j 円であるときに商品 $1, 2, \ldots, i$ の中からいくつかを組合せて得られる満足度の最大値です. まず, 第 1 行は商品 1 しか使えないわけですから簡単です. 第 0 列から 2 列までは, 予算上限が商品 1 の値段 3 よりも小さいので買うことはできず, 0 となります. 一方, 第 3 列以降は, 予算上限が 3 以上となるので商品 1 を買うことができ, 商品 1 の満足度 14 が第 3 列から 9 列まで入ります. 第 2 行は

A. 第 1 行の同じ列の値

B. 第 1 行の 5 列左の値（そのような列がないときは $-\infty$）に 21 を加えた値

のふたつのうちの大きい方を表に書きます. ルール B の 5 は商品 2 の値段, 21 は満足度です. たとえば第 2 行 8 列では, ルール A による値は真上の 14, ルール B による値は第 1 行 3 列の値 14 に 21 を加えた 35 であり, これらの大きい方を取って 35 を書きます. ルール A は商品 2 を買わない場合, B は買う場合に対応します. 第 3 行以降はひとつ上の行を基準にして同様に計算し, 最終的に第 6 行 9 列が求める最大満足度になります. 対応する商品の組合せは, 第 6 行 9 列から始めて, 表の値がルール A と B のどちらで実現されたかをたどることで得られます. たとえば第 6 行 9 列の 37 はルール A によって第 5 行 9 列の値がそのまま利用されています. これは商品 6 を買わないことに対応します. 第 5 行 9 列も同様です. 第 4 行 9 列は, ルール B によって, 第 3 行 3 列の値 14 に商品 4 の満足度 23 を加えることで値 37 が実現されています. すなわち, 商品 4 を買うことに対応します. 表 2 では, このようにたどった結果を太字で示しています. 結局, この例では, 商品 1 と 4 を購入して, 予算をきっちり使い切り, 満足度 37 を達成する解が最適となります. 欲張り法による解では満足度が 36 でしたので, 欲張り法では必ずしも最適解が得られるとはかぎらないこと, とはいってもそれほど悪い解ではないことが確認できました.

表 2. ナップサック問題に対する動的計画法の進行

		予算上限									
		0	1	2	3	4	5	6	7	8	9
商品	1	0	0	0	**14**	14	14	14	14	14	14
	2	0	0	0	**14**	14	21	21	21	35	35
	3	0	0	8	**14**	14	22	22	29	35	35
	4	0	0	8	14	14	22	23	29	35	**37**
	5	0	0	8	14	14	22	23	29	35	**37**
	6	0	1	8	14	15	22	23	29	35	**37**

商品番号	1	2	3	4	5	6
満足度	14	21	8	23	11	1
値段	3	5	2	6	4	1
予算上限			9			

こんな計算で最適解が計算できるのはなんだか不思議な気がするかもしれませんが, 表の意味をじっくり考えると, 正しいことが理解できます. （理解できなくても悲観しないでください.）この例では, すべての可能性を列挙してもたかだか $2^6 = 64$ 通りですので, どうしても信じられない人はチェックしてみるとよいでしょう.

動的計画法による表の計算時間は, 表の大きさ, すなわち $n \times$（予算上限）ですが, これは残念なが

ら多項式時間ではありません.[5] しかし, 予算上限は通常それほど大きな数値にはならないので, 商品数 n が大きいときには, 単純な列挙法が調べるべき場合の数 2^n よりも動的計画法の表のサイズのほうがはるかに小さく, 有利になることが多いのです.

身近な問題—多項式時間で解ける例

身近な例を題材に組合せ最適化のアプローチをいくつか紹介しましたが, いずれも NP 困難問題に対するお話でした. この他にもさまざまな組合せ最適化問題が身近なところで解かれているのですが, その中には多項式時間で解けるものもあります. その一例が, カーナビのルート探索の時に必要となる「現在地から目的地までの最短のルートを求めよ」という最短路問題です. これに対してはダイクストラ法という効率的な方法があり, 多くの教科書で紹介されています (たとえば [1]).

もう一つ目先の変わった例を紹介しましょう. 英語の教科書や英字新聞などを見ると, たいてい右端が縦にきれいにそろっています. 日本語は基本的に各文字の幅が同じですから, ページの右端（縦書きなら下）をそろえるのはたやすいのですが, 単語の長さがまちまちの英語では, 改行位置や単語間のスペースなどを上手に決めないと, 単語をハイフンで切った改行や大きなスペースが多くなり, 見栄えが悪く読みづらい印刷になってしまいます (図 7 参照). じつはこの問題も組合せ最適化問題として定式化でき, 多項式時間の効率よい解法が存在します.

LAFAYETTE — The Boulder
County Health Department's La- ← ハイフンが入った
fayette office is reducing serv- ←
ices and hours because of recent
state budget cuts to public ← スペースが広すぎる
health services.

図 7. 英文印刷のレイアウト

以上の二つの問題に関しては, 教育用のプログラムや講義資料の一部を私のホームページに置いています.[6] 興味があれば使ってみてください.

[5] 擬多項式時間と呼ばれています.
[6] http://www-or.amp.i.kyoto-u.ac.jp/~yagiura/kyouiku-program.html

むすび

組合せ最適化に興味を持ってここまで読んで下さった方は, [4] もご覧いただければ幸いです. 本稿の内容の一部をもう少し詳しく解説しています. 組合せ最適化は他の分野に比べると比較的歴史が浅く, 過去の研究成果にそれほど精通していなくても, 若い柔軟な発想で大きな成果を産み出せる可能性が十分にある分野です. この世界で研究を始めてみませんか?

謝辞 最後に, 宮代隆平氏には最長片道切符の全経路の図をいただきました. また, 品野勇治, 梅谷俊治, 今堀慎治の諸氏には有益なコメントをいただきました. ここに記して謝意を表します.

参考文献

[1] 茨木俊秀, C によるアルゴリズムとデータ構造, 昭晃堂, 1999.

[2] 乾伸雄, 品野勇治, 鴻池祐輔, 小谷善行, 最長しりとり問題の解法, 情報処理学会論文誌「数理モデル化と応用」, 掲載予定.

[3] 宮代隆平, 葛西隆也, 最長片道切符, オペレーションズ・リサーチ, Vol. 49, No. 1, pp. 15–20, 2004.

[4] 柳浦睦憲, “組合せ最適化の数理: 計算困難問題への挑戦,” 数理科学, 第 40 巻 12 号, No. 474, pp. 21–27, 2002.

[5] 柳浦睦憲, 茨木俊秀, 組合せ最適化 — メタ戦略を中心として, 朝倉書店, 2001.

（やぎうら むつのり）