

13. 近似アルゴリズム

1

13.1 近似アルゴリズムの種類

NP困難な問題に対しては多項式時間で最適解を求めることは困難であるので、最適解に近い近似解を求めるアルゴリズムが用いられることがある。

このように、必ずしも厳密解を求めないアルゴリズムは、大きく分けて2つの範疇に分けられる。

2

ヒューリスティックと近似アルゴリズム

ヒューリスティクス
(発見的解法、経験的解法)

- 遺伝的アルゴリズム (GA)
- アニーリング (焼きなまし法)
- タブサーチ
- ローカルサーチ
- ニューラルネット等

精度保証無し、実験的評価が主

近似アルゴリズム

- 定数近似アルゴリズム (APX)
- 多項式近似スキーム (PTAS)
- 完全多項式近似スキーム (FPTAS)

精度保証付き理論的解析が主

ここでは、近似アルゴリズムについてみていく。

3

α 近似アルゴリズム

最小化問題に対して、いつも最適解の α 倍以下の解を入力サイズの多項式時間で求めるようなアルゴリズムを α 近似アルゴリズムという。ここで、 $\alpha \geq 1$ であり α を近似率という。すなわち、最適解を f_{OPT} と表し、アルゴリズムの解の評価値を f_{AL} と表すと、常に次の式を満足する。

$$\frac{f_{AL}(\mathbf{x})}{f_{OPT}(\mathbf{x})} \leq \alpha$$

$$\therefore f_{AL}(\mathbf{x}) \leq \alpha f_{OPT}(\mathbf{x})$$

4

β 近似アルゴリズム

最大化問題に対して、いつも最適解の β 倍以上の解を入力サイズの多項式時間で求めるようなアルゴリズムを β 近似アルゴリズムという。ここで、 $\beta \leq 1$ であり β を近似率という。また、 β を近似保証ということもある。すなわち、最適解を f_{OPT} と表し、アルゴリズムの解の評価値を f_{AL} と表すと、常に次の式を満足する。

$$\frac{f_{AL}(\mathbf{x})}{f_{OPT}(\mathbf{x})} \geq \beta$$

$$\therefore f_{AL}(\mathbf{x}) \geq \beta f_{OPT}(\mathbf{x})$$

5

APX

α (あるいは β) 定数であるようなアルゴリズムが存在するクラスをAPX (APproXimation) という。

6

PTASとFPTAS

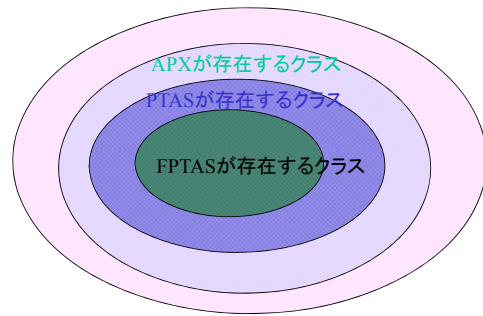
近似率 α (β)を限りなく1に近づけることができるとき、そのような α 近似アルゴリズムをPTAS(Polynomial Time Approximation Scheme, 多項式時間近似スキーム)という。すなわち、任意の正数 $\epsilon > 0$ に対して、とできる入力サイズの多項式時間アルゴリズムをPTASという。さらに、入力サイズと精度の多項式時間アルゴリズムFPTAS(Fully Polynomial Time Approximation Scheme, 完全多項式時間近似スキーム)という。

問題によっては、PTASが存在しない(知られていない)ものがある。また、定数近似すら存在しない問題もある。このようなアルゴリズムの出力は入力サイズに依存した近似値になってしまう。

7

近似アルゴリズムの存在

近似アルゴリズムが存在するクラス

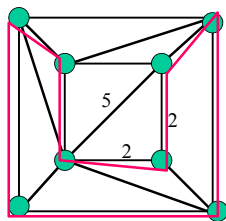


8

13-2. 巡回セールスマン問題

巡回セールスマン問題には、ネットワーク型と、幾何型とがある。

ネットワーク型の巡回セールスマン問題では、入力はい重み付きのグラフであり、出力は頂点を全て巡る閉路で重みの総和が最小のものである。



9

三角不等式を満足しないようなネットワーク型の巡回セールスマン問題は、近似アルゴリズムを得ることすらNP-困難である。

具体的には、定数近似アルゴリズムが存在すると、NP完全問題であるハミルトン閉路問題が多項式時間で解けてしまう。つまり、ハミルトン閉路問題をネットワーク型の定数近似巡回セールスマン問題に帰着できてしまう。このことより、ネットワーク型の巡回セールスマン問題には定数近似アルゴリズムは存在しないと考えられている。

10

ハミルトン閉路問題と巡回セールスマン問題

名称: ハミルトン閉路問題
 インスタンス: グラフG
 問い: G中に、全ての頂点を丁度1度だけ通るような閉路は存在するか?

名称: 巡回セールスマン(NTSP)
 インスタンス: ネットワークN, 正数K
 問い: ネットワーク中の全ての頂点を通るような閉路で重みの総和がK以下となるようなもの存在するか?

11

巡回セールスマン問題への帰着

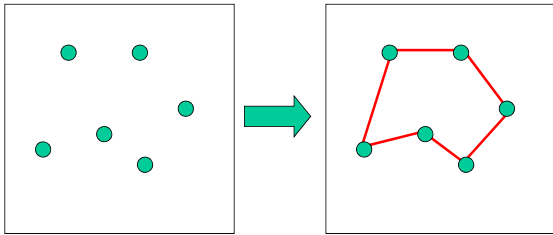
ネットワーク型の巡回セールスマン問題を解く α 近似アルゴリズムが存在すると、次のようにハミルトン閉路問題を巡回セールスマン問題に帰着できる。つまり、ハミルトン閉路問題のインスタンスであるグラフGから巡回セールスマン問題のインスタンスである辺重み付きグラフ(ネットワークN)と整数Kを定めればめることができる。

まず、Gの辺に全て1の重みをつけてネットワークNを構成する。次に、Kとして、 $K = \frac{n}{\alpha}$ として、巡回セールスマン問題へ帰着する。このとき、 α の定数近似であるAPXが存在すると、多項式時間でハミルトン閉路の存在判定ができてしまう。

以上のことより、 $P \neq NP$ である限り、ネットワーク型の巡回セールスマン問題を解く多項式時間アルゴリズムはない。

12

幾何巡回セールスマン問題(GTSP)



実は、幾何巡回セールスマン問題には、定数近似アルゴリズムが存在する。ネットワーク型と、幾何型での最大のちがいは、三角不等式が成り立つかどうかである。ここで三角不等式とは、任意の x, y, z に対して、次が成り立つことである。

$$d(x, y) \leq d(x, z) + d(y, z)$$

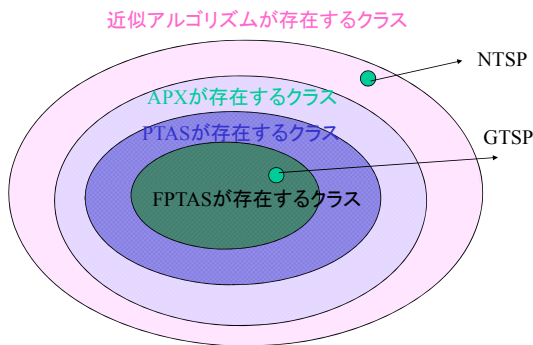
13

2次元(ユークリッド)平面上の点集合を考えれば、2点間の距離は自動的に三角不等式を満たす。

このことは、利用できる情報(三角不等式)が多くなったということであり、ネットワーク型よりも幾何型の方が問題が簡単であることを意味する。実際、幾何型の巡回セールスマン問題は、FPTASを持つことが最近(1998年)に示された。このアルゴリズムは複雑なので、ここでは2近似アルゴリズムとそれを改善した1.5近似アルゴリズムを示す。

14

NTSPとGTSP



15

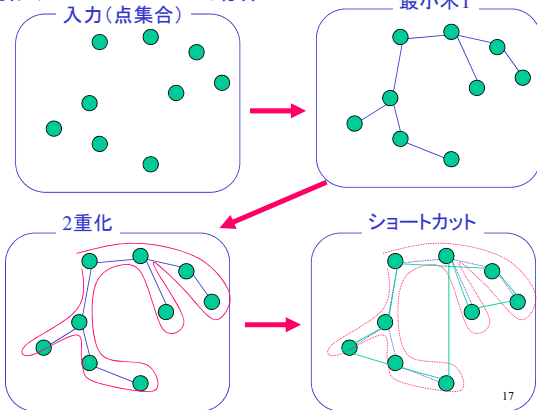
2近似アルゴリズム

- GTSPに対する2近似アルゴリズム
1. 点集合を連結する最小全域木 MST T を求める。
 2. T の辺を辿りながら、全ての点を通る巡回路 C を求める。(T の辺を全て2重化すればいい。)
 3. C で一度通過した点をショートカットする順回路 C' を求める。

次にこのアルゴリズムの動作を示す。

16

2近似アルゴリズムの動作



17

近似率の解析

最適な順回路を C_{OPT} とし、アルゴリズムで得られる順回路を C_{AL} とする。また、 $|C_{OPT}|, |C_{AL}|$ でそれぞれの長さを表すものとする。

順回路 C_{OPT} から任意に一本辺を除去すると、全域木が得られる。よって、最小全域木 T の方が辺の重み総和は小さい。(そもそも、頂点を連結するものなので、重みが最小なものが最小全域木であった。)

よって、次が成り立つ。

$$|T| \leq |C_{OPT}|$$

18

また、アルゴリズムで得られる閉路では、最小木を2重化したものより小さいので、次が成り立つ。

$$|C_{AL}| \leq 2|T|$$

$$\therefore \frac{|C_{AL}|}{2} \leq |T|$$

以上より、

$$\therefore \frac{|C_{AL}|}{2} \leq |T| \leq |C_{OPT}|$$

$$\therefore |C_{AL}| \leq 2|C_{OPT}|$$

$$\therefore \frac{|C_{AL}|}{|C_{OPT}|} \leq 2$$

19

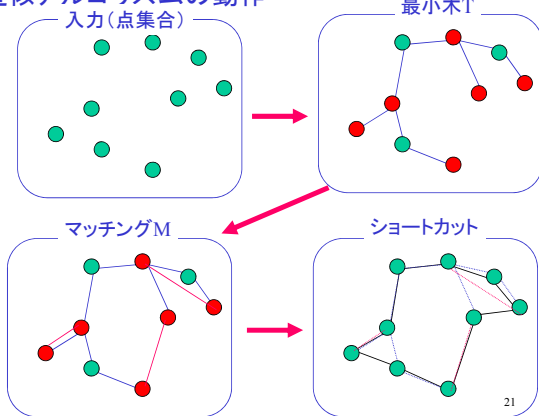
1.5 近似アルゴリズム

GTSPに対する1.5近似アルゴリズム

1. 点集合を連結する最小全域木MST T を求める。
2. T において、次数が奇数の点からなる完全グラフを作り、最小重みマッチング M を求める。
3. $T+M$ はオイラーグラフであるので、一筆書きに対応する順回路 C を求める。
4. 3の順回路 C からできるだけショートカットとして、順回路 C' を構成する。

20

2 近似アルゴリズムの動作



21

近似率の解析

最適な順回路を C_{OPT} とし、アルゴリズムで得られる順回路を C_{AL} とする。また、 $|C_{OPT}|, |C_{AL}|$ でそれぞれの長さを表すものとする。

2近似アルゴリズムの解析と同様にして、

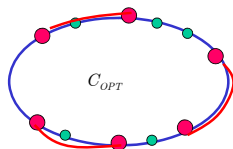
$$|T| \leq |C_{OPT}|$$

を得る。

また、 C_{OPT} 上で、次数が奇数の点(奇点)を結ぶパスを考える。 C_{OPT} 上で、奇点を結ぶパスを交互に選ぶことにより、 C_{OPT} 上の辺集合を2つに分類する。

22

奇点が偶数個しかないことに注意すると、いつもきちんと2分割することができる。わかる。

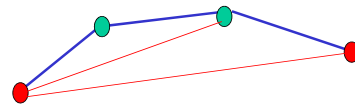


このように、 C_{OPT} の辺集合を、 P_1, P_2 と分割できる。もちろん、 $C_{OPT} = P_1 \cup P_2$

よって、 $|C_{OPT}| = |P_1| + |P_2|$

23

また、三角不等式がなりたっているので、パスで結ぶより直接辺でたどったほうが短い。



よって、最小マッチング M の重み総和に関して、次が成り立つ。

$$|M| \leq \min\{|P_1|, |P_2|\}$$

$$\therefore |C_{OPT}| = |P_1| + |P_2|$$

$$\geq 2 \min\{|P_1|, |P_2|\}$$

$$\geq 2|M|$$

24

一方、アルゴリズムより、

$$|C_{AL}| \leq |T| + |M|$$

以上より、

$$\begin{aligned} |C_{AL}| &\leq |T| + |M| \\ &\leq |C_{opt}| + \frac{1}{2}|C_{opt}| \\ &= \frac{3}{2}|C_{opt}| \\ \therefore \frac{|C_{AL}|}{|C_{opt}|} &\leq 1.5 \end{aligned}$$

25

このように、いろいろな技法を組み合わせると、近似率の改善が行われる。

NP完全問題に対しても、厳密解でなくてもよければ、近似アルゴリズムの適用を考えてみると良い。

26

13-3. ナップザック問題

ナップザック問題の一般形は次のよう書ける。

ナップザック

P 特徴ベクトル $\mathbf{x} = (x_1, x_2, \dots, x_n)$

最大化 $f(\mathbf{x}) = \sum_{i=1}^n v_i x_i$

条件

$$\sum_{i=1}^n s_i x_i \leq b$$

$$x_i \in \{0,1\}$$

27

ナップザック問題における欲張り法 (グリーディ法、Greedy法)

連続ナップザック問題のように、単位価値の高い法から順に選んでいく方法を考察する。このように、部分的に評価関数を改善するだけの方法を欲張り法 (Greedy法) という。(欲張り法でも近似アルゴリズムになっていることもある。これらは、問題やアルゴリズムをきちんと解析しないとわからない。)

28

ナップザックに対する欲張り法

1. 単位価値の高い順にならべる。すなわち、必要なら添え字を付け換えて、

$$\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \dots \geq \frac{v_n}{s_n}$$

とする。

2. $i = 1$ から n まで順番に

$$s_i \leq b - \sum_{j=1}^{i-1} s_j x_j \quad \text{なら} \quad x_i = 1 \quad \text{とし、}$$

そうでないなら

$$x_i = 0$$

とする。

29

欲張り法の性能

欲張り法で得られる解を $x^g = (x_1^g, x_2^g, \dots, x_n^g)$ とおき、最適解を $x^o = (x_1^o, x_2^o, \dots, x_n^o)$ とおく。

$x^g = x^o$ なら、欲張り法と最適解が一致している。

以下では、 $x^g \neq x^o$ のときを考えよう。

このときは、最適解には採用されたが、欲張り解には採用されなかった最初の要素を考えてその添え字を j とする。すなわち、

$$0 = x_j^g < x_j^o = 1$$

このとき、任意の $i \leq j-1$ に対して、

$$x_j^g \geq x_i^g$$

である。

30

よって、

$$f(\mathbf{x}^g) = \sum_{i=1}^n v_i x_i^g \geq \sum_{i=1}^j v_i x_i^g = \sum_{i=1}^j v_i x_i^0 + \sum_{i=1}^j v_i (x_i^g - x_i^0)$$

$$\geq \sum_{i=1}^j v_i x_i^0 + \sum_{i=1}^j \frac{v_j}{s_j} s_i (x_i^g - x_i^0) = \sum_{i=1}^j v_i x_i^0 + \frac{v_j}{s_j} \left(\sum_{i=1}^j s_i x_i^g - \sum_{i=1}^j s_i x_i^0 \right)$$

という式が成り立つ。ここで、次のようなことに注意する。

- 任意の $i \geq j+1$ に対して、 $\frac{v_i}{s_i} \leq \frac{v_j}{s_j}$ である。
- 欲張り法で j が選ばれなかったことから、

$$\sum_{i=1}^j s_i x_i^g = \sum_{i=1}^{j-1} s_i x_i^g > b - s_j$$
- 最適解でも制約式を満たすので、

$$\sum_{i=1}^n s_i x_i^0 \leq b$$

$$\therefore \sum_{i=1}^j s_i x_i^0 \leq b - \sum_{i=j+1}^n s_i x_i^0$$

31

以上より、次の関係式が導ける。

$$\frac{v_j}{s_j} \left(\sum_{i=1}^j s_i x_i^g - \sum_{i=1}^j s_i x_i^0 \right) \geq \frac{v_j}{s_j} \left((b - s_j) - \left(b - \sum_{i=j+1}^n s_i x_i^0 \right) \right)$$

$$\geq \frac{v_j}{s_j} \left(\sum_{i=j+1}^n s_i x_i^0 - s_j \right) \geq \sum_{i=j+1}^n v_i x_i^0 - v_j$$

$$\therefore f(\mathbf{x}^g) \geq \sum_{i=1}^j v_i x_i^0 + \frac{v_j}{s_j} \left(\sum_{i=j+1}^n s_i x_i^0 - s_j \right)$$

$$\geq \sum_{i=1}^n v_i x_i^0 - v_j \geq f(\mathbf{x}^0) - v_j \geq f(\mathbf{x}^0) - v_{\max}$$

なお、ここで、 v_{\max} は価値 v_i の最大値である。

32

欲張り法で悪い評価値を出すインスタンス

$s_1=1, v_1=2$ $s_2=1, v_2=2$ $s_3=1, v_3=2$
 $a_4: s_4=10, v_4=10$

$B = 10$
 $\mathbf{x}^g = (1, 1, 1, 0)$ $f(\mathbf{x}^g) = 6$
 $S = (1, 1, 1, 10)$ $\mathbf{x}^0 = (0, 0, 0, 1)$
 $V = (2, 2, 2, 10)$ $f(\mathbf{x}^0) = 10$
 $b = 10$

33

ナップザックの1/2近似アルゴリズム

ナップザックに対する1/2近似アルゴリズム

1. 欲張り法によって解 \mathbf{x}^g を求める。
2. 価値が最大のもを一つだけ選ぶ。
3. 上の2つのうちで大きい方を解 \mathbf{x}^a として出力する。

34

近似率

アルゴリズムの出力(解) \mathbf{x}^a とする。このとき、以下の式が成り立つ。

$$f(\mathbf{x}^a) \geq \frac{f(\mathbf{x}^g) + v_{\max}}{2} \geq \frac{1}{2} f(\mathbf{x}^0)$$

以上より、1/2近似アルゴリズムであることがわかる。

35

ナップザック問題に対する FPTAS

ナップザック問題に対しては、任意の正数 ϵ に対する近似保証 $(1 - \epsilon)$ のアルゴリズムが知られている。

ナップザックに対する FPTAS

1. 与えられた ϵ に対して、 $K = \frac{\epsilon v_{\max}}{n}$ とおく。
2. 各要素 i に対して、価値を $v_i' = \lfloor \frac{v_i}{K} \rfloor$ と修正する。
3. 修正した価値のもとで、ナップザックの最適解 \mathbf{x}^r を動的計画法で求める。
4. 上記の解 \mathbf{x}^r と最初の価値のもとでの最大値を比べて評価値の高いものをアルゴリズムの出力(解) \mathbf{x}^a とする。

36

FPTASの評価

$$v_i' = \left\lfloor \frac{v_i}{K} \right\rfloor \text{ とおていることにより, } 0 \leq v_i - K v_i' < K$$

よって、任意の解 x に対して、その修正後の評価値 $f'(x)$ に関して次式が成立する。

$$0 \leq f(x) - K f'(x) < nK$$

一方、 x^r は修正した値での最適解なので

$$f'(x^r) \geq f'(x^o)$$

また、

$$f(x^a) \geq v_{\max}$$

37

よって、

$$\begin{aligned} f(x^a) &\geq f(x^r) \geq K f'(x^r) \geq K f'(x^o) \\ &\geq f(x^o) - nK = f(x^o) - \varepsilon v_{\max} \geq f(x^o) - \varepsilon f(x^a) \end{aligned}$$

以上より、

$$\begin{aligned} (1 + \varepsilon) f(x^a) &\geq f(x^o) \\ \therefore f(x^a) &\geq \frac{1}{1 + \varepsilon} f(x^o) > (1 - \varepsilon) f(x^o) \end{aligned}$$

38

計算時間

ステップ3の動的計画法の部分について考察しよう。

まず、動的計画法に基づくナップザックアルゴリズムとして、大きさが決まっているときの価値の最大値を表として構成していた。この動的計画法に基づいた場合、 $O(nb)$ 時間のアルゴリズムが得られた。

ここでは、この動的計画法を以下の方針に切り替える。価値が決まっているときに、大きさの最小値として構成する。このような動的計画法も構成できることに注意する。この場合、価値の最大値は、 v_{\max} であるので、評価値の最大値は、 εv_{\max} である。よって、アルゴリズムの計算量は、 $O(n \varepsilon v_{\max})$

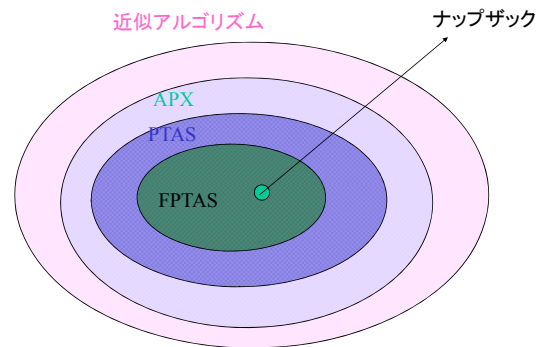
$O(n^2 v_{\max})$ 時間である。

FPTASでは修正した値を用いるので、結局、

$$O\left(n^2 \left\lfloor \frac{v_{\max}}{K} \right\rfloor\right) = O\left(n^2 \left\lfloor \frac{n}{\varepsilon} \right\rfloor\right) = O\left(\frac{1}{\varepsilon} n^3\right)$$

39

ナップザック問題の近似可能性



40