アルゴリズム 77－1
（2001．3．12）

白色雑音を用いる勾配法とそのTSPへの応用

日本IBM 東京基礎研究所　　筑波大学社会工学系
岡野 裕之　　香田 正人

白色雑音を用いて目的関数の微分を求める勾配法SNRを提案する．この方法では目的関数の微分が利用できる必要がなく，微分不可能であったりプログラムの手続きとして定義されたものでもよい．SNRの離散最適化への応用性を調べるために，巡回セールスマン問題を取り上げ，Hopfield-Tank型, Held-Karp型の形式化に適用する．さらに，追加法に基づく形式化を提案する．数値実験の結果，SNRがこれらの形式化に適用できることが示される．

A Gradient Method Using Gaussian White Noise and Its Application to the TSP

Hiroyuki Okano
IBM Japan, Tokyo Research Laboratory
okanoh@jp.ibm.com

Masato Koda
Institute of Policy and Planning Sciences,
University of Tsukuba
koda@shako.sk.tsukuba.ac.jp

A new gradient method, which computes derivatives of the objective functions by injecting Gaussian white noise to each variable, is proposed. In the proposed method, referred to as stochastic noise reaction (SNR), derivatives of the objective functions need not be available, and they may even be non-differentiable or arbitrary subroutines in a computer program. A Hopfield-Tank-type, a Held-Karp-type, and a novel addition heuristic-based formulations for the traveling salesman problem are introduced to discuss SNR's applicability to discrete problems. Through numerical study, SNR is shown to be applicable to these formulations.

1　はじめに

関数の最小化には，目的関数が微分可能なら，例えば勾配法やニュートン法のように，微分を利用する方法が一般的に用いられる．ところが，関数が複雑だったり，変数値が離散的だったり，そもそもプログラムの手続きとして定義されていると微分を利用できない．一方経験的には，微分の定義されない関数でも，変数の微小な変分に対する関数値の変分は小さく，目的関数の地形は(確率的に)滑かである場合が多い．そういった経験則から，離散問題では局所探索がよく用いられる．これに対し筆者らは，確率感度解析に基づいて微分を近似することで，離散問題にも適用可能な勾配法[1]や，誤差伝搬を不要とするニューラルネット学習方式[2]などを提案している．本稿では2章で，確率感度解析に基づく勾配法(Stochastic Noise Reaction: SNR)の算法を述べ，3章以降でこの方法を使った離散問題の近似解法を議論する．

離散問題の例として，巡回セールスマン問題(TSP)を取り上げる．TSPは，頂点集合 $V = \{1, 2, \ldots, |V|\}$ と，頂点間の距離 d_{ij} が与えられ，

$$h(\pi) = \sum_{i=1}^{|V|} d_{\pi(i),\pi(i+1)}, \qquad (1)$$

と定義される目的関数を最小化する順列 π を求める問題である．ここで，$\pi(|V|)$ が $\pi(|V|-1)$ と $\pi(1)$ に隣接するように，π は modulo $|V|$ で定義されているとし，d_{ij} には平面上の距離を仮定する．3章では，知られている2つのTSPの形式化にSNRを適用する．一つはHopfield-Tank型の形式化で，Hopfield と Tank [3] が提案したハードウェアとして実現できるニューラルネットのエネルギー関数を，Brandt [4] らが数値実験でうまく

SNR 算法の枠組み

1. 現状解 x を初期化する． $x := x^0$.
2. 最良解 x^{best} を初期化する． $x^{best} := x$.
3. **For** $k := 1, 2, \ldots, N$ **do begin** // 反復回数 N のループ.
4. $\mu_k := \mu_{max}(1 - 0.9(k-1)/(N-1))$. // $\mu_k \in [\mu_{max} \to 0.1\mu_{max}]$.
5. 降下方向 δx を初期化する． $\delta x := 0$.
6. **For** $j := 1, 2, \ldots, M$ **do begin** // 勾配を近似するループ.
7. 雑音ベクタ ξ^j を生成する. // $\xi_i^j \in N(0,1), i = 1, 2, \ldots, n$.
8. $\delta x_i := \delta x_i - f(x^j)\xi_i^j$. // $E[f(x^*)\xi_i] = E[\frac{\partial f(x^*)}{\partial x_i}]$.
9. **If** $f(x^{best}) > f(x^j)$ **then** $x^{best} := x^j$. // 最良解を保存する.
10. **end;**
11. $x := x + \mu_k \delta x/|\delta x|$. // 現状解を更新する.
12. $x_i := \max\{\min\{x_i, x_{max}\}, x_{min}\}$. // 各変数を許容領域に合わせる.
13. **If** $f(x^{best}) > f(x)$ **then** $x^{best} := x$. // 最良解を保存する.
14. **If** 終了条件が満たされた **then goto** 16.
15. **end;**
16. 最良解 x^{best} を出力する.

図 1: 白色雑音を用いる勾配法の枠組み

ゆくように改良したものに，さらに変更を加えたものである．この関数は微分可能なので，微分を用いる勾配法の結果と比較することで，確率感度解析に基づく勾配法の性能を議論する．もう一つは Held-Karp 型の形式化で，Held と Karp [5, 6] が提案した TSP の近似解法に Christofides 法 [7] の一部を併用して，許容ツアーが必ず得られるようにしたものである．この形式化には，Held と Karp が劣勾配法を与えているので，その方法と確率感度解析に基づく勾配法を比較する．さらに，TSP の構築法として知られる追加法を用いる形式化を提案し，確率感度解析を適用する．この形式化に関しては比較対象がないが，勾配法が適用できるかどうかを議論する．

2 確率感度解析に基づく勾配法

2.1 勾配法

次のような最小化問題を考える: Minimize $f(x)$ subject to $x \in S$. ここで，解 x は n 次元の列ベクタで，許容領域 S は $\mathbb{R}^n$ 内の集合である．勾配法は次のように，現在の解 x の値を徐々に更新し，目的関数 $f(x)$ を局所最小化する:

$$x \leftarrow x + \mu \frac{\delta x}{|\delta x|}, \quad \delta x = -\nabla f(x), \tag{2}$$

ここで，μ はステップ幅，δx は降下方向，$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^T$ は勾配．この勾配法は，本稿では図 1 の枠組みで実行される．ただし，ステップ 5 から 10 は省略され，降下方向 δx が $-\nabla f(x)$ で計算される．劣勾配法では降下方向を問題固有のヒューリスティックで計算する．これらの算法を DA (Deterministic Algorithm) と呼ぶ．

2.2 確率雑音反応法 (SNR)

DA では偏微分や問題固有のヒューリスティックで勾配を計算するが，SNR では各変数 x_i に平均 0, 分散 1 のガウス白色雑音 $\xi_i \in N(0,1)$ を印加し，

$$E\left[\frac{\partial f(x)}{\partial x_i}\right] = \frac{1}{M}\sum_{j=1}^{M} f(x^j)\xi_i^j, \tag{3}$$

によって勾配を近似する．ここで，$E[\cdot]$ は平均を表し，M は平均をとるためのループ回数，x_i^j は確率変数 $x_i^* = x_i + \xi_i$ のある実現値 ($x_i^j = x_i + \xi_i^j$)，ξ_i^j は i 番目の変数に印加した ξ_i の j 番目の雑音．式 (3) の導出には，ノビコフの定理 $E[\delta H(\xi)/\delta \xi_i] =$

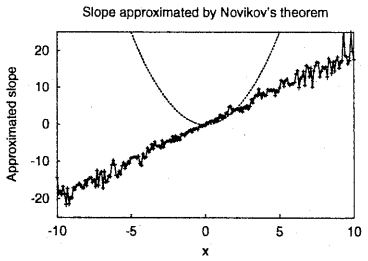

図 2: ノビコフの定理による x^2 の傾きの近似

$E[H(\xi)\xi_i]$ を用いている．ここで，$H(\xi)$ はガウス白色雑音の任意の関数で，$\delta H(\xi)/\delta\xi_i$ は汎関数微分．勾配の n 個の成分が，(3) において，M 回のループによって同時に計算されることに注意する．式 (3) による勾配の近似を図 1 の枠組みで実行する．この算法を SNR (Stochastic Noise Reaction) と呼ぶ．

2.3 SNR の実装

図 1 の枠組みでは，ガウス白色雑音の j 番目の実現値 $\xi_i^j \in N(0,1)$, $i = 1,2,\ldots,n$ をステップ 7 で生成しているが，式 (3) の精度は ξ_i の平均値が 0 であることに強く依存するので，勾配近似のループの前にすべての雑音列を生成しておく．(各反復 k で同じ雑音列を使い回してもよい．) 具体的には，各雑音を $\xi_i^j := \cos(2\pi a)\sqrt{-2\log b}$, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,M$ で生成しておき，

$$\xi_i^j := \xi_i^j - \frac{1}{M}\sum_{j=1}^{M}\xi_i^j, \qquad (4)$$

とすることで $E[\xi_i] = 0$ に正規化する．ここで，a と b は，$(0,1)$-区間上の一様乱数．図 2 は，正規化した $M = 100$ 個の雑音で $f(x) = x^2$ の傾きを近似したもの．多少の誤差はあるが $2x$ の直線が得られている．本稿では，変数の数 n にかかわらず，すべての実験で $M = 100$ とした．

3 SNR と DA の比較

本章では，Hopfield-Tank (HT) 型と Held-Karp (HK) 型の形式化を用いて，SNR と DA を比較する．

3.1 HT 型の TSP 形式化による実験

3.1.1 形式化 (TSP/HT)

次式が HT 型の形式化の目的関数:

$$\begin{aligned} f(x) &= \tfrac{A}{2}\textstyle\sum_X(\sum_i g(x_{Xi}) - 1)^2 \\ &+ \tfrac{B}{2}\textstyle\sum_i(\sum_X g(x_{Xi}) - 1)^2 \\ &+ C\textstyle\sum_X\sum_i g'(x_{Xi}) \\ &+ \tfrac{D}{2}\textstyle\sum_i\sum_X\sum_Y d_{XY}\cdot \\ &\quad g(x_{Xi})(g(x_{Y,i-1}) + g(x_{Y,i+1})). \end{aligned} \qquad (5)$$

この形式化を TSP/HT と呼ぶ．ここで，$x_{Xi}, X \in V, i = 1,2,\ldots,|V|$ は頂点 X がツアーにおいて i 番目かどうかを表し，$g(x)$ はロジスティック関数 $g(x) = \frac{1}{2}(1 + tanh(x))$．この形式化での変数の数は $n = |V|^2$ である．右辺の最初の 2 項がツアー制約，3 番目が許容領域の中心部分のコストを上げるための項，4 番目がツアー長に対応する項になっている．解がツアーを表している時に，$g(x)$ をすべて x で置き換えると，第 4 項はツアー長になる．ロジスティック関数 $g(x)$ を用いることで，許容領域を無制約にし，SNR が変数に雑音を乗せても変数が許容領域を越えないようにしている．解 x は，$\pi(i) = \{X \mid \max_{X\in V} x_{Xi}\}$, $i = 1,2,\ldots,|V|$ が頂点の順列で，かつ $\pi(pos(X)) = X$ である時に許容ツアーと言う．ここで $pos(X) = \{i \mid \max_i x_{Xi}\}$.

3.1.2 数値実験 (TSP/HT)

式 (5) のパラメータは，試行錯誤しながら適当に選び，$A = 5.0$, $B = 5.0$, $C = 1.0$, $D = 4.0$ とした．単位矩形内のランダムな $|V| = 10$ 頂点からなる 10 個のインスタンスを作り，それぞれ 10 回のシミュレーションを実行した．初期解は 0 付近の微小値 $x_{Xi}^0 \in N(0,1/4)$ とした．各シミュレーションは $N = 100$ 反復まで実行し，x_{max}, x_{min} はそれぞれ +10, −10 に設定した．解が上記のような条件で許容ツアーになった時点で，シミュレーションが収束したとみなし，反復回数 k を記録した．N 反復の内に許容ツアーが見つからなかった場合，シミュレーションが失敗したとみなし，全 100 回の試行 (10 インスタンスごとに 10 回の試行) における失敗の割合を失敗率と呼ぶ．平均反復回数は，成功したシミュレーションに関する反復回数の平均とした．

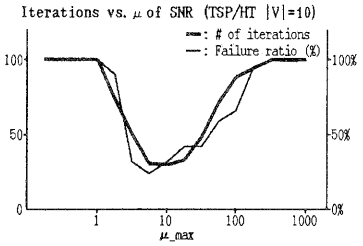

図 3: SNR の TSP/HT に対する反復回数と失敗率

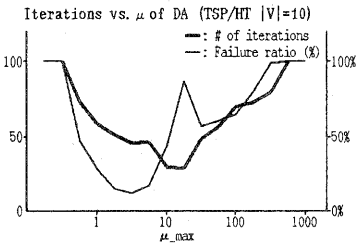

図 4: DA の TSP/HT に対する反復回数と失敗率

図 3, 4 が，SNR と DA を TSP/HT に適用した時の平均反復回数と失敗率を，μ_{max} に対してプロットしたもの．失敗率はいずれも谷型になっていて，$\mu_{max} = 5.6$ および $\mu_{max} = 3.2$ にてそれぞれ最小値となっている．SNR と DA の最小失敗率はそれぞれ 24%と 12%，平均反復回数はそれぞれ 31 と 46 だった．この実験から，μ_{max} が適当に選ばれれば，確率的な手法である SNR の収束の性質が，厳密に勾配を計算する方法と同等であることが分かる．

TSP/HT の形式化で得られるツアーは，大抵は悪い局所解なので，この実験ではツアー長の比較を行わず，許容ツアーへの収束性を比較した．

3.2 HK 型の TSP 形式化による実験

3.2.1 形式化 (TSP/HK)

Held と Karp は [5, 6] において，ツアー制約を緩和して解を 1-tree というグラフで表現する形式化を提案した．1-tree とは頂点 $\{2, 3, \ldots, |V|\}$ に対する最小コスト被覆木 (MST) に，頂点 1 から MST への，異なる 2 本の最小コスト辺を加えたもの (図 5(a))．この形式化では，各頂点 i に関して

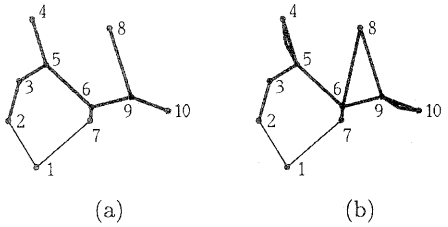

図 5: 1-tree の例と許容ツアーを得る処理の一部．(a) 最小コスト 1-tree．(b) 奇数次数の頂点に対する最小コストマッチングを加えたところ．

実変数 $x_i \in R$ が導入され，頂点間の距離を，

$$d_{ij} := d_{ij} + x_i + x_j, \tag{6}$$

のように変換する．この変換によって最小コストの 1-tree は変化するが，最小コスト (最短) ツアーは影響されない．

目的関数は次のように定義される:

$$f(x) = L + 2\sum_{i=1}^{n} x_i - onetree(x). \tag{7}$$

ここで，L は定数，$n = |V|$，$onetree(x)$ は x で変換された距離における最小コスト 1-tree のコスト．本章での数値実験では，Lin-Kernighan (LK) 法 [8] で得たツアー長を L に設定した．(L を許容ツアー長にすると $f(x) \geq 0$ となる．) 式 (7) を最小化する場合，各頂点 i の次数 $deg(x, i)$ が 2 に近づくように解 x を更新し，最終的に 1-tree を許容ツアーに一致させたい．(許容ツアーはすべての頂点の次数が 2 の 1-tree．) もし x_i が微小値 dx_i だけ増加した時に，対応する最小コスト 1-tree が変化しなければ，$f(x)$ の増加分は $dx_i(2 - deg(x, i))$ となる．したがって，DA で用いる劣勾配として $-deg(x, i)$ を利用できる．($2\sum_i c = \sum_i c \cdot deg(x, i) = 2nc$ のため，定数が $f(x)$ に影響しないことに注意．)

Held と Karp の形式化では許容ツアーが得られるとは限らないので，次に示すような Christofides 法 [7] の一部を用いて許容ツアーを得るようにする．この形式化を TSP/HK と呼ぶ．

$CH(x)$ の手続き

1. 変換された距離での最小コスト 1-tree を得る．
2. 奇数次数の頂点に対する最小コストマッチングを 1-tree に加える (図 5(b))．

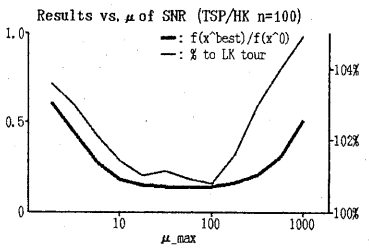

図 6: SNR の TSP/HK に対する結果とツアー長

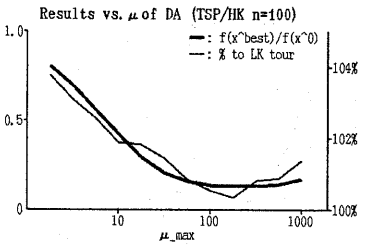

図 7: DA の TSP/HK に対する結果とツアー長

3. 次数が 4 以上 ($deg(x,i) \geq 4$) の頂点 i に接続している 2 本のエッジを切って短絡する．その内 1 本は，接続している辺の中で，変換前の距離で最も長いものとする．もう一本は，それを切るとグラフが非連結にならないような辺の中で，変換前の距離で最も長いものとする．
4. まだ次数 4 以上の頂点があれば 3 に戻る．
5. 頂点の順列 $\pi(i), i = 1, 2, \ldots, |V|$ を出力する．

この形式化では，目的関数の大小関係が必ずしもツアー長 (式 (1)) のそれと一致しない．(つまり $f(x) > f(y)$ でも $h(CH(x)) > h(CH(y))$ とは限らない．) 最小コスト 1-tree，最小コストマッチングの計算，$CH(x)$ の処理はそれぞれ多項式時間で実行できる．

3.2.2 数値実験 (TSP/HK)

単位矩形内のランダムな $|V| = 100$ 頂点からなる 10 個のインスタンスで実験を行った．初期解は $x_i^0 = 0$，許容領域は無制約 ($x_{max} = +\infty, x_{min} = -\infty$) とした．各シミュレーションを $N = 100$ 反復まで実行し，x^{best} の他に，最短のツアーに対応する解 x^{tour} も記録した．(x^{tour} は必ずしも x^{best} と一致しない．)

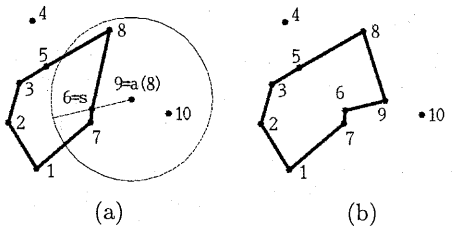

図 8: 追加法で頂点 9 をサブツアーに挿入する例．(a) 挿入位置の候補を探索: (1,7), (7,6), (6,8), (8,5)．(b) 頂点 9 を頂点 6, 8 の間に挿入．

図 6 は，SNR によって得られた $f(x^{best})/f(x^0)$ の平均値と $h(CH(x^{tour}))/LK$ の平均値を，μ_{max} に対してプロットしたもの．ここで LK は LK 法で得られたツアー長を表す．$f(x^{best})/f(x^0)$ と $h(CH(x^{tour}))/LK$ の値は，それぞれ $\mu_{max} = 56$, $\mu_{max} = 100$ で最小値となっている．$h(CH(x^{tour}))$ の最小値は LK の 100.8% で，これは，LK の値が最適解を 2%程度しか越えないことから，最適解を 3%程度しか越えない値と考えられる．図 7 が DA の結果である．$f(x^{best})/f(x^0)$ と $h(CH(x^{tour}))/LK$ は共に，$\mu_{max} = 178$ において最小値となっている．$h(CH(x^{tour}))$ の最小値は LK の 100.3%.

SNR と DA の結果 (図 6, 7) を比較すると，HK 型の目的関数の最小化において，SNR が DA と同等の性能を持っていることが分かる．ここで強調したい点は，微分も劣勾配も参照しない確率的な手法が，HK 型の形式化での TSP 関数を局所最小化できるということである．

SNR, DA, LK の実行時間は，PowerPC 601 112-MHz CPU の IBM RS/6000 で，それぞれ 65, 1.5, 0.15 秒．これは，TSP/HK の形式化が，許容ツアーを得る方法としては LK 法程にはよくないことを意味する．ただし，HK 型の形式化は下界値を得る方法としては意味がある．

4 追加法に基づく TSP 形式化

本章では，追加法と呼ばれる TSP のヒューリスティックに基づく形式化を提案し，それに SNR を適用する．この形式化の勾配を求める他の方法を筆者らは知らないので，本章では DA は扱わない．

4.1 追加法

追加法は，ある1頂点からなるサブツアーから出発して，サブツアーに1つずつ残りの頂点を追加してゆく．1つの頂点を追加する時，ツアー長の増加が最も小さい位置に挿入する．最初の1頂点とサブツアーに追加してゆく頂点の順列 $a(i), i = 1,2,\ldots,|V|$ を「追加列」と呼ぶ．a があらかじめ与えられる (オフラインの) 追加法は，次のような手順になる:

追加法 $AH(a)$ の手続き

1. $a(1)$ を1頂点からなるサブツアーとする．
2. **For** $i = 2,3,\ldots,|V|$ **do begin**
3. サブツアー内で $a(i)$ に最も近い頂点を s とする．
4. $a(i)$ を中心とする半径 $2d_{s,a(i)}$ の円に含まれる頂点集合を $W(s,a(i))$ とする (図 8(a))．
5. サブツアー内の頂点 $t \in W(s,a(i))$ の前後の位置 (辺) で，$a(i)$ を挿入した時に最もツアー長の増加が小さいものを選ぶ．
6. ステップ5で選んだ辺を切って，その位置に $a(i)$ を挿入する (図 8(b))．
7. **end;**
8. 頂点の順列 $\pi(i), i = 1,2,\ldots,|V|$ を出力する．

a がランダムな順列の時，上記の手続きをランダム追加法 (Random Addition: RA) と呼ぶ．ステップ3, 4における最近傍頂点探索と，固定半径の近傍頂点探索は，k-d 木 [9] を使えば効率的に実装できる．(MST の計算と $CH(x)$ の処理より高速．)

追加法において，入力の追加列 a における局所的な変化は，結果のツアーに大きくは影響しない．(これに対し最近傍法では，途中の1手の違いがその後の振るまいを大きく変えてしまう．) 例えば，サブツアーが $|V|-2$ 頂点からなる時 (図 8(b))，残りの2頂点，$a(|V|-1)$ と $a(|V|)$ をどちらの順番で追加しても，多くの場合同じ結果を産む．しかし，追加列の中で前半にあるのか後半にあるのかといった，頂点の順位は，結果に大きな影響をもたらす．例えば，サブツアーに近い点を追加する方法 (最近追加法) のツアーは，ランダム追加法と比べて大きく悪い．つまり，最近追加法のように頂点座標に偏りがある追加列ではツアーは長くなる．むしろランダムな追加列がよい．したがって，追加列内の頂点の順位は，ほぼ独立にツアー長に関係すると考えられる，(各種追加法の比較については，例えば [10] を参照．)

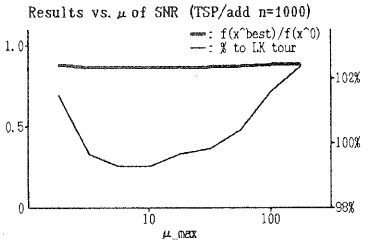

図 9: SNR を TSP/add に適用した結果

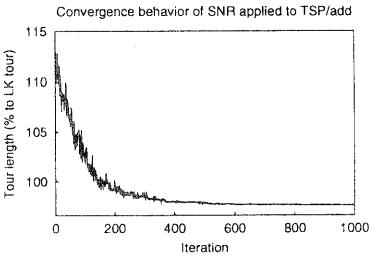

図 10: SNR の TSP/add に対する収束の様子

4.2 追加法に基づく連続関数

前節で示した追加法 $AH(a)$ は離散ベクタ a をとるので，このままでは勾配法を適用できない．連続緩和するために，$n = |V|$ 次元の実ベクタ x を導入する．各成分 x_i は追加列内の頂点 i の優先順位で，x_i を降順にソートすることで $x_{a(i)} \geq x_{a(i+1)}$ となるように追加列を生成する．この追加列を用いる次の形式化を TSP/add と呼ぶ:

$$f(x) = h(AH(decreasingorder(x))), \quad (8)$$

ここで，$decreasingorder(x)$ は優先順位ベクタ x から追加列 a を生成し，$AH(a)$ は追加法で順列 π を生成し，$h(\pi)$ (式 (1)) がツアー長を計算する．

TSP/add の目的関数 (8) は，プログラムの手続きとして実現されるので，微分に基づく勾配法は適用できない．一方，SNR は明示的に目的関数を微分しないので，TSP/add の形式化に適用可能である．

表 1: TSPLIB インスタンスに適用した結果 (最適値に対する百分率)

TSPLIB	SNR	LK	kroA150	100.50	100.96	u574	101.70	102.11
eil51	100.23	101.41	kroB150	100.02	100.92	rat575	103.31	102.66
berlin52	100.00	100.00	pr152	100.18	100.84	p654	101.68	101.72
st70	100.59	101.04	u159	101.31	100.00	d657	101.52	102.41
eil76	101.49	100.56	rat195	102.37	101.38	u724	102.48	104.25
pr76	100.00	100.86	d198	100.47	100.56	rat783	102.87	101.99
rat99	100.74	101.16	kroA200	102.08	103.97	dsj1000	103.23	103.62
kroA100	100.00	100.00	kroB200	102.11	102.63	pr1002	102.53	102.61
kroB100	100.42	101.91	pr226	100.06	100.05	u1060	102.62	102.47
kroC100	100.01	104.08	ts225	102.35	100.25	vm1084	102.87	103.28
kroD100	100.00	101.61	tsp225	103.29	100.36	pcb1173	104.93	102.60
kroE100	100.42	100.17	gil262	100.55	102.61	d1291	105.17	104.50
rd100	100.43	100.00	pr264	101.19	105.16	rl1304	104.50	103.70
eil101	100.95	101.27	a280	103.76	103.10	rl1323	103.75	104.33
lin105	100.00	102.04	pr299	101.56	102.32	nrw1379	103.17	104.35
pr107	100.31	100.30	lin318	101.98	101.35	fl1400	101.36	100.93
pr124	100.00	100.23	linhp318	103.35	103.03	u1432	103.97	102.06
bier127	100.97	102.87	rd400	101.92	102.66	fl1577	105.20	102.91
ch130	100.43	101.65	fl417	100.82	102.64	d1655	104.46	103.92
pr136	100.61	100.15	pr439	101.48	100.72	vm1748	103.46	103.08
pr144	100.39	100.09	pcb442	103.01	103.01	u1817	106.93	104.52
ch150	100.41	100.25	d493	102.21	102.07	rl1889	104.49	105.40
						Ave.	101.86	102.03

4.3 数値実験 (TSP/add)

単位矩形内のランダムな $|V| = 1000$ 頂点からなる 10 個のインスタンスに，SNR, RA, ランダム局所探索, LK を適用した．SNR のシミュレーションは初期解を $x_i^0 = 0$ とし，反復回数は $N = 1000$ とした．図 9 は，$f(x^{best})/f(x^0)$ と $f(x^{best})/LK$ の 10 個のインスタンスに関する平均値を，μ_{max} に対してプロットしたもの．目的関数は，$\mu_{max} = 10$ 付近で最小化しており，その値は LK の 99.3%だった．これは最適値の 1%程度に相当する．図 10 は，ある 1 つのインスタンスに，$\mu_{max} = 10$ に設定した SNR を適用した時の収束の様子を示している．最初の約 400 反復の間は大きく上下しているが，その後徐々に局所解に収束している．局所解での x の各成分は $[-20, +30]$ の範囲だった．SNR と LK の実行時間は，PowerPC 601 112-MHz CPU の IBM RS/6000 で，それぞれ 2808 秒と 6.4 秒．

比較のために，RA を異なる追加列で $M \times N = 100,000$ 回実行し，最もよい値を記録した．(SNR の中では AH が $M \times N$ 回実行される．) また，x の近傍解を $x_i + N(0,1)$ で生成し，目的関数値がよければ近傍解に遷移するという局所探索を，(遷移しない反復も含め) $M \times N$ 反復実行した．これをランダム局所探索と呼ぶ．RA とランダム局所探索で得られた解は，10 個のインスタンスの平均で，それぞれ LK の 109%, 108% だった．つまり，ランダムな追加列やランダムな降下法では，勾配法に基づく方法よりもよい局所解を見つけられない．

この実験の結果，提案した TSP/add の形式化 (8) が，確率的な勾配法が局所解を得るという意味で「準」連続な (つまり微分不可だが滑かな) 地形を持っていると言うことができる．

4.3.1 TSPLIB への適用

ステップ幅 μ_k を決める定数 μ_{max} の設定を省くために，直線探索を行うよう変更した SNR を TSPLIB [11] に適用した．変更した SNR では，式 (2) (図 1 のステップ 11) において，

$$\min_s f(\tilde{x}),\ \tilde{x} = 0.01ns\frac{\delta x}{|\delta x|},\ s = 1, \ldots, 10, \quad (9)$$

を求めて $\mu = 0.01ns$ とした (n は次元，s は整数)．また，$N = 1000$ とし，x^{best} が 100 反復連続して更新されなかった場合は処理を打ち切った．この SNR と LK を，$|V| \leq 1889$ で，最適値が既知の，平面上の距離を仮定した 65 個のインスタンスに適

用した．その結果を表1に示す．TSP/addの形式化とSNRによって，LKと同等のツアーが得られている．

5 おわりに

白色雑音を用いる勾配法SNRを提案し，3つのTSPの形式化で評価した．用いた形式化は，Hopfield と Tank (HK) 型，Held と Karp (HT) 型，そして新しく提案した追加法に基づく形式化である．数値実験を通して，SNRには次の特長があることが分かった:

- 微分に基づく勾配法や劣勾配法と同様の収束性を持つ．
- 対象とする目的関数は，準連続な関数であればどんな関数でもよい．
- プログラムの手続きとして定義された関数など，微分の利用できない関数に対しても適用できる．

離散的なヒューリスティックスと数理計画法の間には，後者の目的関数への制限が強いために，適用分野に隔りがあるが，SNRの上記のような性質がこの隔りを埋めることが期待される．例えば，本稿で提案した追加法に基づくTSPの形式化TSP/addは，連続値のパラメータをとり，ヒューリスティックな手続きによって離散値の解を生成する．SNRによるTSP/addの解法は，確率的な勾配法に基づく，パラメトリック最適化のヒューリスティックになっている．(つまり，TSPのツアーを解とするのでなく，追加法という手続きのパラメータを解にしている．) このような形のパラメトリック最適化に勾配法を用いる手法は，これまで注目されていなかった．

残された課題として，TSP/addの形式化における勾配とは何か，なぜ勾配法がうまく適用できるのかといった考察を深めること，およびSNRの他の組合わせ最適化問題への適用がある．

参考文献

[1] H. Okano and M. Koda, "A New Noise-Based Gradient Method and Its Applications," *IBM Research Report*, RT0390, 2000.

[2] M. Koda and H. Okano, "A New Stochastic Learning Algorithm for Neural Networks," *Journal of Operations Research Society of Japan*, vol. 43, pp. 469-485, 2000.

[3] J. J. Hopfield and D. W. Tank, "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, vol. 52, pp. 141-152, 1985,

[4] R. D. Brandt, Y. Wang, A. J. Laub, and S. K. Mitra, "Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem," *Proc. of IEEE International Conference on Neural Networks*, vol. II, pp. 333-340, 1988.

[5] M. Held and R. M. Karp, "The Traveling-Salesman Problem and Minimum Spanning Trees," *Operations Research*, vol. 18, pp. 1138-1162, 1970.

[6] M. Held and R. M. Karp, "The Traveling-Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming*, vol. 1, pp. 6-25, 1971.

[7] N. Christofides, "Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem," *Report 388*, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, 1976.

[8] S. Lin and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," *Operations Research*, vol. 21, pp. 498-516, 1973.

[9] J. L. Bentley, "Fast Algorithms for Geometric Traveling Salesman Problems," *ORSA Journal on Computing*, vol. 4, pp. 387-411, 1992.

[10] H. Okano, S. Misono, and K. Iwano, "New TSP Construction Heuristics and Their Relationships to the 2-Opt," *Journal of Heuristics*, vol. 5, pp. 71-88, 1999.

[11] G. Reinelt, "TSPLIB–A Traveling Salesman Problem Library," *ORSA Journal on Computing*, vol. 3, pp. 376-384, 1991.