A-025

ノード配置問題に対するアント最適化法
Ant Colony Optimization for the Node Placement Problem

大倉慶一 † **片山 謙吾** ‡ **南原 英生** ‡ **西原 典孝** ‡
Keiichi Ohkura Kengo Katayama Hideo Minamihara Noritaka Nishihara

1. はじめに

近年，波長分割多重（Wavelength Division Multiplexing, WDM）技術を利用したマルチホップシステムのネットワークにおいて，伝送効率が最良となる波長割当（ノード配置）を求めることが重要とされている．これまでに，WDM のネットワークトポロジーの一つである双方向マンハッタンストリートネットワーク（Bidirectional Manhattan Street Network, BMSN）を対象に，伝送効率が最良となるノード配置を求める研究がなされている．本論文では，BMSN における最適なノード配置を求める問題をノード配置問題（Node Placement Problem, NPP）と呼ぶ．NPP は NP 困難 [2] [4] であることが知られている．

NPP に対してはこれまでに，代表的なメタ戦略アルゴリズム [8] である遺伝的アルゴリズムやタブサーチ [5]，アニーリング法 [10] が提案されている．また我々は，反復局所探索法 [3] と Memetic アルゴリズム [9] を提案しており上述した手法より良好な結果を算出することを確認している．その他，代表的なメタ戦略としてアント最適化法 (Ant Colony Optimization, ACO) [1] があるが，NPP に対する適用例は報告されていない．そこで本論文では，ACO と k-swap 局所探索法 (k-swap Local Search, KLS) [3] をハイブリッドしたメタ戦略アルゴリズムを提案する．

本論文は，まず 2 章で NPP と NPP に対して提案されている代表的なメタ戦略について説明する．3 章では本 ACO の特徴である，アントサーチ，フェロモン情報の更新，問題領域固有の情報，k-swap 局所探索法，そして Restart 処理の 5 つの要素について述べる．次いで 4 章では，提案法の性能を評価するために NPP のベンチマーク問題例に適用し，NPP に対して強力なメタ戦略アルゴリズムとして知られている反復局所探索法 [3]，Memetic アルゴリズム [9] との比較を行った．その結果，提案法は小規模な問題例に対しては，IKLS, MA と同等かより良い結果を算出し，大規模な問題例に対しては，MA には若干劣るものの，IKLS と比較した場合，平均的に優れた結果を算出することを示す．

2. ノード配置問題と代表的なメタ戦略

本論文で対象とする双方向マンハッタンストリートネットワーク（BMSN）は，図 1 のような，$n = m \times m$ 個 (m は正数) のネットワークノードが m 行 m 列の格子状に規則的に配置され，各行各列が双方向のトーラスを形成する論理トポロジーを持つ．座標 (x, y) は BMSN の論理トポロジー上の場所を表す．各ノード間のトラヒックの流れを表すトラヒック行列を $T = \{t_{i,j}\}(0 \leq i, j \leq n-1)$ と定義し，行列の要素 $t_{i,j}$ はノード i からノード $j(i \neq j)$ へのトラヒック量を表す．ノード配置問題の目的は，以下に示す BMSN のネットワーク全体に流れるトラヒック量を表す目的関数 f を最小化することである．

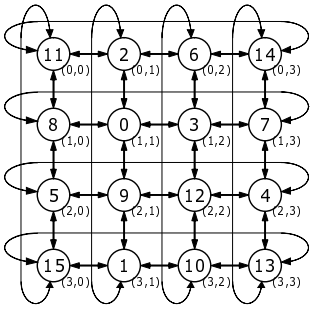

図 1: BMSN の論理トポロジー

$$f(\sigma) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} t_{i,j} \times h(\sigma_i, \sigma_j), \qquad (1)$$

ただし，σ_i は $\{0, \ldots, n-1\}$ の順列で構成され，ノード i の論理トポロジー上の場所 ($\lfloor \sigma_i/m \rfloor$, $\sigma_i \bmod m$) を表し，$h(\sigma_i, \sigma_j)$ は ($\lfloor \sigma_i/m \rfloor$, $\sigma_i \bmod m$) と ($\lfloor \sigma_j/m \rfloor$, $\sigma_j \bmod m$) に配置されたノード間の最短経路ホップ数を表す．

NPP に対してはこれまでに，様々なメタ戦略アルゴリズムが提案されている．嘉藤らは文献 [5] において，遺伝的アルゴリズム（Genetic Algorithm, GA）とタブサーチを提案しており，米津，舩曳らの研究グループ [10] では，貪欲法，アニーリング法を提案している．

また我々はより高性能なメタ戦略アルゴリズムを目指し，反復局所探索法 [3] と Memetic アルゴリズム [9] を提案した．以下では，NPP に対して提案されているこれらの代表的なメタ戦略アルゴリズムについて簡単に述べる．

嘉藤ら [5] は，NPP に対する最初の GA とタブサーチを提案した．一般に GA は，集団と呼ばれる解集合を保持し，集団に対して，交叉，突然変異，選択・淘汰の遺伝的操作を適用しながら探索を行う多点探索手法である．タブサーチは，現在の解の近傍の中から，過去に探索済みの近傍移動などに制限を加えるなどして，より良好な解を探索するメタ戦略である．このような制限により，元の解に戻ること（サイクリング）を回避している．嘉藤らによる，GA とタブサーチの比較実験の結果では，初期解生成に貪欲法を用いたタブサーチが有効であることが示されている．

†岡山理科大学大学院工学研究科
‡岡山理科大学工学部情報工学科

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

上述の嘉藤ら [5] の研究を受け，米津，舩曳らはアニーリング法 [10] を提案した．一般にアニーリング法は，現在の解の近傍内の各解に，解の良さに応じた遷移確率にもとづき，次の解へ遷移する処理を繰り返す．改悪解であっても遷移する確率を与えることによって，局所最適解に陥ることを回避しようとするメタ戦略である．遷移確率は，物理現象の焼きなましのアイディアにもとづいており，温度と呼ばれるパラメータによって調整される．米津らのアニーリング法 [10] は，貪欲法で初期解を生成し，その解に対してアニーリング法を適用することで，解の改善を行う階層型近似アルゴリズムと呼ばれる解法である．タブサーチとの比較実験の結果，アニーリング法にもとづく階層型近似アルゴリズムが有効であることが示されている．

我々の研究グループは，米津，舩曳らの研究を受けて，反復 k-swap 局所探索法 (IKLS) [3] と Memetic アルゴリズム (MA) [9] を提案した．IKLS は，ILS(Iterated Local Search, ILS) をベースにしたメタ戦略アルゴリズムであり，Kick と呼ばれる，遺伝的アルゴリズムの突然変異に相当する処理と局所探索法で構成される．IKLS では，局所探索法として，k-swap 局所探索法 (KLS)，Kick として，Cross Kick とよぶ方法で構成している．また，一般に MA は，GA と局所探索をハイブリッドした手法として知られている．複数個の解を用いた確率的な多点探索にもとづく大域的探索と，局所的な探索の融合によって，より良好な近似解への接近を試みるメタ戦略アルゴリズムである．我々が提案するこれらの IKLS と MA は比較実験により，上述の他のメタ戦略よりも比較的優れた解を算出可能であることを確認している．

3. NPP に対するアント最適化法

ACO はアリの群れ行動を模倣したメタ戦略アルゴリズムであり，複数のアリエージェントが発信するフェロモン情報をもとに，組合せ最適化問題の解を構築する手法である．

ここでは，NPP に対する新しい近似解法として，ACO と我々の局所探索法である KLS をハイブリッドしたメタ戦略アルゴリズムを提案する．本 ACO の擬似コードを図 2 に示す．本文中の num_{ant} はアリエージェントの数を示している．

本 ACO は，初期化の処理（Line 1~6）として，ランダムに num_{ant} 個の解を生成し，それらすべてに KLS を適用し局所最適解を算出する．次いで，ACO の探索で必要となる情報 ACO-Info を初期化する．このとき，Line 6 では，Line 1~4 で得られた num_{ant} 個の局所最適解にもとづきフェロモン情報の初期化を行う．ここで num_{ant} 個の局所最適解を最良解集団とする．最良解集団とは ACO で生成された局所最適解の中から，評価値の良いものから順にアリエージェントと同数の解を保持した集団である．最良解集団の更新は Line 7~15 のループごとに行われる．

以降の Line 7~15 のループ処理が本 ACO の主要処理である．まず，Line 9 で ACO の処理であるアントサーチによる解の再構成を行う．アントサーチでは，

```
     procedure Ant Colony Optimization with KLS
     begin
1      for k := 1 to num_ant do ;
2        generate a random solution σ^k;
3        σ^k := k-swapLocalSearch(σ^k);
4      endfor
5      σ_best := the best solution of {σ^1, ..., σ^num_ant};
6      InitializeACO-Info( );
7      repeat
8        for k := 1 to num_ant do ;
9          σ^k := AntSearch( );
10         σ^k := k-swapLocalSearch(σ^k);
11         if f(σ^k) < f(σ_best) then σ_best := σ^k; endif
12       endfor
13       UpdateACO-Info( );
14       if restart=true then Restart( );
15     until terminate = true;
16     return σ_best;
     end;
```

図 2: NPP に対する ACO の流れ

フェロモン情報 τ と問題領域固有の情報 η を利用し解の再構成を行う．Line 9 で再構成された解 σ^k に対して Line 10 で KLS を適用し，局所最適解を求める．Line 9，10 を繰り返し，アリエージェントと同数の num_{ant} 個の解を算出する．算出された num_{ant} 個の局所最適解の情報にもとづき Line 13 にてフェロモンの更新を行う．Line 14 では Restart 処理を適用するか否かの判定を行う．ここでの判定条件は，暫定最良解 (σ_{best}) が，Line 7~15 のループでアリエージェントの数 num_{ant} と等しいループ回数の間更新されなかった場合に Restart 処理を適用するものとする．以上 Line 7~15 を終了条件を満たすまで繰り返す．

以下に本 ACO の主要な構成要素である，アントサーチ，フェロモン情報の更新，問題領域固有の情報，k-swap 局所探索法，リスタート処理について述べる．

3.1. アントサーチ

アントサーチは与えられた解に対して，フェロモン情報と問題領域固有の情報にもとづき確率的にノードを選択し解を再構成する処理である．本 ACO では，最良解集団に対してアントサーチを行う．

アントサーチでは現在，アリエージェント k がノード i に存在するものとし，N^k をエージェント k の未割当ノードの集合とする．このとき，N^k の中から次の割当ノード j を選択する確率 p は以下の式で与える．

$$p^k(i,j) = \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{l \in N^k} [\tau(i,l)]^\alpha [\eta(i,l)]^\beta} \qquad (2)$$

式 (2) の τ はフェロモン情報を表し，η は問題領域固有の情報を表している．α と β はそれぞれ非負の実数で，フェロモンにより段階的に形成される大域的な情報と，問題領域に基づく局所的な情報をどのくらい重要視するかのパラメータである．

アントサーチにおける解の再構成の流れを図 3 に示す．グラフは再構成を行う解を表している．グラフの丸の中の数字はノード番号を示し，ノード番号の右下の括弧はスロット座標を示している．

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

まずはじめに，与えられた解からランダムにノード i を選択し，ノード i が割り当てられているスロット座標 $(\lfloor \sigma_i/m \rfloor, \sigma_i \bmod m)$ を基点とする行 $(\lfloor \sigma_i/m \rfloor, y(0 \leq y < m))$ と列 $(x(0 \leq x < m), \sigma_i \bmod m)$ に割当てられているノードを取り除く．ここで取り除いたノードの集合を N^k とする．図 3(a) では基点ノード $i = 3$ とし，基点スロット座標は $slot_i = (1, 2)$ としている．この場合 $N^k = \{0, 3, 6, 7, 8, 10, 12\}$ となる．

次いで，取り除かれたスロットの中から，基点スロット以外のスロットをランダムに一つ選択する．選択されたスロットをノード配置予定スロットとよぶ．図 3(b) ではスロット座標 (0, 2) が選択されたとする．

図 3(b) において，スロット（0, 2）に隣接しているノードの番号は 2 と 14 である．ここで，式 (2) により，アリエージェント k がノード 2 と 14 に存在すると仮定した場合の，N^k に含まれるノード群のそれぞれに対して確率 p を求める．ノードの選択は確率 p にもとづいたルーレット選択で行い，確率 p が高ければ高いほど選択されやすくなる．なお，N^k に含まれるノード群のそれぞれに対して求めた確率 p の総和を 1 とする．ここで選択されたノードは配置候補ノードとする．配置候補ノードの集合は，配置予定スロットに隣接する配置済みノードの個数と等しい数のノードで構成される．さらに配置候補ノードの集合の中からルーレット選択により決定したノードを配置予定スロットに配置する．

図 3(c) はノードの選択処理によりスロット座標 (0, 2) にノード 12 が配置されたことを示している．配置されたノードは N^k から削除される．よって $N^k = \{0, 3, 6, 7, 8, 10\}$ となる．以上の処理を未配置ノード集合 N^k が空集合になるまで繰り返し行い，解の再構成を行う．

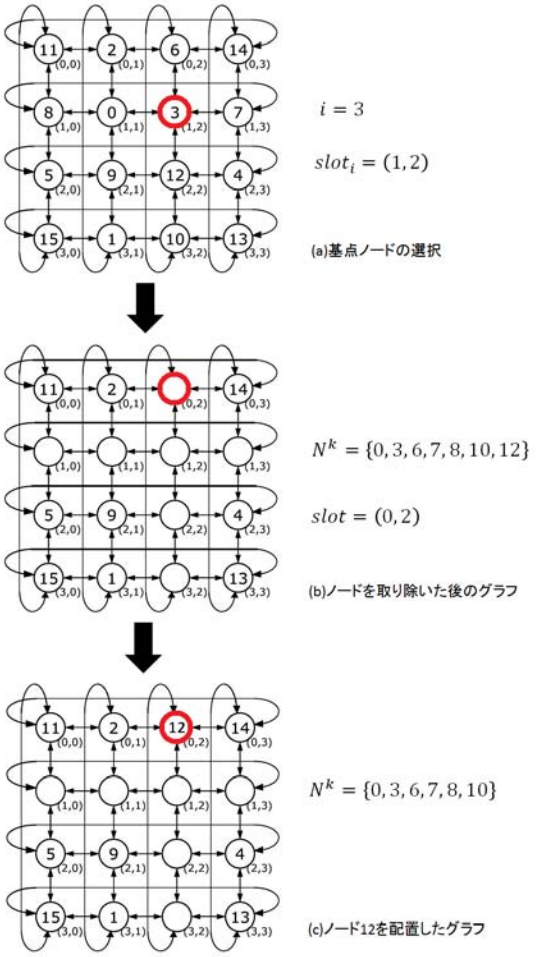

図 3: アントサーチにおける解の再構成の流れ

3.2. フェロモン情報の更新

アント最適化手法では，フェロモンという概念を導入することが重要な要素となってくる．これは，アントの過去の探索履歴を他のアントに反映させることで，評価値の良い解を生成することができるためである．本 ACO では，フェロモンの更新ルールを以下に示す方法とした．

$$\tau(i,j) \leftarrow (1-\rho)\tau(i,j) + \sum_{k=1}^{num_{ant}} \Delta\tau^k(i,j), \qquad (3)$$

$$\Delta\tau^k(i,j) = \begin{cases} f(\sigma_{worst})/f(\sigma^k) & \text{if } (i,j) \in \sigma^k \\ 0 & \text{otherwise,} \end{cases} \qquad (4)$$

ここで，σ^k と $f(\sigma^k)$ はアントサーチにより得られたノード配置 σ^k とその評価値 $f(\sigma^k)$ を表している．σ_{worst} と $f(\sigma_{worst})$ は各繰り返しで得られた最悪の配置 σ_{worst} とその評価値 $f(\sigma_{worst})$ を表す．また，ρ はフェロモンの蒸発率を表すパラメータであり，通常は $0 < \rho < 1$ の間で設定される．このパラメータ ρ の影響によって，過去のフェロモンの量は徐々に減少していき，新しいフェロモンの情報が大きな影響をもつことになる．このフェロモン更新ルールにより，より良い解が見つかったとき即座にその解の周辺を集中的に探索することが可能となる．またフェロモン情報の更新は BMSN グラフの枝に対して行う．

3.3. 問題領域固有の情報

問題領域固有の情報とは，アリエージェントの探索を効率良く行わせるための情報である．文献 [1] において巡回セールスマン問題（Traveling Salesman Problem, TSP）を対象に提案されている ACO では，問題領域固有の情報を都市間の距離の逆数としている．これは，次の都市までの距離が短い都市を，アリエージェントが次の移動先として選択しやすくすることを目的としている．本論文で対象とする NPP では，解の評価は，トラヒック量と，高付加トラヒックが割り当てられているノード間のホップ数を計算し行う．各ノード間に割り当てられているトラヒック量と，そのノード間のホップ数を掛け合わせた値の総和が最小となるノード配置を求めることが NPP の目的である．本実験で使用したベンチマーク問題例では，高負荷トラヒック量を 1，低負荷トラヒック量を 0 と定義しているため，高付加トラヒックが割り当てられているノード間のホップ

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

```
    procedure k-swap Local Search(σ)
    begin
1     σ_best := σ, g_LastImp := ∞, P_out := {0, ..., n − 1};
2     repeat
3       σ := σ_best, g := 0, g_best := 0, P_in := {0, ..., n−1};
4       i_base := random(P_out);
5       P_in := P_in\{i_base}, P_out := P_out\{i_base};
6       repeat
7         find a node i with min_{i∈P_in} δ_i := SwapGain(i_base, i, σ);
8         σ := SwapMove(i_base, i, σ);
9         g := g + δ_i, P_in := P_in\{i};
10        if g_best > g then σ_best := σ, g_best := g;
11      until P_in = ∅ or g > g_LastImp;
12      if g_best < 0 then P_out := {0, ..., n−1}, g_LastImp := |g_best|;
13    until P_out = ∅;
14    return σ_best;
    end;
```

図 4: NPP に対する k-swap 局所探索法

数が 1 の状態（隣接している状態）が最良のノード配置となる．

そこで本 ACO での問題領域固有の情報は，アントサーチにおいて，次に配置するノードと高付加が割り当てられている配置済みノード間のホップ数の和の逆数とする方法を採用した．これは文献 [1] で定められた問題領域固有の情報の距離の部分をホップ数に置き換えたものである．ただし次に配置するノードに対して高負荷が割り当てられていない場合はホップ数が 0 であるため逆数を取ることが出来ない．仮にノード j に対して高負荷トラヒックが割り当てられていないとする．この場合は問題領域固有の情報は 0 として扱い，確率 $p(i,j)$ も 0 となる．そしてアントサーチにおいて N^k に含まれるすべてのノードの確率 p が 0 だった場合，N^k の中からランダムにノードを 1 つ選択し配置する．配置されたノードは N^k から取り除く．

3.4. k-swap 局所探索法

我々は，NPP に対する高性能な局所探索法として，可変深度探索 (Variable Depth Search, VDS) [6] [7] のアイデアにもとづく k-swap 局所探索法 (KLS) [3] を提案している．KLS は，単純な近傍操作を連鎖的に適用することで生成される解集合をあらためて大きな近傍と捉えることによって，より広範囲の近傍を効率的に探索する方法である．NPP の単純な近傍として，1 組のノードの位置を交換する 1-swap 近傍があり，VDS のアイデアに基づく連鎖的な 1-swap 近傍の適用により得られる解集合を k-swap 近傍とよぶ．図 4 に KLS の手順を示す．

このアルゴリズムは外ループ (Line2∼13) と内ループ (Line6∼11) の二つのループ処理から構成される．擬似コードで用いられている変数について説明する．

σ は現在保持している解である．また，σ_{best} は KLS 内で見つかった最良解である．g は探索によって得られた解と探索を開始する前の解の評価値の改善幅（ゲイン値）である．KLS の内ループでは，現在の解に対して任意の要素 i, j を交換し，そのゲイン値を求める (Line7∼9)．そして，ゲイン値が改善ならば交換後の解 σ を σ_{best} に，ゲイン値 g を g_{best} に保存する (Line10)．一度交換した要素は，内ループの処理が終わるまで交換禁止とする (Line9)．交換できる要素がなくなれば内ループは終了する (Line11)．外ループでは，内ループで生成した k-swap 近傍の中で目的関数値を最も改善する解へ移動する操作を，k-swap 近傍内に改善解が存在しなくなるまで繰り返す．

また，k-swap 近傍では近傍の深さが深いときには，交換可能なノードが限定されてしまい，改善解を発見できる可能性が非常に低くなる場合がある．したがって，ある程度の深さまで探索を行い，改善解が見つからなかった場合には，強制的に近傍の生成を終了させることが望まれる．具体的な処理としては，前回の外ループで発見した最良解のゲイン値の絶対値である $g_{LastImp}$ を (Line12) で保持しておき，内ループの解のゲイン値 g が $g_{LastImp}$ を超えたときに内ループを終了する (Line11)．これにより処理の高速化を図っている．

3.5. Restart 処理

我々は予備実験を通して，ACO では探索の進行に伴い，探索が集中化しすぎてしまい，早い段階で探索が停滞気味になっていることを確認している．

そこで，本論文では探索に多様性を持たせることを目的とした Restart 処理として，最良解集団 (**3** 参照) を用いた突然変異処理を適用した．Restart 処理では，まず最良解集団に含まれるそれぞれの解に対して突然変異処理を適用する．ここで適用する突然変異処理として，我々が文献 [3] で提案している Cross–Kick を適用する．そして突然変異処理後の解に対して，さらに KLS を適用し局所最適解を算出する．解の選択処理として，これまで保持していた解集団と新たに算出された解の中から評価値の良い解を選択し，新たな最良解集団とする．この時，探索に多様性を持たせるという観点から，解の構造が異なるものを優先的に選択する．その後，更新された最良解集団の解にもとづきフェロモン情報の更新を行う．

以下に Restart 処理中で適用する突然変異処理である Cross–Kick について説明する．

Cross–Kick [3] は，ある基準にもとづきノード i を選択し，ノード i の行と列に割当てられているノード群をランダムに入れ替える方法である．基点ノード i の選択方法は，グラフを構成するすべてのノードの中からランダムに一つ選択するものとした．

4. 実験結果

本 ACO の性能を評価するために，2 種類の実験を行った．

実験 I は，提案法 ACO with KLS のパラメータの検証を行う．パラメータの検証では，アリエージェントの数である num_{ant} の値が 10 の場合と m(問題サイズ 16，64，256，1024 に応じてそれぞれ 4，8，16，32) とした場合の比較を行う．この実験 I を通して，num_{ant} を変更した場合での探索性能の検討を行う．

次に実験 II は，提案法 ACO と多スタート k-swap 局所探索法（Multi-start k-swap Local Search, MKLS），反復局所探索法（Iterated k-swap Local Search, IKLS），および Memetic アルゴリズム（MA）

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

表 1: 提案法 ACO のパラメータの検証

n	Opt	ACO with KLS ($num_{ant} = 10$)					ACO with KLS ($num_{ant} = m$)				
		best	avg	worst	Q	Time[s]	best	avg	worst	Q	Time[s]
16	19	**19**	**19.00**	**19**	**1.00**	0.001	**19**	**19.00**	**19**	**1.00**	< 0.001
64	76	**76**	77.15	81	**0.98**	3.765	**76**	**76.95**	**80**	**0.98**	3.400
256	307	330	340.75	**356**	**0.90**	300.000	**327**	**340.45**	357	**0.90**	223.283
1024	1228	**1591**	**1679.15**	**1790**	**0.73**	3411.722	1752	1833.85	1937	0.66	3600.000

表 2: 他の近似解法との比較

n	Opt	ACO with KLS					MKLS				
		best	avg	worst	Q	Time[s]	best	avg	worst	Q	Time[s]
16	19	**19**	**19.00**	**19**	**1.00**	0.001	**19**	**19.00**	**19**	**1.00**	< 0.001
64	76	**76**	77.15	81	**0.98**	3.765	80	83.10	87	0.91	4.878
256	307	**330**	**340.75**	**356**	**0.90**	300.000	399	412.50	426	0.74	132.450
1024	1228	1591	1679.15	1790	0.73	3411.722	2243	2334.85	2436	0.52	1529.891

n	Opt	IKLS					MA				
		best	avg	worst	Q	Time[s]	best	avg	worst	Q	Time[s]
16	19	**19**	**19.00**	**19**	**1.00**	< 0.001	**19**	**19.00**	**19**	**1.00**	< 0.001
64	76	**76**	79.55	83	0.95	3.175	**76**	**76.85**	**80**	**0.98**	3.988
256	307	333	348.25	366	0.88	216.373	331	345.45	357	0.89	228.807
1024	1228	1577	1689.85	1849	0.72	2862.324	**1549**	**1631.75**	**1704**	**0.75**	3436.890

との比較を行う．MKLS は，ランダムに初期解を生成し，その解に対して KLS を適用する処理を繰り返すものである．IKLS と MA は，我々が NPP に対する強力な近似解法として文献 [3] [9] においてそれぞれ提案しているものである．

4.1. 実験設定

本実験で扱ったベンチマーク問題例，実験方法，および比較する各解法に設定した具体的なパラメータ値について説明する．ベンチマーク問題例は [11] にて公開しており，問題サイズ 16(= 4 × 4), 64(= 8 × 8), 256(= 16 × 16), 1024(= 32 × 32) のそれぞれに対して 20 例題づつある．これらのベンチマーク問題例は，文献 [5] と同じ方法で生成している．生成法については以下のとおりである．

まず，高負荷トラヒック t_H と低負荷トラヒック t_B の 2 種類のトラヒックを定義し，適当なノード配置 σ を与える．σ は実行可能解であれば良いが，$\sigma = \{0, \ldots, n-1\}$ とする．各ノードは隣接する 4 つのノードと直接通信可能であるので，ネットワーク全体では $4n$ 本の論理的な隣接リンクが存在する．これら $4n$ 本の隣接リンクから a 割のリンク $(4n \times a)$ をランダムに選択して，リンクに対応するトラヒック行列 T に高負荷トラヒックを割り当てる．このとき，ある一つのノードから他のノードへ割り当てる高負荷トラヒックの最大本数 L_{max} を 4 以下に限定することで最適解が既知の問題例を生成することができる．こうして得られたトラヒック行列 T は，σ のとき最適ノード配置となり，最適解が既知の問題例が生成できる．

本実験ではこの各問題サイズの 20 例題に対して一回ずつ提案法と比較解法を適用した．本実験におけるすべての解法の終了条件は，あらかじめ設定した許容計算時間に達したときとし，各問題サイズ 16，64，256，1024 のそれぞれに対して 1 秒，10 秒，300 秒，3600 秒と設定した．

ACO のパラメータであるフェロモンの初期値 $\tau_0 = 1$ とし，フェロモンの下限値 $\tau_{\min} = 1$，フェロモン情報の重要度 $\alpha = 1$，問題領域固有の情報の重要度 $\beta = 1$，フェロモンの蒸発率の情報 $\rho = 0.1$ とした．アリエージェントの数 num_{ant} については後述する．MA のパラメータは文献 [9] と同様に設定し，個体数は 10 とした．なお IKLS はパラメータ値の設定を必要としない解法である．実験環境は HP Workstation xw4300(CPU: Pentium4 3.4GHz，RAM: 4GB)，解法は C 言語によってすべてコード化し，使用コンパイラは最適化オプション-O3 を追加した gcc version 4.4.1 である．

4.2. 実験 I の結果

提案法 ACO と MKLS との比較結果を表 1 に示す．表中の n は問題サイズ，$f(\sigma_{opt})$ は既知の最適解の評価値を表している．best は各解法によって得られた解の最良解の評価値，avg は得られた解の評価値の平均，worst は得られた解の最悪解の評価値，Q は得られた解の精度を $f(\sigma_{opt})/f(\sigma)$ の式で算出している．Time は最良解を発見するまでの平均時間 (秒) を表している．太字で示した数値は各解法の結果を比較して他の解法と同等かより良い結果である．

表 1 の ACO with KLS （$num_{ant} = 10$）は num_{ant} が 10 の場合の結果である．ACO with KLS（$num_{ant} = m$）は，num_{ant} を m とした場合の結果である．問題サイズが 16〜256 のものに関してはどちらもほぼ同等の解を算出している．しかしながら問題サイズが 1024 の場合に関しては ACO($num_{ant} = 10$) が優れた結果を示している．特に avg の項目では num_{ant} が 10 と m の場合での評価値の差が 154.7 も開いている．この様な結果となった理由としては，問題サイズ 1024 での ACO（$num_{ant} = m$）が十分に反復を行っていないことが挙げられる．$num_{ant} = 10$ とした場合が探索全体を通して平均で 100.1 回の反復を行っているのに対して，$num_{ant} = m$ の場合は 25.8 回の反復しか行っていないことを観測した．num_{ant} が増加すると，一回の反復にかかる時間が増加することによりフェロモンの更

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.

新回数が少なくなり，良質なフェロモン情報が蓄積されるまでに長い時間を要することが考えられる．この結果より，高速化処理を施し短時間で良質なフェロモン情報の蓄積が可能になれば，ACO は大規模な問題例に対してもさらに良好な解の算出が可能になると考えられる．

4.3. 実験 II の結果

MKLS と IKLS，MA との比較結果を表 2 に示す．表 2 の上段左の ACO with KLS は提案法の $num_{ant} = 10$ とした場合の結果を表している．続いて上段右には MKLS の結果，下段左は IKLS，そして下段右が MA の結果である．

まず上段に示している提案法と MKLS の比較に着目する．すべての問題例サイズにおいて，提案法 ACO with KLS が best，avg，worst の項目で MKLS よりも良い結果を示した．提案法 ACO with KLS と MKLS との差異は ACO の構成要素であるアントサーチ，フェロモン情報の更新，問題領域固有の情報，Restart 処理の 4 つの要素の有無である．この結果から，ACO 内のアントサーチによる解の再構成においてフェロモン情報と問題領域固有の情報を元に確率的にノードの選択を行うことにより，より良好な解に向けて効率的な探索が行われていると考えられる．

次に提案法と表 2 下段の IKLS，MA との比較について述べる．問題サイズが最小の 16 の結果に着目すると，比較的短時間にすべての解法で最適解を算出することを確認した．また，問題サイズ 64 の結果の場合は，提案法は MA には若干劣るものの，IKLS との比較では best，avg，worst すべての項目で同等以上の解を算出した．問題サイズが 256 の結果では提案法は他の手法と比較してすべての項目において優れた解を算出している．問題サイズが 1024 の場合に関しては提案法は MA より若干劣る解を算出したが，IKLS と比較した場合，平均的に優れた解を算出することを観測した．この様な結果となった理由としては，提案法と他の IKLS および MA との差異である，フェロモン情報と問題領域固有の情報にもとづく確率的なノード選択が有効に働いたためと考えられる．

5. むすび

ノード配置問題（NPP）に対してはこれまでに，代表的なメタ戦略アルゴリズム [8] である遺伝的アルゴリズムやタブサーチ [5]，アニーリング法 [10] が提案されている．また我々は，反復局所探索法 [3] と Memetic アルゴリズム [9] を提案している．その他，代表的なメタ戦略としてアント最適化法 (Ant Colony Optimization, ACO) [1] があるが，NPP に対する適用例は報告されていなかった．

本論文では，NPP に対して ACO に k-swap 局所探索法 [3] をハイブリッドしたメタ戦略アルゴリズムを提案した．提案法の性能を評価するために NPP に対する強力な近似解法である反復局所探索法（IKLS）と Memetic アルゴリズム (MA) との比較実験を行った．提案法の性能は，小規模な問題例に対しては，IKLS，MA と同等かより良い結果を算出し，特に問題サイズ 256 においては IKLS，MA よりも優れた結果を算出した．また大規模な問題例に対しては，MA には若干劣るものの，IKLS よりは平均的に優れた結果を算出することを確認した．今後の課題としては，大規模な問題例に対するさらなる解精度の向上などが挙げられる．

参考文献

[1] M. Dorigo, L.M. Gambardella, “Ant Colony System: A Cooperative Leaning Approach to the Traveling Salesman Problem,” IEEE Transactions on Evolutionary Computation， vol.1, no.1, pp.53–66, 1997.

[2] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

[3] K. Katayama, H. Yamashita, H. Narihisa “Variable Depth Search and Iterated Local Search for the Node Placement Problem in Multihop WDM Lightwave Networks,” Proc. of 2007 IEEE Congress on Evolutionary Computation (CEC-2007), pp.3508–3515, 2007.

[4] 嘉藤　学, 河北隆二, 永持　仁, 尾屋祐二, “格子型光波ネットワークの再構成アルゴリズムについて,” 信学論 (B–I), vol.J80–B–I, no.10, pp.709–718, Oct. 1997.

[5] 嘉藤　学, 尾屋祐二, “光波ネットワークの網再構成–メタヒューリスティックスの適用–,” 信学論 (B), vol.J82–B, no.12, pp.2225–2237, Dec. 1999.

[6] B. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” Bell System Technical Journal, vol.49, pp.291–307, 1970.

[7] S. Lin and B. Kernighan, “An effective heuristic algorithm for the traveling salesman problem,” Operations Research, vol.21, pp.498–516, 1973.

[8] 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.

[9] 山下浩司, 片山謙吾, 南原英生, 成久洋之, “ノード配置問題に対する Memetic アルゴリズム,” 進化計算研究会 進化計算シンポジウム 2007 講演論文集, 北海道 洞爺湖, pp.99–102, Dec. 27–28, 2007.

[10] 米津政隆, 舩曳信生, 木谷友哉, 横平徳美, 中西透, 東野輝夫, “双方向マンハッタンストリートネットワークのノード配置問題に対する階層型近似アルゴリズムの提案,” 信学論 (D–I), vol.J86–D–I, no.2, pp.99–107, Feb. 2003.

[11] Benchmark Instances of the Node Placement Problem (NPP), “http://k2x.ice.ous.ac.jp/˜katayama/bench/npp/”

Copyright © 2011 by Information Processing Society of Japan and
The Instiute of Electronics, Information and Communication Engineers
All rights reserved.