一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CS2020-56(2020-11)

ACO に基づく最短ホップルーティングアルゴリズム

関　心†　　中里　秀則†

†早稲田大学大学院基幹理工学研究科　〒169-0072 東京都新宿区大久保 3-14-9
E-mail:　†guanxin@toki.waseda.jp,　††nakazato@waseda.jp

あらまし　ACO（Ant colony Optimization）ルーティングアルゴリズムは，経路を決定するための有効な手法であるが，収束速度が遅いことや局所最適に陥る可能性があるなどの欠点がある．ノード数が多い場合，この ACO の欠点がパフォーマンスに大きいな影響を与える．遅い収束速度と局所最適の問題を解決するために，本論文は ACO に基づく最短ホップルーティングアルゴリズムを提案し，パラメータの影響を分析した．実際のネットワークトポロジに適用した場合に，この提案アルゴリズムは収束速度をある程度で改善し，局所最適を回避して，ネットワークの最短ホップ経路を取得できる．

キーワード　ACO，収束速度，局所最適，最短ホップ経路

ACO-based Shortest Hop Routing Algorithm

Guan Xin†　　and　Hidenori Nakazato†

†Graduate School of Information Engineering, Waseda University　Okubo 3-14-9, Sinjuku-ku, Tokyo, 169-0072 Japan
E-mail:　†guanxin@toki.waseda.jp,　††nakazato@waseda.jp

Abstract　ACO (Ant Colony Optimization) routing algorithm is an effective routing algorithm. However, this algorithm has some shortcomings, such as slow convergence speed and local optimal solution. In the case of a large number of nodes, the shortcomings of the ACO will have a great impact on performance. In order to solve the problems of slow convergence speed and local optimal solution, this paper proposes a shortest hop routing algorithm based on the ACO algorithm, and analyzes the influence of parameters on the algorithm. For real network topologies, this algorithm can improve the convergence speed to a certain extent, avoid local optimal solutions, and obtain the shortest hop path of the network.

Keywords　ACO, convergence speed, local optimal solution, shortest hop path

1. はじめに

近年，新しいネットワークの仕組みに対する研究が進んでいる．ICN（Information Centric Network）はその一つである．現在 ICN に対して，SoCCeR（Services over Content Centric Routing）[8]などのルーティング方法が提案された．SoCCeR は，ACO (Ant Colony Optimization)を利用した手法である[1]．ACO では，アリ（ant）が動く時に経路にフェロモン（pheromone）を残す．アリが移動を繰り返した後，フェロモンの最も多い経路が最短経路となるというルーティングアルゴリズムである．ICN において，特定のコンテンツに対して，まず Interest ant をランダムに送信し，データ（あるいは Data ant）が帰ってきたら，その遅延時間やホップ数をもとに，フェロモン値を計算し，フェロモン値の高い経路に(通常の Interest ant でない)Interest を転送する．しかし，ACO には，収束速度が遅いことや局所最適などの欠点がある．ノード数が多い場合，ACO の欠点がパフォーマンスに大きいな影響を与える可能性がある．

以下，2.で ACO に関する研究について述べ，3.で提案手法について説明する．4. シミュレーションとその結果の分析を行い，5.で結論を述べる．

2. ACO に関する研究

Marco Dorigo ら[1]は最初の ACO アルゴリズムを提案し，巡回セールスマン問題（TSP 問題）の解決方法を求めた．彼らは ACO の各パラメータを分析して，良い結果を得た．同時に他の ACO 方法も提案した．しかし，これらの方法は収束速度が遅く，局所最適の問題もあった．

Chen Jia ら[3]は狼群アルゴリズムを加える動的 ACO アルゴリズムを提案した．このアルゴリズムはある程度で収束速度を改善したが，単純な環境の場合にしか適用できない．ノード数が多い複雑なネットワークに対して，パフォーマンスの品質はまだ改善する必要がある．

Quan Qi ら[6]はフェロモンの補償に基づく ACO 最適転送アルゴリズム（CP-ACO）を提案した．このア

ルゴリズムは，SoCCeR の収束速度と局所最適の問題を解決した．しかし，使われたトポロジはただ 7 つのノードを含んで，また実際のネットワークのノードの制限（CPU 速度やテーブルルックアップ速度など）を考えなかった．

Zhan Shichang ら[9]は ACO の性能，機能に対して，パラメータの最適選択に関わる検討を行った．この研究の結果は[1]の中の結論に類似しており，また巡回セールスマン問題に対する最適化のみ検討した．

Zhang Guo ら[11]は分布均一性と転送戦略の組み合わせに基づく ACO アルゴリズム（RED-ACO）を提案した．このアルゴリズムによるコンテンツ要求の遅延とネットワーク負荷の削減のパフォーマンスは，SoCCeR より優れていた．

高橋 [12]は ACO と遺伝アルゴリズムの性能を比較して，Oliver30（TSP に使用されるベンチマーク）を用いて実験した．そして，ACO とシミュレーテッドアニーリングアルゴリズムを組み合わせて，収束速度を改善した．

ACO における遅い収束速度と局所最適の問題を解決するために，本論文では ACO に基づく最短ホップルーティングアルゴリズムを提案し，パラメータの影響を分析した．実際のネットワークトポロジの場合に，このアルゴリズムは収束速度をある程度で改善し，局所最適を回避して，ネットワークの最短ホップ経路を取得できる．

3. ACO に基づく最短ホップルーティングアルゴリズム（H_ACO）

ACO における遅い収束速度と局所最適を改善するには，フェロモンの更新方法が最も重要だと考える．最短ホップ経路を得るために，ホップ数を中心に，フェロモンの更新式を修正して，最短ホップルーティングアルゴリズムを提案する．以下に提案アルゴリズムを示す．

（1）初期化

ノード（ルータ）の数 n，アリ（interest ant，以下はアリとする）の数 m，ジェネレーション g，パラメータ ρ，α，β と Q，初期のフェロモンの値 c，また転移確率 p_{ij} を設定する．転移確率 p_{ij} は，ノード i からノード j へアリが移動する確率である．ノード j がノード i の隣接ノードである場合，p_{ij} は 0 以外の値をもつ．転移確率の初期値は，ノードのインタフェイス数（以下，フェイス数）を基に，各フェイスに同じ確率を与える．ここで，各フェイスは次のノードのノード番号を記録している．また，ジェネレーション g は，経路探索を行う一纏まりの期間を表し，各ジェネレーションの初めに n 匹のアリを発生させ，その全てのアリが目的地に辿り着くと一つのジェネレーションが終わる．ジェネレーション g は経路を学習する回数に相当する．

（2）アリの移動

各ジェネレーションの最初にアリの出生地と目的地のノード（の番号）をランダムに発生する．アリは各時刻 t に移動し，ノード間の移動には単位時間，すなわち，時間 1 かかるものとする．ノード数は n であるので，アリがその目的地のノードに到着するには最大で時間 n 必要になる．

以下で時刻 t でのアリの動きを例として説明する．まずアリは自分の現在の状態をチェックする．もしアリが目的地に到着した状態や袋小路に陥った状態である場合，このアリは移動しない（できない）．もしこのアリがこれらの状態ではなかったら，次の行動をする．もしアリが現在いるノードの隣接ノードに目的地のノードがあれば，直接に目的地のノードに移動する．もし目的地のノードに直接到着する経路がなかったら，まず今到着しているノードのフェイスをチェックする．これまでまだアリが訪れていない隣接ノードの内，転移確率 p_{ij} が最大のノードに移動する．転移確率が最大のフェイスが複数ある場合，ランダムにこれらのノードの中の一つに移動する．もし今到着しているノードからはどのフェイスを辿っても他の全てのノードに行けない状態（ループあるいは袋小路）になっている場合，このアリは動かない．

（3）フェロモン値の計算と転移確率の更新

一つのジェネレーションが終わると，フェロモンの値を計算して，転移確率を更新する．まず，$\Delta\tau_{ij}^{k}$ を計算する．$\Delta\tau_{ij}^{k}$ は目的地に到着した第 k 番目のアリがノード i とノード j を結ぶリンクを通過した場合に，そのリンクに残したフェロモン値である．$\Delta\tau_{ij}^{k}$ は以下の式で計算する．

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{H_{ij}^{*}}, & \text{第}k\text{番目のアリが辺}(i,\ j)\text{を通過して} \\ & \text{目的地に到着した．} \\ 0, & \text{上記以外．} \end{cases} \tag{1}$$

ここで，Q は定数で，H_{ij}^{*} は第 k 番目のアリが目的地に到着した経路で通過したホップ数である．

$\Delta\tau_{ij}^{k}$ から，$\Delta\tau_{ij}$ を以下により求める．

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

この後，$\Delta\tau_{ij}$ によって，フェロモン値を更新する．$\tau_{ij}(t)$ は時刻 t における辺 $(i,\ j)$ のフェロモン値である．$\tau_{ij}(t)$ は以下により求める．

$$\tau_{ij}(\mathrm{t}) = \rho * \tau_{ij}(t-1) + \Delta\tau_{ij} \quad (3)$$

ここで，ρ は係数で(1-ρ)はフェロモンの蒸発率を示す．また，$\tau_{ij}(0)=c$ である．最後に各ノードのフェイスの転移確率を更新する．転移確率は以下の式で計算する．

$$p_{ij}(\mathrm{t}) = \begin{cases} \dfrac{[\tau_{ij}(\mathrm{t})]^{\alpha} * \left[\dfrac{1}{H_{ij}}\right]^{\beta}}{\sum[\tau_{ik}(\mathrm{t})]^{\alpha} * \left[\dfrac{1}{H_{ik}}\right]^{\beta}}, & \text{リンク}(i,j)\text{が} \\ & \text{存在する．} \\ 0, & \text{上記以外．} \end{cases} \quad (4)$$

ここで，

$$H_{ij} = \begin{cases} \text{ノード}i\text{から}j\text{までのホップ数,ノード}i\text{から} \\ \quad j\text{までの経路が見つけられた場合．} \\ n\text{（ノード数）,上記以外．} \end{cases} \quad (5)$$

（4）アリの状態の再生（次のジェネレーションの準備）

転移確率を更新してから，新しいアリの出生地と目的地のノード（の番号）を与える．フェロモンの値，転移確率，ホップ数をそのままにして，g の最大値まで繰り返す（(3) に戻る）．

4. シミュレーションと結果の分析

4.1.ネットワークトポロジの生成

ネットワークトポロジでシミュレーションするには，Waxman[10]，Salama[7]，Zhou Ling[5]，Yang Li[4]のネットワークトポロジ発生器を参考した．辺が存在する確率は以下の式で計算する．

$$P(u,v) = \beta' * k * \frac{\varepsilon}{n} * \exp(-\frac{d}{L * \alpha'}) \quad (5)$$

α'，β'，d，L は[10]における表記と同様で，α'と β' はパラメータ，d はノード間の距離，L は地図の最大距離，k は定数，ε は制御因子，n はノード数である．具体的には，α'=0.3，β'=0.1，k=25，ε=4 とした．もし一部分のノードが分離して孤立した場合，それを次数の最も小さいノードと接続させる．

4.2. アルゴリズムのパフォーマンス

アルゴリズムのパフォーマンスを，以下に述べるいくつの観点から評価する．

まず,ベターカウンター(BC)とオーバヘッド(OH)について評価する．BC は、各ジェネレーションでアリが新しい経路あるいは既知の経路よりホップ数が短い経路を見つけた回数であり、アルゴリズムが経路を見つける能力を表す．OH は各ジェネレーションでアリがノードを訪れた回数の積算値である。OH が小さいということはと,アリが通過したノード数が少なく,最短ホップ数経路を見つける速度が速いことを示す．BC と OH をもとに,H_ACO の最適パラメータを探す．

次に，ネットワークトポロジを発生してから，フロイドアルゴリズム（Floyd–Warshall algorithm）[2]で各ノードから他の全てのノードへの最も短い経路を探して，その最短ホップ経路を求める．ここで，隣接ノード間の距離は 1 ホップである． H_ACO アルゴリズムで求めた経路のホップ数と，最短ホップ経路のホップ数を比較して,一致したら,それを記録する．この後，[1]による ACO アルゴリズムについて，同じパラメータの条件でシミュレーションをして,H_ACO の結果と比較する．

4.3.パラメータの選択

H_ACO に対する最適パラメータを探す．図 1 と図 2 はアリの数 m とパフォーマンスの関係を示す（n=30，ρ=0.5，Q=10，c=1，α=1，β=2）．図 1 と図 2 によって，m の最適値は m=30=n とする．同じように，パラメータ ρ，Q，初期のフェロモン値 c（Q=1 と Q=10 でシミュレーションした），α と β 両方とパフォーマンスの関係についてシミュレーションを行い，シミュレーションの条件とその結果から求めた各パラメータの最適値（本評価で利用する設定値）を表 1 に示す．

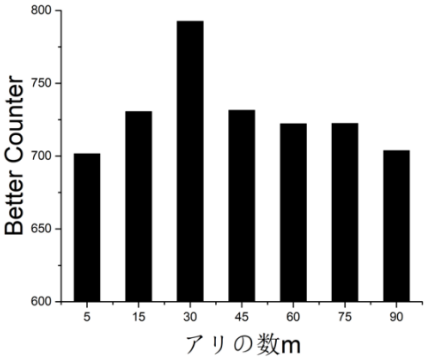

図 1　アリの数 m と BC の関係

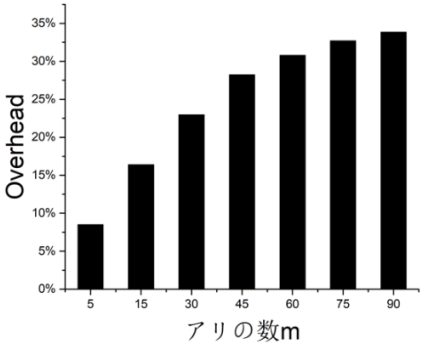

図 2　アリの数 m と OH の関係

表 1　最適パラメータシミュレーションの条件と結果

条件 / パラメータ	m	ρ	Q	c	α	β	設定値
ρ	30	/	10	1.0	1	2	0.6
Q	30	0.6	/	1.0	1	2	1
c	30	0.6	1	/	1	2	1
			10				1
α	30	0.6	1	1.0	/	2	1
β	30	0.6	1	1.0	1	/	2

（ここで，n=m=30 とする.）

以下はそれぞれのパラメータについて説明する.

アリの数 m はノード数と同じ時，パフォーマンスは最も良い．これは[1]の中の結論と同じである．BC から見ると，m が小さ過ぎる場合，各ジェネレーションで残したフェロモンの量が少なすぎて，良い経路を見つけにくいことがある．一方 m が大き過ぎる場合，最初に残したフェロモンの量が多く，局所最適が問題になってしまう可能性がある.また,m が大きくなると，通過するノード数が増加して，OH も増加する．適当なパラメータ ρ を設定すると，ちょうどいいフェロモンを残して，更に多く良い経路を見つけ，局所最適に陥らない．良い経路を見つけられれば，通過したノード数が減少して，OH も減少する．ρ は一定の範囲でパフォーマンスは良く，これは[3]の中の結果と合致する．ここで，最も BC を大きくできる 0.6 と設定することとした．Q は定数として設定する．Q は一つジェネレーションの中で残すフェロモンの総量と考えることができる． [1]と[3]では Q は影響を与えないと述べられているが，このシミュレーションによれば，BC には影響がないが，Q を増加すると，OH が増加する．その故，Q を 1 と設定する．c はフェロモンの初期値なので，Q は c に影響を与えるかどうかについてもシミュレーションにより確認する．c を大きくすると最初のフェロモンが多く，局所最適になってしまう可能性がある．また，Q の値は c に対して，BC にあまり影響はないが，Q が大きくなると，OH も増加する．α と β はそれぞれ範囲があるので，別々にシミュレーションを実施し，パフォーマンスによって，α=1 と β=2 とすることにした．これらの結果によって，パフォーマンスを比較する上で使用するパラメータ値を選んだ．具体的には m=ノード数 n，ρ=0.6，Q=1，c=1，α=1，β=2.
以上の様にパラメータを設定することにより，H_ACO により更に良い経路を見つけることができる.

4.4 H_ACO と ACO のパフォーマンス比較

H_ACO と ACO のパフォーマンスを比較して，H_ACO が収束を早めることと局所最適の課題を解決するかどうかを検証する．ここで, H_ACO と ACO を同じパラメータでシミュレーションする．具体的には n=m=30，ρ=0.6，Q=1，c=1，α=1，β=2 である．結果は以下の図 3 と図 4 に示す.

図 3 によれば，H_ACO は同じ g に対して BC は ACO より大きな値となっている．その原因は，フェロモンの更新式の修正により，収束速度が増えるため，経路を見つける能力が上がるためである．図 4 に示す「最短経路発見率」はアルゴリズムが見つけた経路が最短経路である経路の総数を各アルゴリズムが見つけた経路の総数で割った値である．図 4 によって，g が十分大きいとき，H_ACO の最短経路発見率はほぼ 100%になる．一方，ACO はおよそ 80%である．この結果は，H_ACO は ACO より局所最適を回避できることを示す．OH について，両方式はあまり差がない.

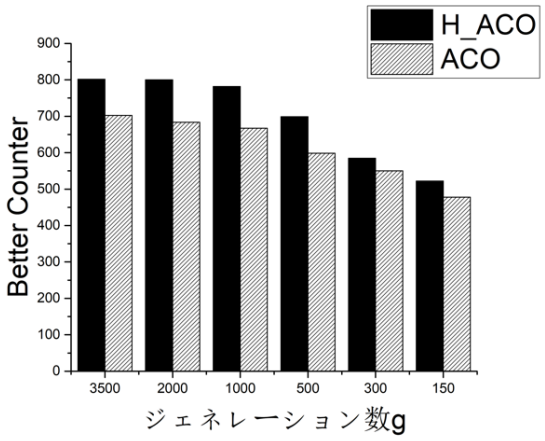

図 3　ジェネレーション g と BC の関係

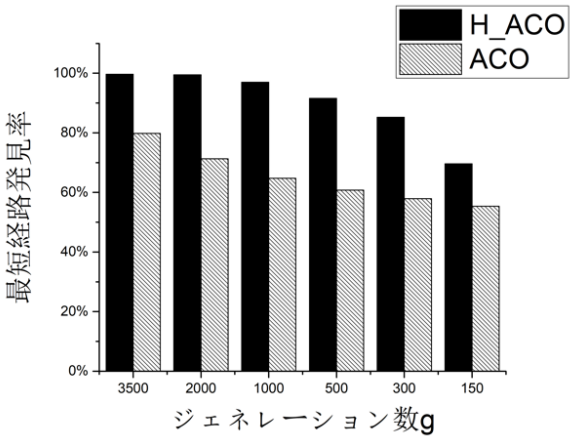

図 4　ジェネレーション g と最短経路発見率の関係

図 5 と図 6 はノード数を増加させた時の H_ACO と ACO のパフォーマンスである．ここで，各ノード数の場合に，経路探索に用いたジェネレーション数 g を表

表 1　ノード数に対して用いたジェネレーション数

ノード数 *n*	ジェネレーション数 *g*
30	1000
50	1500
100	4500
200	12000
300	25000

2 に示す．

ノード数が増えると，H_ACO の BC と最短経路発見率は ACO より高い．しかし，ノード数の増加とともに，H_ACO と ACO の最短経路発見率が下がる．その原因は，ノード数が大きくなると，両方式ともにある程度局所最適になる可能性が増えるためである．ただし，H_ACO は局所最適に陥る可能性は ACO の場合より少ないということができる．

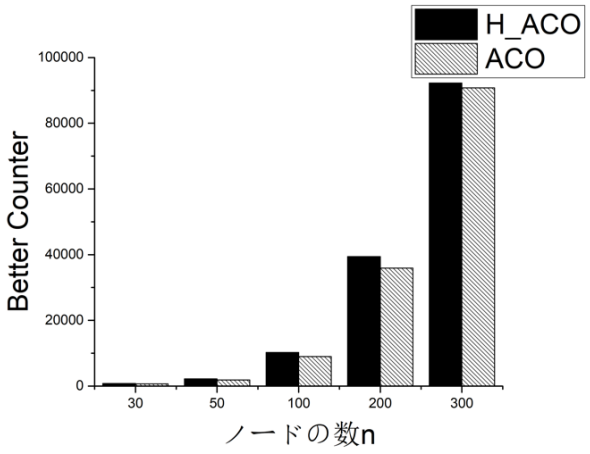

図 5　ノード数 *n* と BC の関係

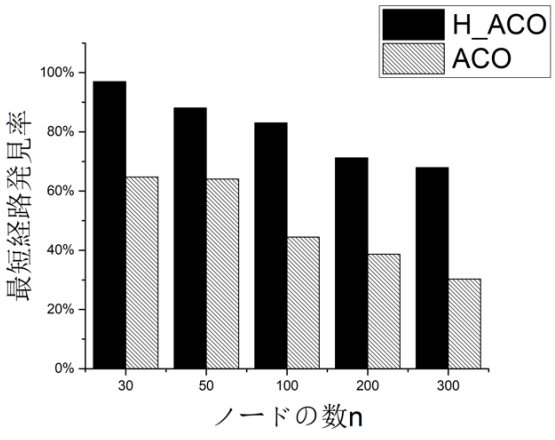

図 6　ノード数 *n* と最短経路発見率の関係

ここで，更にパフォーマンスを明確にするために，経路発見失敗率を計算した．「経路発見失敗率」はアルゴリズムで発見できなかった経路の数を経路の総数で割った値である．図 7 にシミュレーションの結果を示す．ノード数が増えると，H_ACO と ACO の経路発見失敗率も増えるが，H_ACO の方が低い．以上により，H_ACO の収束速度が速いことと局所最適を回避することが示された．

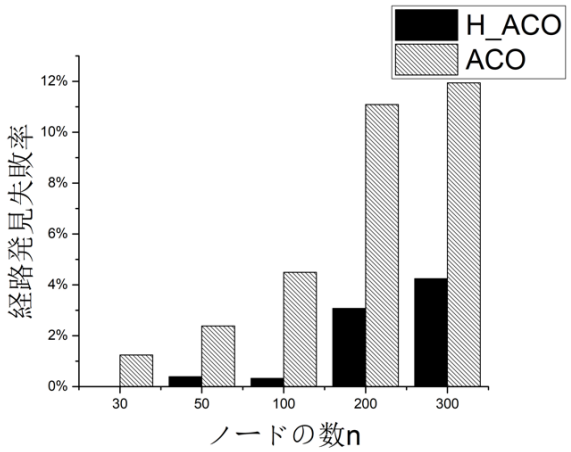

図 7　ノード数 *n* と経路発見失敗率の関係

5. 結論

本論文は収束速度が遅く，局所最適という ACO アルゴリズムが持つ問題を改善するために，ACO に基づく最短ホップルーティングアルゴリズム（H_ACO）を提案した． H_ACO と ACO のパフォーマンスを比較した結果，H_ACO は ACO より収束速度が速く，局所最適もある程度回避できて，ネットワークの最短ホップ経路を取得できることが示された．H_ACO アルゴリズムでは，ACO よりもパフォーマンスを向上させることができたが，ノード数が大きい場合には収束速度と局所最適の問題がまだ十分に解決できているとはいえない．フェロモンの更新式以外の点について，さらに改善できる可能性があり，今後検討を継続していく予定である．

文　献

[1] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 26, No. 1, pp. 29–41, Feb 1996.

[2] Robert W. Floyd. Algorithm 97: Shortest path. Commun. ACM, Vol. 5, No. 6, p. 345, June 1962.

[3] Chen Jia, You Xiaoming, Liu Sheng, and Li Juan. Research on dynamic hierarchical ant optimization algorithm and its application. Application Research of Computers, No. 2, pp. 380–384, 2019.

[4] Yang Li. Simulation modeling of network topology. Journal of Hubei University of Education, Vol. 25, No. 8, August 2008.

[5] Zhou Ling. Design and implement of random network topology using waxman-salama model. Journal of Hunan Institute of Science and Technology, 2008.

[6] Quan Qi. Research on optimization strategy of routing and forwarding in content center network. Master's thesis, Chongqing University, 2018.

[7] H. F. Salama, D. S. Reeves, and Y. Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, pp. 332–345, April 1997.

[8] Shashank Shanbhag, Nico Schwan, Ivica Rimac, and Matteo Varvello. SoCCeR: Services over content-centric routing. In Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ICN '11, pp. 62–67, New York, NY, USA, 2011. ACM.

[9] Zhan Shichang, Xu Zuo, and Wu Jun. The optimal selection on the parameters of the ant colony algorithm. BULLETIN OF SCIENCE AND TECHNOLOGY, No. 5, pp. 381–386, 2003. (in Chinese).

[10] B. M. Waxman. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, pp. 1617–1622, Dec 1988.

[11] G.-Y Zhang, B. Tang, J.-G Sun, and J.-N Li. Ant colony routing strategy based on distribution uniformity degree for content centric network. Tongxin Xuebao/Journal on Communications, Vol. 36, 06 2015.

[12] 高橋良英. 遺伝的アルゴリズムとアーントコロニー最適化手法による巡回セールスマン問題の解法. 計測自動制御学会東北支部第236 回研究集会, 資料番号 236-4, June 2007.