

予測交通量に基づくアントコロニー最適化法 による時間依存 TSP の解法

落合純一[†] 狩野均[†]

本論文では、時間依存 TSP (TDTSP) を対象に、予測交通量を考慮したアントコロニー最適化法を提案する。TDTSP は、都市間の旅行時間が時々刻々と変化するタイプの TSP である。予測交通量は都市間の旅行時間の変化を予測したものととして探索開始前に計算し、この予測交通量に基づいて探索を行う。筆者らは以前に、静的環境での TSP を対象に、MAX-MIN Ant System の探索効率を向上させる手法を提案した。最近の研究で、予測交通量を定式化し、この手法で予測交通量を扱えるようにした。また、TSP のベンチマークから TDTSP の問題を作成する方法についても述べる。評価実験は、都市数 51 から 318 のベンチマークを用いて行った。その結果、MMAS と比べて解の精度を落とすことなく探索速度が向上したことを確認した。

Solving Time-Dependent Traveling Salesman Problems using Ant Colony Optimization Based on Predicted Traffic

Junichi Ochiai[†] and Hitoshi Kanoh[†]

In this paper, we propose an ant colony optimization based on the predicted traffic for time-dependent traveling salesman problems (TDTSP), where the travel time between cities changes with time. Prediction values required for searching is assumed to be given in advance. We previously proposed a method to improve the search rate of MAX-MIN Ant System for static TSPs. In the current work, the method is extended so that the predicted travel time can be handled and formalized in detail. We also present a method of generating a TDTSP to use in evaluating the proposed method. Experimental results using benchmark problems with 51 to 318 cities suggested that the proposed method is better than the conventional MMAS in the rate of search.

1. はじめに

アントコロニー最適化法 (ACO) は、蟻の採餌行動をモデル化したメタヒューリスティクスであり、様々な組合せ最適化問題に適用されている[1]-[3]。ACO では、複数の蟻がフェロモン行列を基にして確率的に解を作成し、作成された複数の解の中で良いものの情報をフェロモン行列にフィードバックさせる。そして、更新されたフェロモン行列を基にして確率的に解を作成することで探索が進む。よって、フェロモン行列は探索領域の情報を集めたものであり、フェロモン行列の扱いは ACO の性能に大きく影響を与える[1]。

現在までに、ACO の性能向上に関する様々な論文が発表されており[1][2]、多くのものは目的関数に時間要素が含まれていない静的な問題を対象としている。しかし、現実問題に適用する場合、目的関数に時間要素が含まれる動的な問題を解かなければならないことが多々ある。そこで、本論文では時間依存巡回セールスマン問題 (TDTSP) を対象とする。TDTSP とは、都市間の旅行時間が時々刻々と変化するタイプの TSP であり、現実の交通量の変化を表している[3]。TDTSP を解く場合の一般的な方法は、都市間の旅行時間が変化するごとに再探索を行うものである[4]-[6]。しかし、この方法で見つかる解は最良解とは限らない。また、都市間の旅行時間の更新間隔が短い場合、以前の探索の情報が有益であるとは限らない。

本論文では、ACO をベースとして予測交通量を考慮する解法を提案する。筆者らは以前に、TSP を対象とした ACO の手法を提案した[7]。この手法は、最も性能が良い ACO の 1 つとされている MAX-MIN Ant System (MMAS) [8] を高速化したものである。その基本方針は、前もって局所最適解を複数用意し、その局所最適解を用いて初期のフェロモン行列に偏りを持たせることで、最適解に近い解を高速に求めることである。よって、TDTSP を解く解法として適している。予測交通量に基づく問題の解法として、現実世界の時間依存配送計画問題を対象に、ACO をベースとしたもの[9]や進化戦略をベースとしたもの[10]が提案されているが、本論文では TDTSP に特化した解法を提案する。

以下では、まず対象問題について述べる。ここでは、一般的な TSP と TDTSP について述べ、TSP のベンチマークから TDTSP を作成する方法についても述べる。次に、ベースとなる MMAS とその高速化の処理について述べる。次に、提案手法について述べる。最後に、都市数 51 から 318 のベンチマークを用いて、MMAS と提案手法を比較した結果について述べる。

[†] 筑波大学
University of Tsukuba

2. 関連研究

2.1 巡回セールスマン問題 (TSP)

TSP[11]は完全グラフ $G=(N,E)$ として表現できる. ここで, N は頂点集合であり, E は辺集合である. TSP では頂点を都市, 辺を経路として扱う. 全ての経路には距離が与えられており, 都市 i, j 間の経路 $(i, j) \in E$ の距離を d_{ij} とする. TSP の目的は, 各頂点を 1 回だけ通る最短の巡回路を求めることである. 本研究では, TSP のベンチマークとして TSPLIB[12]の 51 都市から 318 都市の問題を対象に実験を行った.

2.2 時間依存巡回セールスマン問題 (TDTSP)

以下では, 巡回路の旅行時間を総旅行時間と記し, 都市間の旅行時間を単に旅行時間と記す. TDTSP は, 旅行時間が時々刻々と変化するタイプの TSP である. 都市 i, j 間の旅行時間を t_{ij} とすると, 交通量の変化は t_{ij} の変動で表すことができる. 本研究では, 以下の式で定義されている旅行時間の変動を用いた.

$$t_{ij}(n_{up}(T)) = t_{ij}(n_{up}(T)-1) \times (1 + C'_f \times rand)$$

$$C'_f = \begin{cases} C_f & \text{if } rand \geq 0 \\ \frac{C_f}{1+C_f} & \text{otherwise} \end{cases}$$

$$n_{up}(T) = \left\lceil \frac{T}{\Delta T} \right\rceil$$

$$t_{ij}(0) = d_{ij}$$

ここで, T は現在時刻, $t_{ij}(n_{up}(T))$ は都市 i を出発する時刻が T の場合の都市 i, j 間の旅行時間, C_f は旅行時間の変動係数, ΔT は旅行時間の更新間隔である. 本研究では, $\{t_{ij}(0), t_{ij}(1), \dots\}$ を予測交通量として探索前に求め, 予測交通量から総旅行時間を計算した.

2.3 MAX-MIN Ant System (MMAS)

まず, 一般的な ACO のアルゴリズム[2]を図 1 に示す. ACO では, 初期化の後, 3 つの主な処理を探索の終了条件を満たすまで繰り返す. まず, 複数の蟻による解の作成を行う. 次に, 作成された解に対して局所探索を行う. 最後に, 作成された解を基にフェロモン行列の更新を行う.

次に, MMAS の蟻による解の作成について述べる. 複数の蟻はそれぞれランダムに初期都市に配置され, 確率的に都市を選択して移動し, 初期都市に戻る. 蟻 k が都市 i にいるとき, 次に移動する都市として j を選ぶ確率 p_{ij}^k は以下の式で定義されている.

Procedure ACO ()

```

Set parameters;
Initialize pheromone trails;
while (terminal condition not met) {
    Construct ant solutions;
    Apply local search; /* option */
    Update pheromone;
}
    
```

図 1 一般的な ACO のアルゴリズム

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N^k \\ 0 & \text{otherwise} \end{cases}$$

ここで, N^k は蟻 k の未訪問都市の集合, τ_{ij} は都市 i, j 間のフェロモンの値, α はフェロモンの重み, β は距離の重みである. η_{ij} はヒューリスティクス値と呼ばれ問題ごとに異なる値である. 距離が短い巡回路を求める TSP の場合は都市間の距離の逆数, TDTSP の場合は旅行時間の逆数となる.

次に, MMAS のフェロモンの更新について述べる. フェロモンの更新は以下の式で定義されている.

$$\tau_{ij} \leftarrow [(1-\rho)\tau_{ij} + \Delta\tau_{ij}^{\text{best}}]_{\tau_{\min}}^{\tau_{\max}} \quad (1)$$

$$[x]_b^a = \begin{cases} a & \text{if } x > \tau_{\max} \\ b & \text{if } x < \tau_{\min} \\ x & \text{otherwise} \end{cases}$$

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \frac{1}{f(s^{\text{best}})} & \text{if } (i, j) \in s^{\text{best}} \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_{\max} = \frac{1}{\rho \times f(s^{\text{best}})} \quad (2)$$

$$\tau_{\min} = \frac{1 - \sqrt[n]{p_{\text{best}}}}{\left(\frac{n}{2} - 1\right) \times (1 - \sqrt[n]{p_{\text{best}}})} \times \tau_{\max} \quad (3)$$

ここで、 ρ はフェロモンの蒸発率、 s^{best} はフェロモンの更新に用いる解、 $f(s^{\text{best}})$ は s^{best} の目的関数値である。 s^{best} は一律に決まらず、ステップごとの最良解や探索開始時からの最良解を用いる。 τ_{\max} はフェロモンの最大値、 τ_{\min} はフェロモンの最小値、 n は都市数である。 p_{best} は、最適解の経路のフェロモンの値が τ_{\max} 、それ以外の経路のフェロモンの値が τ_{\min} と仮定し、 η_{ij} を無視した場合、蟻が作成する解が最適解である確率である。 p_{best} の値は 0.05 が良く用いられ、本研究でも 0.05 を用いた。

2.4 局所最適解を用いたフェロモン行列の初期化

筆者らは、MMAS の高速化のために、初期のフェロモン行列に偏りを持たせる手法を提案した[7]。この手法では、フェロモン行列の初期化の前に Nearest Neighbor 法 (NN 法) と 2-opt により複数の局所最適解を求め、フェロモン行列を一律の値で初期化した後、求めた局所最適解に含まれている経路に対するフェロモンの値を増やす。NN 法と 2-opt により作成した局所最適解の集合には、最適解の経路がほぼ含まれているため、効率良く探索領域を狭めることができ、探索効率が向上した。

3. 提案手法

3.1 総旅行時間の計算方法

総旅行時間の計算例を図 2 に示し、計算式を以下で定義する。

$$T(s) = \sum_{i=1}^n t_{N_{i-1}N_i}(n_{\text{up}}(T_{i-1}))$$

$$T_i = \begin{cases} 0 & \text{if } i = 0 \\ T_{i-1} + t_{N_{i-1}N_i}(n_{\text{up}}(T_{i-1})) & \text{otherwise} \end{cases}$$

都市	N_0	\rightarrow	N_1	\rightarrow	...	\rightarrow	N_{n-1}	\rightarrow	$N_n = N_0$
都市間の距離			$d_{N_0N_1}$		$d_{N_1N_2}$		$d_{N_{n-2}N_{n-1}}$		$d_{N_{n-1}N_n}$
都市間の旅行時間			$t_{N_0N_1}(n_{\text{up}}(T_0)) = d_{N_0N_1}$		$t_{N_1N_2}(n_{\text{up}}(T_1))$		$t_{N_{n-2}N_{n-1}}(n_{\text{up}}(T_{n-2}))$		$t_{N_{n-1}N_n}(n_{\text{up}}(T_{n-1}))$
N_0 を出発してからの経過時間	$T_0 = 0$		T_1				T_{n-1}		$T_n = T(S)$

図 2 巡回路の旅行時間の計算例

ここで、 N_0 は最初に出発する都市、 N_i は i 番目に訪問する都市、 T_i は N_0 を出発してからの経過時間である。総旅行時間を計算するとき、同一の巡回路でも最初に出発する都市が異なると旅行時間も異なる。よって、 N_0 は固定するものとし、評価実験ではベンチマークの 1 番目の都市を N_0 とした。

3.2 Nearest Neighbor 法 (NN 法) による初期解の作成

本研究では最初に出発する都市は固定しているため、NN 法で解を作成すると 1 つの解しか作成されない。そこで、最初に出発する都市の次に訪問する都市をランダムに選択するものとし、 $(n-1)$ 通りの解が存在するように変更した。提案手法で用いた NN 法のアルゴリズムを図 3 に示す。

```

Procedure NN (second_city)
    Sp = (N_0 second_city);
    N' = N - {N_0, second_city};
    i = second - city;
    T = t_{N_0}(n_{up}(0));
    while (N' is not empty) {
        k = arg min_{j \in N'} t_{ij}(n_{up}(T));
        Sp ← Sp ⊗ k; /* k is added to Sp */
        N' ← N' - {k};
        T ← T + t_{ik}(n_{up}(T));
        i = k;
    }
    Sp ← Sp ⊗ N_0;
    return Sp as a solution;
    
```

図 3 提案手法で用いた NN 法のアルゴリズム

3.3 フェロモン行列の初期化

MMAS では、フェロモン行列の初期化の前に NN 法で初期解を求め、初期解からフェロモン行列の初期値を求めるのが一般的である。MMAS のフェロモン行列の初期値 τ_0 (MMAS) として以下の式を用いた。

$$\tau_0(\text{MMAS}) = \frac{n-1}{\rho \sum_{k=1, k \neq N_0}^n T(S_k^{\text{NN}})}$$

ここで、 S_k^{NN} は k 番目の NN 法で作成した初期解である。

提案手法では、一律の値でフェロモン行列の初期化を行わず、NN 法で作成した初期解を用いてフェロモン行列に偏りを持たせる。図 4 に提案手法のフェロモン行列の初期化のアルゴリズムを示す。

```

Procedure Initialize_pheromone()
  for (k = 1; k ≤ n; ++k) {
    if (k ≠ N0) {
      S0k = NN(k);
      Improve S0k using 2-opt; /* option */
    }
    Let τij ← τ0 ∀(i, j) ∈ E; /* see (4) */
    Update τij individually using (5);
  }

```

図 4 提案手法のフェロモン行列の初期化

$$\tau_{ij} \leftarrow \tau_0 = \frac{n-1}{\rho \sum_{k=1, k \neq N_0}^n T(S_{0k})} \quad (4)$$

$$\tau_{ij} \leftarrow (1-r)\tau_{ij} + \frac{r}{n-1} \sum_{k=1, k \neq N_0}^n \delta\tau_{ij}^k \quad (5)$$

$$\delta\tau_{ij}^k = \begin{cases} \frac{1}{T(S_{0k})} & \text{if } (i, j) \in S_{0k} \\ 0 & \text{otherwise} \end{cases}$$

ここで、 S_{0k} は k 番目の NN 法で作成した初期解、 r は範囲 $[0.0, 1.0]$ の初期解の重みである。 $r=0$ の場合、初期解によるフェロモンの偏りを与えないため、フェロモン行列の初期化は MMAS と似たものとなる。 S_{0k} を 2-opt で局所探索した場合、一般的に $T(S_{0k}) < T(S_k^{\text{NN}})$ となる。よって、 $r=0$ と MMAS の違いは、一律にフェロモン行列を初期化する初期値のみである。一方、 $r=1$ の場合、初期解に含まれている経路のみにフェロモンの値を増やし、初期解に含まれていない経路のフェロモンは 0 に初期化する。

3.4 提案手法のアルゴリズム

図 5 に提案手法のアルゴリズムを示す。

```

Procedure Main()
  Input TDTSP;
  Set parameters;
  Initialize_pheromone();
  while (terminal condition not met) {
    for (k = 1; k ≤ n-1; ++k)
      construct solution Sk using (6);
    Sib = arg min_{k=1, n-1} T(Sk);
    if (T(Sib) < T(Sgb)) Sib = Sgb;
    Calculate τmax ant τmin using (2) and (3);
    Update pheromone using (1);
  }
  Output final solution Sgb

```

図 5 提案手法のアルゴリズム

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^r [\eta_{ij}(T)]^p}{\sum_{l \in N^k} [\tau_{il}]^r [\eta_{il}(T)]^p} & \text{if } j \in N^k \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\eta_{ij}(T) = \frac{1}{t_{ij}(n_{\text{up}}(T))}$$

目的は旅行時間が短い巡回路を求めることであるため、時刻 T において都市 i からの旅行時間が短い都市ほど高い確率で選ばれるように、ヒューリスティクス値を旅行時間の逆数とした。TSP を対象とした場合、MMAS では 1 回のステップで解を作成する蟻の数は、一般的には n が用いられる。本研究では TDTSP を対象としており、初期都市を固定しているため、MMAS と提案手法の両方で蟻の数は $n-1$ とした。

4. 評価実験

4.1 実験方法

TDTSP の例題として、TSPLIB[12]で公開されている eil51, eil75, eil101, kroA200, lin318 の 5 つの問題を選び、2.2 節で示した方法を用いて作成した。これらの問題は小さいサイズの問題であるが、現実世界を考え、セールスマンが 1 日に訪問できる顧客の数は高々 200 ほどであることから、上記の問題が評価実験に適していると考えた。表 1 に実験に用いた問題の都市数、静的環境での最適解、旅行時間の更新間隔、旅行時間の変更回数を示す。日本では最新の道路交通情報を 5 分間隔で得ることができ、各問題の静的環境での最適解の旅行時間から、eil51, eil76, eil101 の旅行時間の単位は分と仮定し、旅行時間の更新間隔は 5 (分) とした。同様に、kroA200 と lin318 の旅行時間の単位は秒と仮定し、旅行時間の更新間隔は 300 (秒) とした。表 1 を見ると、旅行時間の更新回数は 86 回から 140 回であるため、旅行時間が頻繁に変動していることがわかる。

実験で用いたパラメータの値を表 2 に示す。 C_f 以外のパラメータの値は予備実験により求めた。また、本論文で示す実験は、蟻が解を作成した後の局所探索や、初期解を求めるときの 2-opt を用いずに行った。これは、MMAS と提案手法のみの比較を目的とし、局所探索の影響を取り除くためである。よって、MMAS と提案手法の計算時間の違いはほとんどない。以降で示す実験結果は、30 回繰り返した結果の平均値である。

4.2 実験結果

1000 ステップ後の最良解の精度の比較を表 3 に、ステップごとの最良解の精度の比較を図 6 に示す。表 3 から、1000 ステップ後の最良解に関しては、提案手法と MMAS の平均値には有意差が見られなかった。また、旅行時間が変動しているため、静的環境の最適解よりも総旅行時間が短い解が得られている。図 6 を見ると、提案手法の方が MMAS よりも明らかに探索の効率が良い。よって、解の精度を落とすこと無く、探索速度の向上に成功していることが確認できる。

表 1 実験に用いた問題

問題名	都市数	最適解	旅行時間の更新間隔	旅行時間の変更回数
eil51	51	426	5	86
eil76	76	538	5	108
eil101	101	629	5	129
kroA200	200	29368	300	98
lin318	318	42029	300	140

表 2 実験条件

パラメータ	記号	値
旅行時間の変動係数	C_f	0.1
フェロモンの重み	α	1.0
移動時間の重み	β	5.0
フェロモンの蒸発率	ρ	0.02
初期解の重み	r	0.9
最大ステップ数	-	1000
蟻の数	-	$n-1$

表 3 1000 ステップ後の最良解の精度の比較

問題	MMAS		提案手法	
	平均値	標準偏差	平均値	標準偏差
eil51	386	5	386	5
eil76	488	5	486	6
eil101	590	5	591	6
kroA200	26966	257	26870	218
lin318	39114	396	39042	301

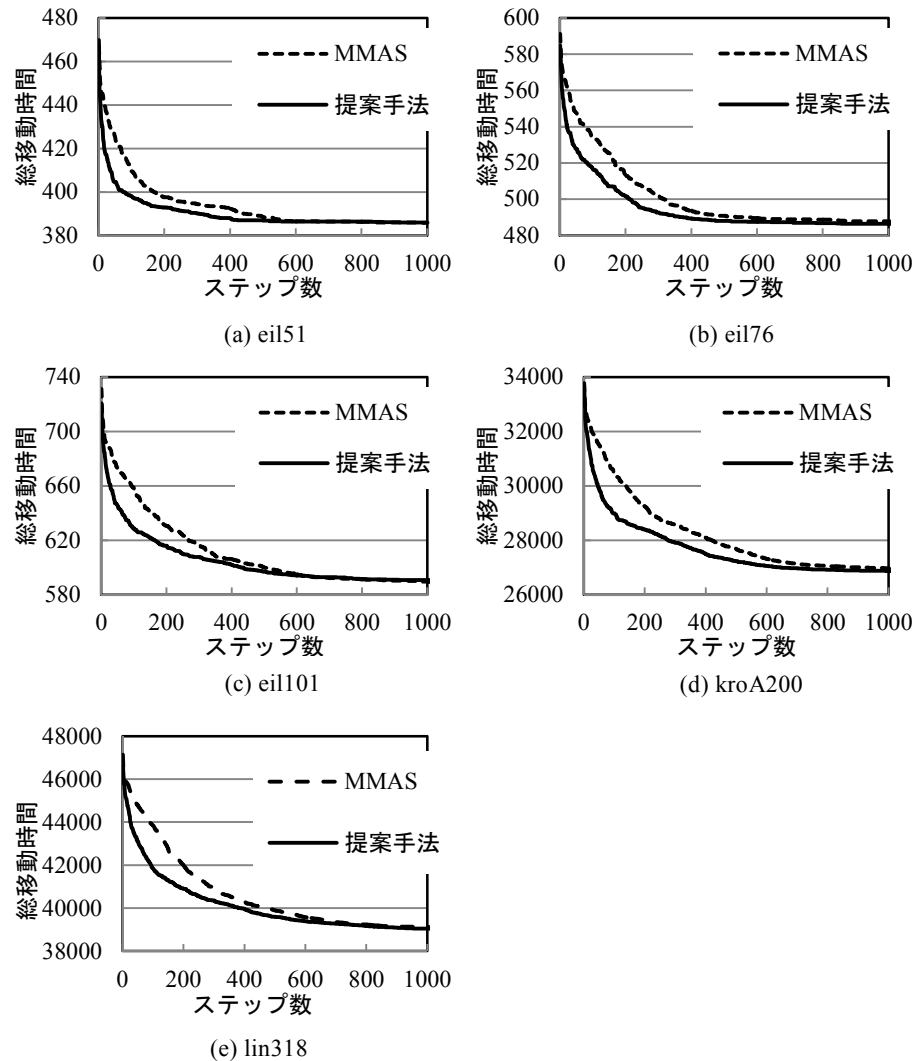


図 6 ステップごとの最良解の精度の比較

5. おわりに

本論文では、TDTSPを対象に、予測交通量に基づくACOを提案した。評価実験は、TSPLIBの都市数51から318の問題から、TDTSPの問題を作成して行った。その結果、提案手法はMMASと比べて、最良解の精度を落とすことなく、探索速度が向上することを確認した。

今後の予定として、本論文の実験結果は5つの問題に対するものであるため、他の問題に対する実験も必要であると考えられる。次の段階として、旅行時間が変動するごとに再探索する手法を組み合わせることが考えられる。この手法は、旅行時間の更新間隔が短い場合や、予測交通量の精度が悪い場合に有効であると考えられる。次に、現実問題に対して適用し、提案手法の有効性を示すことが重要である。また、本研究ではTDTSPを対象としたが、提案手法の考え方は他の時間依存組合せ最適化問題に対しても適用できる。

参考文献

- 1) M. Dorigo and T. Stützle: Ant Colony Optimization. MIT Press, 2004.
- 2) M. Dorigo, M. Birattari and T. Stützle: Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. IEEE Computational Intelligence Magazine, Vol. 1, No. 4, pp. 28-39, IEEE Computational Intelligence Society, 2006.
- 3) M. Dorigo and T. Stützle: Ant Colony Optimization: Overview and Recent Advances. Handbook of Metaheuristics, International Series in Operations Research & Management Science, Vol. 146, pp. 227-263, Springer, 2010.
- 4) M. Mavrouniotis and S. Yang: Ant Colony Optimization with Immigrants Schemes in Dynamic Environments. Lecture Notes in Computer Science, Vol. 6239, pp. 371-380, Springer, 2010.
- 5) M. Guntsch and M. Middendorf: Pheromone Modification Strategies for Ant Algorithms Applied to Dynamic TSP. Lecture Notes in Computer Science, Vol. 2037, pp. 213-222, Springer, 2001.
- 6) C. J. Eyckelhof and M. Snoek: Ant Systems for a Dynamic TSP: Ant Caught in a Traffic Jam. Lecture Notes in Computer Science, Vol. 2463, pp. 88-99, Springer, 2002.
- 7) 亀田陽介, 狩野均: 局所最適解をフェロモン初期化に用いたアントコロニー最適化法によるTSPの解法. 情報処理学会数理モデル化と問題解決研究会, MPS-63-28, pp. 109-112, 2007.
- 8) T. Stützle and H. H. Hoos: MAX-MIN Ant System. Future Generation Computer System, Vol. 16, No. 8, pp. 889-914, Elsevier, 2000.
- 9) A. V. Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli and L. M. Gambardella: Time Dependent Vehicle Routing Problem with a Multi Ant Colony System. European Journal of Operational Research, Vol. 185, Issue 3, 1174-1191, ELSEVIER, 2008.
- 10) H. Kanoh and S. Tsukahara: Solving Real-World Vehicle Routing Problems with Time Windows using Virus Evolution Strategy. International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol. 12, No. 4, pp. 283-294, IOS Press, 2010.
- 11) G. Reinelt: The Traveling Salesman: Computational Solution for TSP Applications. Lecture Notes in Computer Science, Vol. 840, Springer, 1994.
- 12) Traveling Salesman Problem (TSPLIB)
<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>