European Journal of Operational Research 217 (2012) 459–469

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

# An Ant Colony Optimisation algorithm for solving the asymmetric traffic assignment problem

Luca D'Acierno [a,*], Mariano Gallo [b], Bruno Montella [a]

[a] Department of Transportation Engineering, 'Federico II' University of Naples, via Claudio 21, 80125 Naples, Italy
[b] Department of Engineering, University of Sannio, piazza Roma 21, 82100 Benevento, Italy

ARTICLE INFO

*Article history:*
Received 8 October 2010
Accepted 14 September 2011
Available online 24 September 2011

*Keywords:*
Traffic
Ant Colony Optimisation
Signal settings design
Stochastic traffic assignment

ABSTRACT

In this paper we propose an Ant Colony Optimisation (ACO) algorithm for defining the signal settings on urban networks following a local approach. This consists in optimising the signal settings of each intersection of an urban network as a function only of traffic flows at the accesses to the same intersection, taking account of the effects of signal settings on costs and on user route choices. This problem, also known as Local Optimisation of Signal Settings (LOSS), has been widely studied in the literature and can be formulated as an asymmetric assignment problem. The proposed ACO algorithm is based on two kinds of behaviour of artificial ants which allow the LOSS problem to be solved: traditional behaviour based on the response to pheromones for simulating user route choice, and innovative behaviour based on the pressure of an ant stream for solving the signal setting definition problem. Our results on real-scale networks show that the proposed approach allows the solution to be obtained in less time but with the same accuracy as in traditional MSA (Method of Successive Averages) approaches.



## 1. Introduction

Generally, travel time of private system users can be split into three parts:

- running time, that is the time spent by vehicles travelling along roads;
- waiting time (delay) at intersections, that is the time spent crossing intersections;
- searching time for parking, that is the time spent at destination looking for a parking space.

In urban contexts, the second term can make a substantial contribution to total travel time. Therefore, the effective optimisation of signalised intersections can significantly improve performance of urban road systems.

The problem of optimising signal settings, generally indicated as the *Signal Setting Design Problem* (*SSDP*), can be considered as a particular case of the more general *Network Design Problem* (*NDP*) where the design variables are only the signal setting parameters (number of phases, cycle length, effective green times, etc.) while all other supply variables (such as widths or lane numbers) are fixed and invariable. In this kind of problem, link flows are descriptive variables, i.e. the analyst cannot directly modify them, but can influence them by changing design variables.

In the literature (see for instance Marcotte, 1983; Cantarella et al., 1991; Cantarella and Sforza, 1995; Cascetta et al., 2006; Cascetta, 2009) two kinds of approaches can be identified for solving the SSDP: the global approach and the local approach. In the first case, the problem, known as *Global Optimisation of Signal Settings* (*GOSS*), consists of calculating signal parameters by minimising an objective function, that is:

$$\hat{\boldsymbol{g}} = \arg_{\boldsymbol{g}} \min Z(\boldsymbol{g}, \boldsymbol{f}^*) \tag{1}$$

subject to:

$$\boldsymbol{f}^* = \boldsymbol{\Lambda}(\boldsymbol{c}(\boldsymbol{g}, \boldsymbol{f}^*)) \tag{2}$$

$$\boldsymbol{g} \in \boldsymbol{S}_g \tag{3}$$

$$\boldsymbol{f}^* \in \boldsymbol{S}_f \tag{4}$$

where $\boldsymbol{g}$ is the vector of signal setting parameters to be optimised; $\hat{\boldsymbol{g}}$ is the optimal value of $\boldsymbol{g}$; $Z$ is the objective function, for instance the total travel time, to be minimised; $\boldsymbol{f}^*$ is the equilibrium vector of link flows to be calculated by means of Eq. (2); $\boldsymbol{\Lambda}$ is the equilibrium assignment function; $\boldsymbol{c}$ is the link cost vector depending on signal setting parameters (vector $\boldsymbol{g}$) and equilibrium flows (vector $\boldsymbol{f}^*$); $\boldsymbol{S}_g$ is the feasibility set of vector $\boldsymbol{g}$; $\boldsymbol{S}_f$ is the feasibility set of vector $\boldsymbol{f}^*$.

The first constraint (Eq. (2)) represents the equilibrium assignment constraint that provides user flows ($\boldsymbol{f}^*$) as a function of link costs (vector $\boldsymbol{c}$) which depend on design variables (vector $\boldsymbol{g}$) and equilibrium flows (vector $\boldsymbol{f}^*$). This constraint is formulated as a

* Corresponding author. Tel.: +39 081 7683947; fax: +39 081 7683946.
*E-mail addresses:* dacierno@unina.it (L. D'Acierno), gallo@unisannio.it (M. Gallo), montella@unina.it (B. Montella).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.09.035

*fixed-point problem* because user choices (and related flows) are generally influenced by network performance (i.e. users tend to choose the least cost alternatives) and transportation system performance is affected by user flows (i.e. an increase in on-road vehicles generally produces an increase in travel times).

According to constraint (3), signal setting parameters have to satisfy some conditions (such as the minimum value assumed by green times or the sum of all effective green times which have to be equal to the effective cycle). Finally, constraint (4) expresses that flows have to belong to feasibility sets that impose flow consistency (for instance, the sum of all incoming flows in a node has to be equal to the sum of all outgoing flows if the node is not a centroid).

A static approach to the GOSS problem is adopted by Sheffi and Powell (1983), Yang and Yagar (1995), Wong and Yang (1997), Chiou (1999) and Cascetta et al. (2006). The GOSS in the case of coordinated intersections is studied by Pillai et al. (1998) and Wey (2000). Moreover, a dynamic approach for the GOSS problem is proposed by Abdelfatah and Mahmassani (1999) and Abu-Lebdeh and Benekohal (2003).

The second approach to the SSDP, known as *Local Optimisation of Signal Settings* (*LOSS*), is based on assuming that signal settings at each intersection depend only on entering flows of the intersection according to a pre-specified control policy, such as the equisaturation method (Webster, 1958), the local delay minimization and Smith's $P_0$ policy (Smith, 1980, 1981), or other isolated intersection optimisation methods (such as those of Chang and Lin, 2000; Wong and Wong, 2003). Hence, the LOSS problem can be formulated as:

$$\hat{\boldsymbol{g}} = \boldsymbol{\Omega}(\boldsymbol{f}^*) \tag{5}$$

subject to:

$$\boldsymbol{f}^* = \boldsymbol{\Lambda}(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}^*)) \tag{6}$$

$$\boldsymbol{g} \in \boldsymbol{S}_g \tag{7}$$

$$\boldsymbol{f}^* \in \boldsymbol{S}_f \tag{8}$$

where $\boldsymbol{\Omega}$ is the control policy which provides signal setting parameters as a function of equilibrium flows. In this case constraints (6)–(8) assume the same meaning as constraints (2)–(4).

Since control policy (5) has to provide a unique value of vector $\boldsymbol{g}$ for each value of equilibrium flow vector $\boldsymbol{f}^*$, the LOSS problem can be reformulated as a *fixed-point problem* indicated also as an *asymmetric equilibrium problem* (Cascetta et al., 2006), that is:

$$\boldsymbol{f}^* = \boldsymbol{\Lambda}(\boldsymbol{c}(\boldsymbol{\Omega}(\boldsymbol{f}^*),\boldsymbol{f}^*)) \tag{9}$$

$$\boldsymbol{f}^* \in \boldsymbol{S}_f \tag{10}$$

The asymmetric equilibrium problem has been extensively studied elsewhere: the static approach is analysed by Allsop (1974), Dafermos (1980), Dafermos (1982), Florian and Spiess (1982), Smith and Van Vuren (1993), Lee and Hazelton (1996) and Cascetta et al. (2006), and the dynamic approach by Han (1996), Hu and Mahmassani (1997) and Lo et al. (2001). Finally, real-time applications of actuated signals are studied by Rakha (1993), Wolshon and Taylor (1999) and Mirchandani and Head (2001).

The LOSS approach vis-à-vis the GOSS has the advantage of solving signal settings optimisation with a significantly lower computing time, leading to a solution that is not significantly worse than that obtained by solving the GOSS problem (see Cascetta et al., 2006). This issue is of great importance for real-scale problems, where LOSS applications are mainly related to two cases: (a) to estimate the total travel time on the network produced at equilibrium by a system of actuated traffic signals; (b) to adopt it within urban network design models and algorithms for the joint calculation of equilibrium traffic flows and local optimal signal settings.

While reducing computing times in case (a) is not very important, since the solution has to be found only once, it is of great importance in case (b) where the solution has to be found many times inside the algorithm for solving the urban network design problem. An example of the urban network design model and algorithm that uses the LOSS approach for estimating traffic flows and signal settings can be found in Gallo et al. (2010), where a classic MSA-FA (Method of Successive Averages with Flow Averages) was adopted for solving the LOSS problem, leading to high computing times on real-scale networks. Indeed, the LOSS problem had been solved 52,735 times, resulting in over 113 hours spent on processing.

The aim of this paper is to provide a solution algorithm based on the *Ant Colony Optimisation* (*ACO*) paradigm for solving the SSDP in the case of the LOSS approach. Indeed, the solution of the asymmetric assignment problem requires more computing time than a classic assignment problem, since the signal settings have to be updated at each iteration and there is a double circular dependence (local optimal signal settings depend on flows that, in turn, depend on costs, that depend on both flows and signal settings). Therefore, we steered our research into ACO-based algorithms which, based on the food source search of ant colonies, have in many cases shown their efficiency in terms of calculation times (an extended overview of ACO-based algorithms can be found in Dorigo and Stützle, 2004).

Although ACO algorithms are generally considered meta-heuristic algorithms, in the literature we can find two classes of them:

– 'sheer' ACO algorithms;
– 'pseudo' ACO algorithms.

In the first class we can find most of the ACO algorithms proposed in the literature. These algorithms are based on the imitation of ant behaviour and can be considered meta-heuristic for solving locally optimisation problems (i.e. these algorithms provide local optimal solutions). The main applications to transportation systems were proposed by Poorzahedy and Abulghasemi (2005) in the case of network design and by de Oliveira and Bazzan (2006) for traffic control.

Recently another class of ACO algorithms (i.e. the 'pseudo' ACO algorithms) have been proposed. These algorithms are not meta-heuristic and their aim is not to solve optimisation problems, but to reformulate other algorithm classes. These 'pseudo' ACO algorithms are considered MSA (Method of Successive Averages) algorithms, which are iterative methods for solving fixed-point problems, where their iterative procedures adopt a pheromone interpretation. The main applications to the transportation system of this second class of ACO algorithms were proposed by D'Acierno et al. (2006) and Mussone and Matteucci (2007) for solving road traffic assignment, and D'Acierno et al. (2010) for public transport assignment. That said, an exhaustive and detailed review of ACO algorithm applications to transportation systems can be found in Teodorovic (2008).

Since the application of ACO theory in the case of (symmetric) assignment problems has been proved to be an 'algorithmic trick' to modify the updating rule of MSA algorithms and hence speed up the calculation of the equilibrium solution (D'Acierno et al., 2006; Mussone and Matteucci, 2007), in this paper we modify the algorithm proposed by D'Acierno et al. (2006) for generating an ACO-based algorithm able to solve the LOSS problem by significantly reducing computing times, which is an important task for managing real-scale networks.

This paper is organised as follows: Section 2 describes the general framework of 'pseudo' ACO algorithms in the case of the traffic assignment problem; in Section 3, an ACO algorithm for solving the LOSS problem is proposed; in Section 4, a comparison among traditional algorithms and the proposed ACO-based algorithm is provided in the case of three real-scale networks; finally, Section 5 summarises conclusions and further research prospects.

## 2. The ACO approach in the traffic assignment problem

As shown in the previous section, a LOSS problem can be formulated by means of a fixed-point problem. Since D'Acierno et al. (2006) have shown that, when the (symmetric) assignment problem is formulated as a fixed point problem, the use of artificial ants (i.e. the Ant Colony Optimisation approach) allows the solution to be obtained in less time but with the same accuracy compared with traditional algorithms, we propose to adopt the ACO approach for solving the LOSS problem.

In order to describe the main features of the ACO approach analytically, a short introduction to the behaviour of real and artificial ants is worth providing. In particular, in the case of real ants the search for food sources is based on the following simple rules:

– each ant provides a pheromone trail along its path;
– each ant follows a path if there is a pheromone trail;
– if there is no pheromone trail, ants choose their paths randomly;
– there is evaporation of pheromone trails, which causes short paths to have a more intense pheromone trail;
– if there is a diversion point (i.e. where different paths start), ants follow the path with the most intense pheromone trail.

Hence, if a path is blocked by an obstacle, ants initially choose their path randomly. Thus, due to evaporation, they choose to follow the path with the most intense pheromone trail, which is the shortest path.

Similarly, artificial ants can be described by these simple rules:

– the probability of choosing a path, indicated as *transition probability*, which depends on the intensity of the pheromone trail and, sometimes, a visibility term which expresses a sort of distance that could affect probability choices, that is:

$$p^t(l|i) = \frac{(\tau_l^t)^\alpha \cdot (\eta_l^t)^\beta}{\sum_{l' \in FS(i)} \left(\tau_{l'}^t\right)^\alpha \cdot \left(\eta_{l'}^t\right)^\beta} \tag{11}$$

where $p^t(l|i)$ is the probability of choosing link $l$, with $l = (i,j)$, at diversion node $i$; $\tau_l^t \left[\tau_{l'}^t\right]$ is the intensity of the pheromone trail on link $l[l']$ at iteration $t$; $\eta_l^t \left[\eta_{l'}^t\right]$ is the visibility term on link $l[l']$ at iteration $t$; $FS(i)$ is the set of links belonging to the forward star of node $i$; $\alpha$ and $\beta$ are model parameters which express the relevance respectively of the pheromone trail and visibility term in ant choices;
– the *pheromone increase* is the rule that expresses the quantity produced by each ant at any iteration, that is:

$$\Delta\tau_l^t = \lambda_l(\boldsymbol{p}^t, \boldsymbol{X}) \tag{12}$$

where $\Delta\tau_l^t$ is the pheromone trail produced by each ant on link $l$ at iteration $t$; $\boldsymbol{p}^t$ is the transition probability matrix at iteration $t$ whose generic element is $p^t(l|i)$; $\boldsymbol{X}$ is the matrix of model parameters; $\lambda_l$ is the function that expresses the increase in the pheromone trail on link $l$ depending on $\boldsymbol{p}^t$ and $\boldsymbol{X}$;
– the *pheromone trail update* is the rule governing the evaporation of the pheromone trail and how the trail is increased by the contribution of each ant. In the literature there are two kinds of formulations for the pheromone trail update, that is:

$$\tau_l^t = (1-\rho) \cdot \tau_l^{(t-1)} + \rho \cdot \Delta\tau_l^t \tag{13}$$

and

$$\tau_l^t = (1-\rho) \cdot \tau_l^{(t-1)} + \Delta\tau_l^t \tag{14}$$

where $\rho$ is the evaporation term.

As shown in the previous section, the analytical formulation of an ACO-based algorithm, described by means of Eqs. (11)–(14), can be used both for formulating meta-heuristic algorithms for solving optimisation problems and reformulating iterative algorithms for solving fixed-point problems. The second option is going to be shown in this section.

Generally, a *traffic assignment problem* can be formulated as a fixed-point problem which can be obtained by combining the supply model with the demand model. The supply model describes transportation systems performance depending on user flows, that is:

$$\boldsymbol{C} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}) + \boldsymbol{C}^{NA} \tag{15}$$

where $\boldsymbol{C}$ is the vector of path generalised costs whose generic element $C_k$ is the path cost related to path $k$; $\boldsymbol{A}$ is the link-path incidence matrix whose generic element $a_{l,k}$ is equal to 1 if link $l$ belongs to path $k$, 0 otherwise; $\boldsymbol{c}$ is the vector of link costs whose generic element $c_l$ is the cost of link $l$; $\boldsymbol{g}$ is the matrix of signal setting parameters whose generic element $g_{l,h}$ is the $h$-th parameter (such as effective green or cycle length) related to link $l$; $\boldsymbol{f}$ is the vector of link flows whose generic element $f_l$ is the flow on link $l$; $\boldsymbol{C}^{NA}$ is the vector of non-additive path costs whose generic element $C_k^{NA}$ is the non-additive path cost of path $k$ (such as: road tolls at motorway entrance/exit points or transit fares).

Likewise, the demand model imitates user choices influenced by transportation system performance, that is:

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{P}(-\boldsymbol{C})\boldsymbol{d} \tag{16}$$

where $\boldsymbol{P}$ is the matrix of path choice probabilities whose generic element $p_{k,od}$ expresses the probability of users travelling between origin–destination pair $od$ choosing path $k$; $\boldsymbol{d}$ is the demand vector whose generic element $d_{od}$ expresses the average number of users travelling between origin–destination pair $od$ in a time unit.

Therefore, the traffic assignment model can be obtained by combining Eq. (15) with (16), that is:

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{P}(-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}) - \boldsymbol{C}^{NA})\boldsymbol{d} \tag{17}$$

It is worth noting that Eq. (17) represents the assignment constraint described by Eq. (2) or equivalently Eq. (6).

The existence and the uniqueness of the (equilibrium) solution of problem (17) may be stated, as shown by Cantarella (1997) and Cascetta (2009), if:

– choice probability functions, $\boldsymbol{P}(-\boldsymbol{C})$, are continuous;
– link cost functions, $\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f})$, are continuous;
– each origin–destination pair is connected (i.e. $I_{od} \neq \Phi\ \forall od$);
– path choice models are expressed by strictly decreasing functions with respect to path generalised costs, that is:

$$\left[\boldsymbol{P}(-\boldsymbol{C}') - \boldsymbol{P}(-\boldsymbol{C}'')\right]^{\mathrm{T}}(\boldsymbol{C}' - \boldsymbol{C}'') < 0 \quad \forall \boldsymbol{C}' \neq \boldsymbol{C}'' \tag{18}$$

– cost functions are expressed by monotone non-decreasing functions with respect to link flows, that is:

$$[\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}') - \boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}'')]^{\mathrm{T}}(\boldsymbol{f}' - \boldsymbol{f}'') \geqslant 0 \quad \forall \boldsymbol{f}' \neq \boldsymbol{f}'' \tag{19}$$

Moreover, several authors (Sheffi and Powell, 1981; Daganzo, 1983; Cantarella, 1997) proposed to solve problem (17) by means of a solution algorithm known as the *Method of Successive Averages* (*MSA*). In particular, two algorithms, a *Flow Averaging* (*MSA-FA*) and a *Cost Averaging* (*MSA-CA*) algorithm, were proposed based respectively on the following sequences:

$$\boldsymbol{f}^{t+1} = \boldsymbol{f}^{t} + (1/t)\cdot(\boldsymbol{f}(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}^{t})) - \boldsymbol{f}^{t}) \text{ with } \boldsymbol{f}^{1} \in \boldsymbol{S_f} \quad (20)$$

$$\boldsymbol{c}^{t+1} = \boldsymbol{c}^{t} + (1/t)\cdot(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}(\boldsymbol{c}^{t})) - \boldsymbol{c}^{t}) \text{ with } \boldsymbol{c}^{1} = \boldsymbol{c}(\boldsymbol{f}^{1});\ \boldsymbol{f}^{1} \in \boldsymbol{S_f} \quad (21)$$

where $\boldsymbol{f}(\cdot)$ is the network loading function described by Eq. (16).

MSA algorithms stop when a stopping criterion is satisfied. In particular, in the case of MSA-FA and MSA-CA the stopping criterion is always:

$$\max_{l}\left|\frac{f_l(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}^{\,t})) - f_l^t}{f_l^t}\right| < \varepsilon_{MSA} \quad (22)$$

where $f_l^t$ is the link flow associated to link $l$ at iteration $t$; $f_l(\cdot)$ is the link flow function which provides the flow of link $l$ by means of Eq. (16); $\varepsilon_{MSA}$ is the threshold used in the stopping criterion of MSA algorithms.

Both MSA algorithms can be used for solving the fixed point problem (17), providing the same results if the existence and uniqueness of the solution can be stated. Therefore, use of one algorithm rather than the other can be addressed only out of computational or implementation reasons.

Eq. (22) should represent the threshold that we accept as the maximum difference between the theoretical solution of problem (17) and the numerical result obtained by implementing a solution algorithm. Since we can solve problem (17) only numerically, we are not able to evaluate a priori the theoretical value of (17). Hence, we may adopt as a threshold the maximum difference between two successive iterations of the same algorithm. Indeed, at each iteration these algorithms generally provide asymptotically a solution ever closer to the theoretical value.

However, it is necessary to fix a threshold value (value $\varepsilon_{MSA}$ in Eq. (22)) which represents a fair trade-off between the accuracy of the solution and the calculation times involved. Moreover, it is worth noting that results of assignment algorithms (variable $\boldsymbol{f}$) are functions, by means of Eq. (17), of the travel demand (variable $\boldsymbol{d}$) which is estimated exogenously. Therefore the accuracy adopted for estimating travel demand has to be kept also in the flow estimation in terms of $\varepsilon_{MSA}$ value.

By means of an extension of Blum's theorem (Blum, 1954), Cantarella (1997) stated the convergence of MSA algorithms described by Eqs. (20) and (21).

On analysing Eqs. (20) and (21), we may note that an assignment algorithm can be formulated as a sequence of network loadings (i.e. implementation of Eq. (16)). In particular, Dial's algorithm (Dial, 1971) is a stochastic network loading algorithm based on the following rules:

- let $Z_{d,i}$ be the minimum cost between node $i$ and destination node $d$;
- a link $l = (i,j)$ is considered for a feasible path, indicated as a *Dial-efficient path*, only if:

$$Z_{d,i} > Z_{d,j} \quad (23)$$

- it is possible to associate a numerical quantity to each node and each link, indicated respectively as node weight and link weight, calculated as:

$$W_{d,d} = 1 \quad (24)$$

$$w_{d,(i,j)} = \begin{cases} \exp(-c_{(i,j)}/\theta)\cdot W_{d,j} & \text{if } Z_{d,i} > Z_{d,j} \\ 0 & \text{if } Z_{d,i} \leqslant Z_{d,j} \end{cases} \quad (25)$$

$$W_{d,i} = \sum_{(i,h)\in FS(i)} w_{d,(i,h)} \quad (26)$$

where $W_{d,i}$ is the node weight associated to node $i$ in the case of destination node $d$; $w_{d,(i,j)}$ is the link weight associated to link $(i,j)$ in the case of destination node $d$; $c_{(i,j)}$ is the link cost associated to link $(i,j)$; $\theta$ is the Gumbel parameter which expresses the perception error in user behaviour which belong to interval $]0;+\infty[$, where 0 represents a perfectly deterministic choice and $+\infty$ represents a completely random choice (details on random utility models can be found in Cascetta, 2009); $FS(i)$ is the set of links belonging to the forward star of node $i$;

- the probability of choosing link $l = (i,j)$ at diversion node $i$ is equal to:

$$p(l|i) = w_{d,(i,j)}/W_{d,i} \quad (27)$$

- the link flow $f_l$ of generic link $l$ can be calculated as:

$$f_{l=(i,j)} = \sum_d e_{d,(i,j)} = \sum_d E_{d,i}\cdot w_{d,(i,j)}/W_{d,i} \quad (28)$$

with:

$$E_{d,i} = \sum_{(h,j)\in BS(j)} e_{d,(h,j)} \quad (29)$$

$$E_{d,o} = d_{od} \quad (30)$$

where $BS(j)$ is the set of links belonging to the backward star of node $j$; $d_{od}$ is the average number of users travelling between origin–destination pair $od$ in a time unit.

Adopting an ACO-based paradigm, D'Acierno et al. (2006) proposed an algorithm for solving the (symmetric) traffic assignment problem based on the following assumption:

- the initial intensity of the pheromone trail on each link $l$, associated to ant colony $od$, indicated as $\tau^0_{od,l}$, is a function of path costs, that is:

$$\tau^0_{od,l} = \sum_{k:l\in k} T^0_{od,k} \quad (31)$$

with:

$$T^0_{od,k} = \begin{cases} \exp\left(-C^0_k/\theta\right) & \text{if } k \in I_{od} \\ 0 & \text{if } k \notin I_{od} \end{cases} \quad (32)$$

where $T^0_{od,k}$ is the initial intensity of the pheromone trail on path $k$; $C^0_k$ is the initial cost of path $k$; $\theta$ is the parameter of the path choice model; $I_{od}$ is the set of all available (or considered) paths that join origin node $o$ with destination node $d$;

- the increase in the pheromone trail, indicated as $\Delta\tau^t_{od,l}$, can be expressed by a function of path costs, that is:

$$\Delta\tau^t_{od,l} = \sum_{k:l\in k} \Delta T^t_{od,k} \quad (33)$$

with:

$$\Delta T^t_{od,k} = \begin{cases} \exp\left(-C^t_k/\theta\right) & \text{if } k \in I_{od} \\ 0 & \text{if } k \notin I_{od} \end{cases} \quad (34)$$

where $\Delta T^t_{od,k}$ is the increase in the pheromone trail at iteration $t$; $C^t_k$ is the path $k$ cost at iteration $t$;

- updating of the pheromone trail can be expressed as:

$$\tau^{t+1}_{od,l} = (1-\rho)\cdot\tau^t_{od,l} + \rho\cdot\Delta\tau^{t+1}_{od,l} \quad (35)$$

where evaporation coefficient $\rho$ is variable and equal to $1/t$ (according to the approach proposed by Li and Gong, 2003).

In particular, D'Acierno et al. (2006) stated the theoretical equivalence of the proposed ACO-based algorithm to the application of an MSA algorithm where the successive averages were applied to the weight of Dial's algorithm (Dial, 1971), that is:

$$w^{t+1}_{d,(i,j)} = w^{t}_{d,(i,j)} + (1/t)\cdot\left(\Delta w^{t+1}_{d,(i,j)} - w^{t}_{d,(i,j)}\right) \quad (36)$$

$$W^{t+1}_{d,i} = W^{t}_{d,i} + (1/t)\cdot\left(\Delta W^{t+1}_{d,i} - W^{t}_{d,i}\right) \quad (37)$$

by fixing $\tau^{t+1}_{od,l} = w^{t+1}_{od,l}$ and $\Delta\tau^{t+1}_{od,l} = \Delta w^{t+1}_{od,l}$.

In this case, the MSA stopping criterion was:

$$\max_l \left|\frac{f_l\left(\boldsymbol{w}(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}^t))\right) - f^t_l}{f^t_l}\right| = \max_l \left|\frac{f_l\left(\boldsymbol{\tau}(\boldsymbol{c}(\boldsymbol{g},\boldsymbol{f}^t))\right) - f^t_l}{f^t_l}\right| < \varepsilon_{MSA} \quad (38)$$

where $\boldsymbol{w}$ is the vector of Dial weights, whose generic element is $w_{d,(i,j)}$; $\boldsymbol{\tau}$ is the vector of intensity of the pheromone trail, whose generic element is $\tau_l$. Finally, D'Acierno et al. (2006) stated the convergence of the proposed algorithm by means of the extension of Blum's theorem (Blum, 1954) provided by Cantarella (1997).

## 3. The proposed assignment algorithm

In order to develop an ACO-based algorithm for solving the LOSS problem, we modified the artificial ants proposed by D'Acierno et al. (2006) by adding the following behaviour: at signalised intersections each ant flow provides a pressure value which is directly proportional to the number of ants which travel in a time unit (flow) and inversely proportional to the 'net' width of their road (i.e. the road width reduced by the space occupied by parked vehicles). Therefore, in a simple case where there is an intersection with two entering one-way roads (as shown in Fig. 1), pressure values are:

$$\begin{cases} Press_1 = f_1/L_1 \\ Press_2 = f_2/L_2 \end{cases} \quad (39)$$

As in the case of two flows with the same priority level (for instance the case of evacuation), the number of elements able to cross the intersection in a time unit is directly proportional to their pressure level. Therefore we may consider that each road has a shutter whose opening times are directly proportional to pressure levels, that is:

$$\begin{cases} \%time_1 = Press_1/(Press_1 + Press_2) \\ \%time_2 = Press_2/(Press_1 + Press_2) \end{cases} \quad (40)$$

In the case of an intersection with four bi-directional roads (as shown in Fig. 2), we may assume that influences between opposite roads can be neglected (the lower the flows of left turns, the more this assumption holds). In this case the pressure value which affects

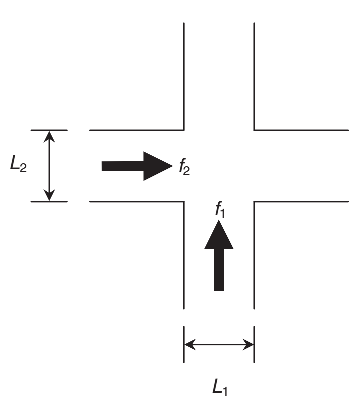



**Fig. 1.** Example of intersection with two entering one-way roads.

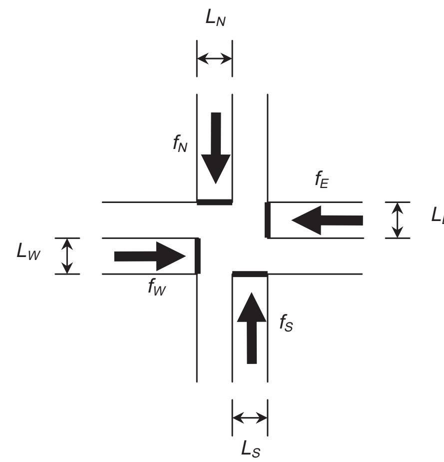



**Fig. 2.** Example of intersection with four bi-directional roads.

crossing phenomena can be calculated by means of the following relation:

$$\begin{cases} Press_1 = \max(Press_N; Press_S) = \max(f_N/L_N; f_S/L_S) \\ Press_2 = \max(Press_E; Press_W) = \max(f_E/L_E; f_W/L_W) \end{cases} \quad (41)$$

while the shutter operations can be described by Eq. (40).

With reference to the behaviour of real road users, it may be stated that shutter operations are equivalent to a traffic light where signal setting parameters are calculated by means of the equisaturation method (Webster, 1958). Indeed, since saturation flows (i.e. the maximum flows which can cross an intersection with the assumption that traffic lights are always green) is directly proportional to the net width of the road, that is:

$$s_i \propto L_i \quad (42)$$

$\%time_i$ terms are directly proportional to saturation rates (where the saturation rate is the ratio between a flow and the related saturation), that is:

$$\begin{cases} \%time_1 = (f_1/s_1)/(f_1/s_1 + f_2/s_2) \\ \%time_2 = (f_2/s_2)/(f_1/s_1 + f_2/s_2) \end{cases} \quad (43)$$

According to the definition of shutter operations, term $\%time_i$ represents the effective green ratio, that is:

$$\begin{cases} \%time_1 = g_1/C_E \\ \%time_2 = g_2/C_E \end{cases} \quad (44)$$

where $g_1$ and $g_2$ are effective green lengths, and $C_E$ is the difference between the cycle length and the total lost time per cycle.

By combining Eq. (43) with Eq. (44), we obtain:

$$\begin{cases} g_1/C_E = (f_1/s_1)/(f_1/s_1 + f_2/s_2) \\ g_2/C_E = (f_2/s_2)/(f_1/s_1 + f_2/s_2) \end{cases} \quad (45)$$

which is the analytical formulation of the equisaturation method for determining green distributions.

The modified behaviour of artificial ants allows simulation of the control policy $\boldsymbol{\Omega}$ for solving the LOSS problem. Likewise, in the case of a non-signalised intersection (i.e. intersection where ants do not have to simulate any control policy) there is no 'interference' among ant flows entering the same intersection, for instance by means of exclusive lanes or 'virtual' ramps.

### *3.1. Formulation of the algorithm for solving the LOSS problem*

The MSA-FA algorithm for solving the LOSS problem proposed by Cascetta et al. (2006) can be described as follows:

Step 0: $k = 0; \boldsymbol{f}^1 = \mathbf{0}$
Step 1: $k = k + 1$
Step 2: $\boldsymbol{g}^k = \boldsymbol{\Omega}(\boldsymbol{f}^k)$
Step 3: $\boldsymbol{c}^k = \boldsymbol{c}(\boldsymbol{g}^k, \boldsymbol{f}^k)$
Step 4: $\boldsymbol{f}^k_{SNL} = \boldsymbol{AP}(-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}^k - \boldsymbol{C}^{NA})\boldsymbol{d}$
Step 5: $\boldsymbol{f}^{k+1} = \left((k-1)\cdot \boldsymbol{f}^k + \boldsymbol{f}^k_{SNL}\right)/k$
Step 6: If $k = 1$ then go to Step 1
Step 7: If $\max_l \left\{ \left| f^k_{SNL,l} - f^k_l \right| / f^k_l \right\} > \varepsilon_{MSA}$ then go to Step 1
Step 8: $\hat{\boldsymbol{g}} = \boldsymbol{\Omega}(\boldsymbol{f}^{k+1})$
Step 9: STOP

Likewise, in order to obtain a more complete analysis of the problem, we propose to develop an MSA-CA algorithm for solving the LOSS problem. The algorithm features can be summarised as follows:

Step 0: $k = 0; \boldsymbol{f}^0 = \mathbf{0}; \boldsymbol{g}^0 = \boldsymbol{\Omega}(\boldsymbol{f}^0); \boldsymbol{c}^1 = \boldsymbol{c}(\boldsymbol{g}^0, \boldsymbol{f}^0)$
Step 1: $k = k + 1$
Step 2: $\boldsymbol{f}^k_{SNL} = \boldsymbol{AP}(-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}^k - \boldsymbol{C}^{NA})\boldsymbol{d}$
Step 3: $\boldsymbol{f}^k = \boldsymbol{f}^k_{SNL}$
Step 4: $\boldsymbol{g}^k = \boldsymbol{\Omega}(\boldsymbol{f}^k)$
Step 5: $\boldsymbol{y}^k = \boldsymbol{c}(\boldsymbol{g}^k, \boldsymbol{f}^k)$
Step 6: $\boldsymbol{c}^{k+1} = ((k-1)\cdot \boldsymbol{c}^k + \boldsymbol{y}^k)/k$
Step 7: If $k = 1$ then go to Step 1
Step 8: If $\max_l \left\{ \left| f^k_{SNL,l} - f^{k-1}_l \right| / f^{k-1}_l \right\} > \varepsilon_{MSA}$ then go to Step 1
Step 9: $\hat{\boldsymbol{g}} = \boldsymbol{g}^k$
Step 10: STOP

Finally, we propose an ACO-based algorithm, indicated as MSA-ACO, for solving the LOSS problem that can be described as follows:

Step 0 $k = \mathbf{0}; \boldsymbol{\tau}^0 = \mathbf{0}; \boldsymbol{f}^1 = \mathbf{0}$
Step 1: $k = k + 1$
Step 2: $\boldsymbol{g}^k = \boldsymbol{\Omega}(\boldsymbol{f}^k)$
Step 3: $\boldsymbol{c}^k = \boldsymbol{c}(\boldsymbol{g}^k, \boldsymbol{f}^k)$
Step 4: $\boldsymbol{\Delta\tau}^k = \tau(\boldsymbol{c}^k)$
Step 5: $\boldsymbol{\tau}^k = ((k-1)\cdot \boldsymbol{\tau}^{k-1} + \boldsymbol{\Delta\tau}^k)/k$
Step 6: $\boldsymbol{f}^k_{SNL} = \boldsymbol{f}(\boldsymbol{\tau}^k)$
Step 7: $\boldsymbol{f}^{k+1} = \boldsymbol{f}^k_{SNL}$
Step 8: If $k = 1$ then go to Step 1
Step 9: If $\max_l \left\{ \left| f^k_{SNL,l} - f^k_l \right| / f^k_l \right\} > \varepsilon_{MSA}$ then go to Step 1
Step 10: $\hat{\boldsymbol{g}} = \boldsymbol{\Omega}(\boldsymbol{f}^{k+1})$
Step 11: STOP

where $\boldsymbol{\Delta\tau}^k = \tau(\boldsymbol{c}^k)$ and $\boldsymbol{f}^{k+1}_{SNL} = \boldsymbol{f}(\boldsymbol{\tau}^k)$ are respectively the synthetic formulations of Eqs. (33) and (28), since $\boldsymbol{\tau}^k = \boldsymbol{w}^k$ and $\boldsymbol{\Delta\tau}^k = \Delta\boldsymbol{w}^k$.

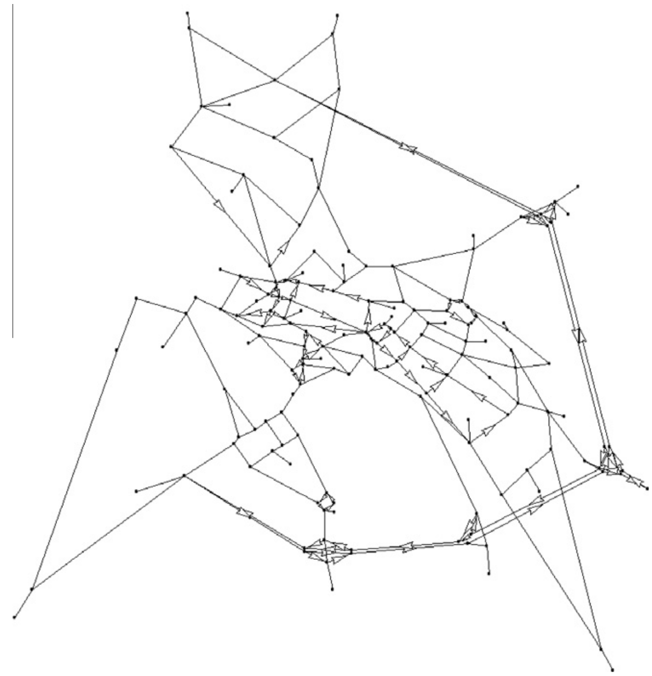


**Fig. 3.** Benevento network (Italy).

As shown by Cascetta et al. (2006), the use of control policy $\boldsymbol{\Omega}$ does not allow us to state the uniqueness of the equilibrium solution of problem (17) and hence related algorithm convergence, even if no multiple equilibrium solutions were found in tests and the convergence was always reached.

## 4. Numerical results

In order to compare the proposed ACO-based algorithm with other MSA algorithms, we applied them on three different real-scale networks: Benevento, Salerno and Naples, whose graphs are reported in Figs. 3–5.

These networks concern three cities in the south of Italy whose main features are reported in Table 1. Note that we consider traffic lights at all junctions where they may be useful for traffic control, even if some of these junctions are not currently signalised. A comparison among link number and peak-hour trips highlights great differences in terms of network density. Indeed, although Benevento and Salerno have almost the same number of trips (about 20,000

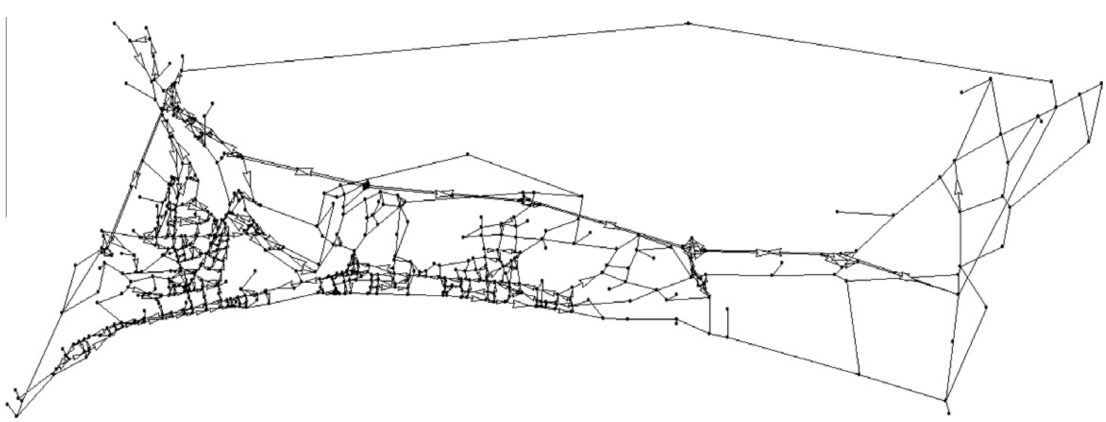


**Fig. 4.** Salerno network (Italy).

**Fig. 5.** Naples network (Italy).

per hour), the Salerno network is 3 times greater in terms of the number of links and nodes and 2 times greater in terms of signalised intersections. Likewise, although Salerno and Naples have almost the same number of links and nodes, the Naples network is twice as large in terms of peak-hour trips.

These three networks were adopted for comparing the proposed ACO algorithm, indicated as MSA-ACO, with the two MSA algorithms: MSA-FA and MSA-CA. Since all algorithms provide the same MSA framework, their complexity is the same. The only difference is the convergence speed related to their averaged parameters, and the main purpose of adopting the ACO approach is to speed up the equilibrium solution calculation. Therefore, we compared them by adopting the following indicators: number of iterations, calculation times and differences in solutions. In particular, the last indicator can be defined as follows:

– 'FA solution error' represents the maximum percentage flow difference between the solution of the MSA-FA algorithm and that of the other algorithms, evaluated by means of:

$$\max_l \left| \frac{f_l^{MSA-FA} - f_l^{MSA-CA}}{f_l^{MSA-FA}} \right| \tag{46}$$

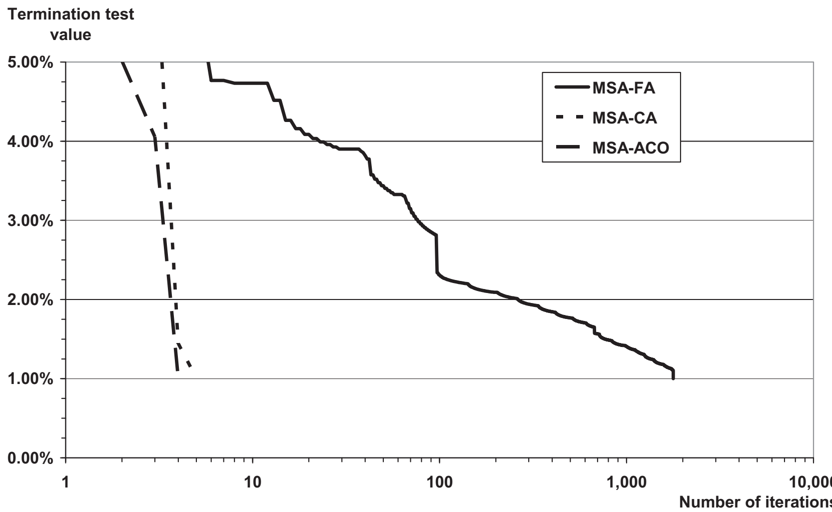
$$\max_l \left| \frac{f_l^{MSA-FA} - f_l^{MSA-ACO}}{f_l^{MSA-FA}} \right| \tag{47}$$

**Table 1**
Network features.

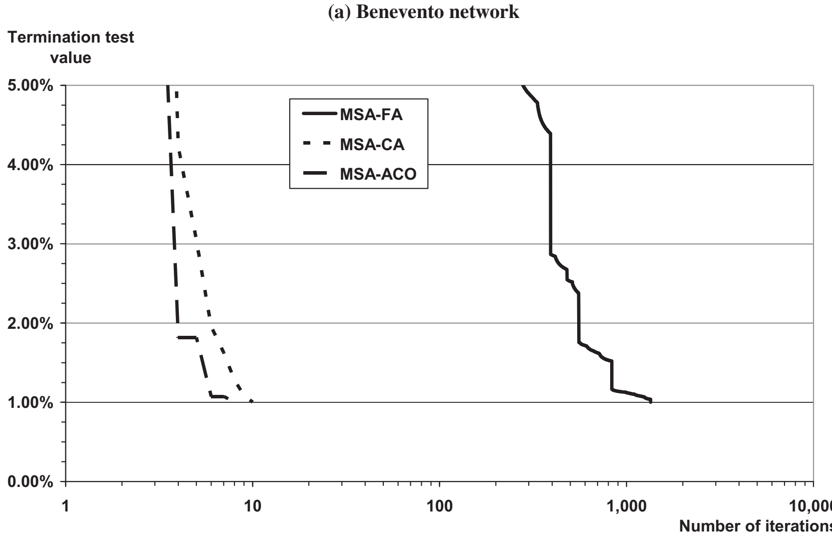
| Network | Number of links | Number of nodes | Number of signalised intersections | Number of centroid nodes | Number of OD pairs | Number of peak-hour trips |
|---|---|---|---|---|---|---|
| Benevento | 382 | 161 | 85 | 36 | 1296 | 17,870 |
| Salerno | 1133 | 529 | 289 | 62 | 3844 | 21,176 |
| Naples | 1532 | 628 | 350 | 70 | 4900 | 49,693 |

**Table 2**
Algorithm performance.

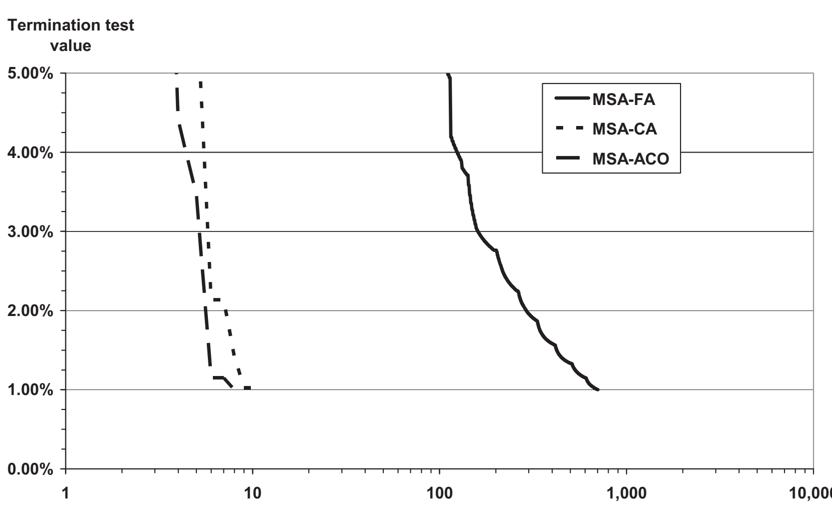
| Network | Algorithm name | Number of iterations | Convergence > 98.00% | Calculation time (s) | FA solution error (%) | CA solution error (%) | ACO solution error (%) |
|---|---|---|---|---|---|---|---|
| Benevento | MSA-FA | 1780 | 887 | 77 | | 3.13 | 2.24 |
| | MSA-CA | 5 | 5 | <1 | 3.23 | | 3.87 |
| | MSA-ACO | 4 | 4 | <1 | 2.29 | 4.03 | |
| Salerno | MSA-FA | 1345 | 208 | 695 | | 4.25 | 6.13 |
| | MSA-CA | 10 | 5 | 6 | 4.44 | | 10.23 |
| | MSA-ACO | 8 | 4 | 5 | 6.31 | 9.28 | |
| Naples | MSA-FA | 703 | 97 | 628 | | 10.04 | 9.58 |
| | MSA-CA | 11 | 5 | 10 | 11.16 | | 7.03 |
| | MSA-ACO | 8 | 4 | 8 | 8.75 | 7.19 | |




(a) Benevento network




(b) Salerno network

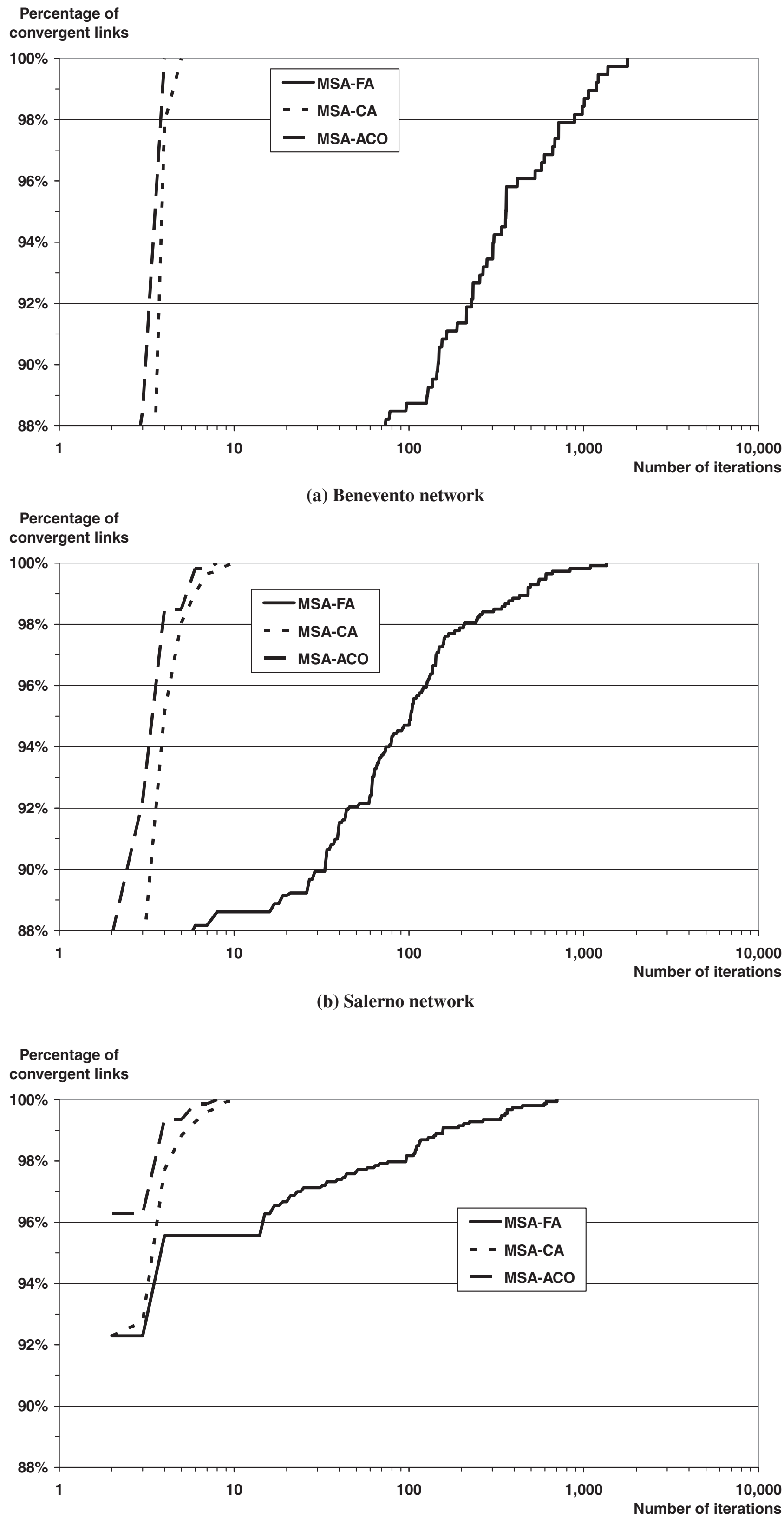



(c) Naples network

**Fig. 6.** Algorithm performances in terms of termination test values.

where $f_l^{MSA\text{-}FA}$, $f_l^{MSA\text{-}CA}$ and $f_l^{MSA\text{-}ACO}$ are flows on link $l$ calculated respectively by means of MSA-FA, MSA-CA and MSA-ACO algorithms;

– 'CA solution error' represents the maximum percentage flow difference between the solution of the MSA-CA algorithm and that of the other algorithms, evaluated by means of:




(a) Benevento network

(b) Salerno network

(c) Naples network

**Fig. 7.** Algorithm performances in terms of convergent links.

$$\max_l \left| \frac{f_l^{MSA-CA} - f_l^{MSA-FA}}{f_l^{MSA-CA}} \right| \tag{48}$$

$$\max_l \left| \frac{f_l^{MSA-CA} - f_l^{MSA-ACO}}{f_l^{MSA-CA}} \right| \tag{49}$$

– 'ACO solution error' represents the maximum percentage flow difference between the solution of the MSA-ACO algorithm and that of the other algorithms, evaluated by means of:

$$\max_l \left| \frac{f_l^{MSA-FA} - f_l^{MSA-ACO}}{f_l^{MSA-ACO}} \right| \tag{50}$$

$$\max_l \left| \frac{f_l^{MSA-CA} - f_l^{MSA-ACO}}{f_l^{MSA-ACO}} \right| \tag{51}$$

Algorithm performances in terms of the above three indicators are compared in Table 2. We report two different indicators for analysing the number of iterations: the number of iterations for achieving the convergence of all links (indicated in the table as '*Number of iterations*') and the number of iterations for achieving the convergence of at least 98% of links (indicated in the table as '*Convergence* > 98.00%').

Likewise we use two kinds of diagrams (Figs. 6 and 7) to provide at each iteration, for each algorithm and for each network: the termination test value (being calculated by means of Eq. (22) in the case of MSA-FA and MSA-CA, and by means of Eq. (38) in the case of MSA-ACO) and the percentage of convergent links (being calculated as the ratio between the number of links satisfying at each iteration the termination test and the total number of links).

The first result, shown by Figs. 6 and 7, is that the proposed ACO-based algorithm requires in all analysed network calculation times, estimated both in terms of number of iterations and percentage of convergent links, lower than other MSA algorithms (i.e. MSA-CA and MSA-FA). However, by analysing details of each network application, we may note that in the case of the Benevento network, the proposed MSA-ACO algorithm required a number of iterations lower than 99.78% with respect to MSA-FA and 20.00% with respect to MSA-CA. Likewise, the convergence of at least 98% of links required a number of iterations lower than 99.55% with respect to MSA-FA and 20.00% with respect to MSA-CA. A comparison in terms of calculation times was not applicable because both MSA-CA and MSA-ACO required less than 1 second.

In terms of the maximum difference calculated by means of Eqs. (46)–(51), we obtain that the adoption of an MSA-ACO instead of a traditional algorithm provides at most an error in flow estimation of 2.29% with respect to MSA-FA and 4.03% relative to MSA-CA, where the maximum difference between MSA-FA and MSA-CA is equal to 3.23%.

In the case of the Salerno network, the proposed MSA-ACO algorithm required 99.41% fewer iterations than MSA-FA and 20.00% fewer than MSA-CA. Likewise, the convergence of at least 98% of links required 98.08% fewer iterations than MSA-FA and 20.00% fewer than MSA-CA. In terms of calculation times the MSA-ACO requires 99.28% less computing time than MSA-FA and 16.67% less than MSA-CA.

Moreover, in terms of maximum difference, we obtain that the adoption of an MSA-ACO instead of other MSA algorithms provides at most an error in flow estimation of 6.31% vis-à-vis MSA-FA and 9.38% compared with MSA-CA, where the maximum difference between traditional MSA algorithms is 4.44%.

Finally, in the case of the Naples network, improvement in MSA-ACO in terms of the number of iterations was 98.86% compared to MSA-FA and 27.27% compared to MSA-CA. Likewise, in the case of 98% convergent links, MSA-ACO provides improvements of 95.88% with respect to MSA–FA and 20.00% with respect to MSA-CA. Further, the MSA-ACO requires 98.73% less computing time than MSA-FA and 20.00% less than MSA-CA. Moreover, in terms of maximum difference, we obtain that the adoption of an MSA-ACO instead of traditional algorithms provides at most an error in flow estimation of 8.75% compared with MSA-FA and 7.19% compared with MSA-CA, where the maximum difference between MSA-FA and MSA-CA is 11.16%.

The comparison of MSA-ACO performances in all three real-scale networks shows that the proposed algorithm provides the same result as traditional MSA algorithms (the maximum difference in terms of solution error has the same order of magnitude as the same error calculated between traditional algorithms) but with a reduction of calculation times of about 99% over MSA-FA and 20% over MSA-CA.

## 5. Conclusion and research prospects

In this paper we proposed an ACO-based algorithm that can be used to solve the LOSS problem. In particular, the proposed algorithm, belonging to 'pseudo' ACO algorithms, consists in modifying the averaged parameter in the MSA framework for improving convergence speed in the search for a solution. The proposed algorithm was obtained by modifying ant behaviour proposed by D'Acierno et al. (2006) by introducing at each signalised intersection to be designed a priority level in ant behaviour similar to people in the case of evacuation.

The proposed algorithm, whose complexity is the same as the MSA-class, was compared with other MSA algorithms in the case of three real-scale networks by using three different indicators: number of iterations, calculation times and differences in solution. All applications showed that the proposed algorithm provides a greater reduction in the number of iterations, hence calculation times, with respect to traditional algorithms (about 99% in the case of MSA-FA and 20% in the case of MSA-CA), whilst achieving the same accuracy. Moreover, it is worth noting that even when adopting different networks with different features (these networks differ in the number of links, number of nodes, number of signalised intersections and number of trips), MSA-ACO continued to behave positively.

In terms of future research, the proposed algorithm could be applied in various real-scale networks in order to verify whether its efficiency is maintained. Further, we propose to extend the model to the case of the GOSS problem and more complex path choice models (such as C-Logit and Probit). Finally, we advocate using the proposed algorithm as a simulation model for imitating the behaviour of transportation systems in more complex design problems or in real-time management in order to highlight the advantages of adopting an ACO approach.

## References


Abdelfatah, A., Mahmassani, H.S., 1999. System optimal time-dependent path assignment and signal timing in traffic networks. Transportation Research Record 1645, 185–193.

Abu-Lebdeh, G., Benekohal, R.F., 2003. Design and evaluation of dynamic traffic management strategies for congested conditions. Transportation Research Part A 37, 109–127.

Allsop, E.R., 1974. Some possibilities for using traffic control to influence trip distribution and route choice. In: Buckley, D.J. (Ed.), Proceedings of the Sixth International Symposium on Transportation and Traffic Theory. Elsevier, New York (NY), USA, pp. 345–373.

Blum, J.R., 1954. Multidimensional stochastic approximation methods. The Annals of Mathematical Statistics 25, 737–744.

Cantarella, G.E., 1997. A general fixed-point approach to multimodal multi-user equilibrium assignment with elastic demand. Transportation Science 31, 107–128.

Cantarella, G.E., Sforza, A., 1995. Network design models and methods for urban traffic management. In: Gartner, N.H., Improta, G. (Eds.), Urban Traffic Networks: Dynamic Flow Modeling and Control. Springer-Verlag, Berlin, Germany, pp. 123–153.

Cantarella, G.E., Improta, G., Sforza, A., 1991. Road network signal setting: Equilibrium conditions. In: Papageorgiou, M. (Ed.), Concise Encyclopedia of Traffic and Transportation Systems. Pergamon Press, Oxford, United Kingdom, pp. 366–371.
Cascetta, E., 2009. Transportation Systems Analysis: Models and Applications. Springer, New York (NY), USA.
Cascetta, E., Gallo, M., Montella, B., 2006. Models and algorithms for the optimization of signal setting on urban networks with stochastic assignment models. Annals of Operations Research 144, 301–328.
Chang, T.H., Lin, J.T., 2000. Optimal signal timing for an oversaturated intersection. Transportation Research Part B 34, 471–491.
Chiou, S.W., 1999. Optimization of area traffic control for equilibrium network flows. Transportation Science 33, 279–289.
D'Acierno, L., Montella, B., De Lucia, F., 2006. A stochastic traffic assignment algorithm based on Ant Colony Optimisation. In: Dorigo, M., Gambardella, L.M., Bittari, M., Martinoli, A., Poli, R., Stützle, T. (Eds.), Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, vol. 4150. Springer-Verlag, Berlin, Germany, pp. 25–36.
D'Acierno, L., Gallo, M., Montella, B., 2010. Ant Colony Optimisation approaches for the transportation assignment problem. WIT Transactions on the Built Environment 111, 37–48.
Dafermos, S., 1980. Traffic equilibrium and variational inequalities. Transportation Science 14, 42–54.
Dafermos, S., 1982. Relaxation algorithms for the general asymmetric traffic equilibrium problem. Transportation Science 16, 231–240.
Daganzo, C.F., 1983. Stochastic network equilibrium with multiple vehicle types and asymmetric, indefinite link cost Jacobians. Transportation Science 17, 282–300.
de Oliveira, D., Bazzan, A.L.C., 2006. Traffic lights control with adaptive group formation based on swarm intelligence. In: Dorigo, M., Gambardella, L.M., Bittari, M., Martinoli, A., Poli, R., Stützle, T. (Eds.), Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, vol. 4150. Springer-Verlag, Berlin, Germany, pp. 520–521.
Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. Transportation Research 5, 83–111.
Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. The MIT Press, Cambridge (MA), USA.
Florian, M., Spiess, H., 1982. The convergence of diagonalization algorithms for asymmetric network equilibrium problems. Transportation Research Part B 16, 477–483.
Gallo, M., D'Acierno, L., Montella, B., 2010. A meta-heuristic approach for solving the Urban Network Design Problem. European Journal of Operational Research 201, 144–157.
Han, B., 1996. Optimising traffic signal settings for periods of time-varying demand. Transportation Research Part A 30, 207–230.
Hu, T.Y., Mahmassani, H.S., 1997. Day-to-day evolution of network flows under real-time information and reactive signal control. Transportation Research Part C 5, 51–69.
Lee, S., Hazelton, M., 1996. Stochastic optimization of combined traffic assignment and signal control junction modelling. In: Lesort, J.B. (Ed.), Proceedings of the 13th International Symposium on Transportation and Traffic Theory. Pergamon Press, Oxford, United Kingdom, pp. 713–735.
Li, Y., Gong, S., 2003. Dynamic ant colony optimisation for TSP. The International Journal of Advanced Manufacturing Technology 22, 528–533.
Lo, H.K., Chang, E., Chan, Y.C., 2001. Dynamic network traffic control. Transportation Research Part A 35, 721–744.
Marcotte, P., 1983. Network optimization with continuous control parameters. Transportation Science 17, 181–197.
Mirchandani, P., Head, L., 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. Transportation Research Part C 9, 415–432.
Mussone, L., Matteucci, M., 2007. An application of Ant Colony Systems for DUE and SUE assignment in congested transportation networks. In: Proceedings of the 11th World Conference on Transport Research, Berkeley (CA), USA; 2007.
Pillai, S.R., Rathi, A.K., Cohen, S.L., 1998. A restricted branch-and-bound approach for generating maximum bandwidth signal timing plans for traffic networks. Transportation Research Part B 32, 517–529.
Poorzahedy, H., Abulghasemi, F., 2005. Application of ant system to network design problem. Transportation 32, 251–273.
Rakha, A.H., 1993. A simulation approach for modeling real-time traffic signal controls. Ph.D. Dissertation, Queen's University of Kingston, Canada.
Sheffi, Y., Powell, W.B., 1981. A comparison of stochastic and deterministic traffic assignment over congested networks. Transportation Research Part B 15, 53–64.
Sheffi, Y., Powell, W.B., 1983. Optimal signal settings over transportation networks. Journal of Transportation Engineering 109, 824–839.
Smith, J.M., 1980. A local traffic control policy which automatically maximises the overall travel capacity of an urban road network. Traffic Engineering and Control 21, 298–302.
Smith, J.M., 1981. Properties of a traffic control policy which ensure the existence of a traffic equilibrium consistent with the policy. Transportation Research Part B 15, 453–462.
Smith, J.M., Van Vuren, T., 1993. Traffic equilibrium with responsive traffic control. Transportation Science 27, 118–132.
Teodorovic, D., 2008. Swarm intelligence systems for transportation engineering: Principles and applications. Transportation Research Part C 16, 651–667.
Webster, V.F., 1958. Traffic signal settings. Road Research Technical Paper No. 39. Road Research Laboratory: London, United Kingdom.
Wey, W.M., 2000. Model formulation and solution algorithm of traffic signal control in an urban network. Computers Environment and Urban Systems 24, 355–377.
Wolshon, B., Taylor, W.C., 1999. Analysis of intersection delay under real-time adaptive signal control. Transportation Research Part C 7, 53–72.
Wong, K.C., Wong, S.C., 2003. Lane-based optimization of signal timings for isolated junctions. Transportation Research Part B 37, 63–84.
Wong, C.S., Yang, H., 1997. Reserve capacity of a signal-controlled road network. Transportation Research Part B 31, 397–402.
Yang, H., Yagar, S., 1995. Traffic assignment and signal control in saturated road networks. Transportation Research Part A 29, 125–139.