Adapting the Pheromone Evaporation Rate in Dynamic Routing Problems

Michalis Mavrovouniotis and Shengxiang Yang

School of Computer Science and Informatics, De Montfort University
The Gateway, Leicester LE1 9BH, United Kingdom
`m.mavrovouniotis@hotmail.com`, `syang@dmu.ac.uk`

Abstract. Ant colony optimization (ACO) algorithms have proved to be able to adapt to dynamic optimization problems (DOPs) when stagnation behaviour is avoided. Several approaches have been integrated with ACO to improve its performance for DOPs. The adaptation capabilities of ACO rely on the pheromone evaporation mechanism, where the rate is usually fixed. Pheromone evaporation may eliminate pheromone trails that represent bad solutions from previous environments. In this paper, an adaptive scheme is proposed to vary the evaporation rate in different periods of the optimization process. The experimental results show that ACO with an adaptive pheromone evaporation rate achieves promising results, when compared with an ACO with a fixed pheromone evaporation rate, for different DOPs.

1 Introduction

Ant colony optimization (ACO) algorithms have shown good performance when applied to difficult optimization problems under static environments [2]. However, in many real-world applications, we have to deal with dynamic environments, where the optimum changes and needs re-optimization. It is believed that ACO algorithms can adapt to dynamic changes since they are inspired from nature, which is a continuous adaptation process [7]. In practice, they can adapt by transferring knowledge from past environments, using the pheromone trails, to speed up re-optimization. The challenge to such algorithms lies in how quickly they can react to dynamic changes in order to maintain the high quality of output instead of early stagnation behaviour, where all ants construct the same solutions and lose their adaptation capabilities.

Developing strategies for ACO algorithms to deal with stagnation behaviour and address dynamic optimization problems (DOPs) has attracted a lot of attention, which includes local and global restart strategies [6], memory-based approaches [5], pheromone manipulation schemes to maintain diversity [3], and immigrants schemes to increase diversity [8].

The adaptation capabilities of ACO rely on the pheromone evaporation where a constant amount of pheromone is deducted to eliminate pheromone trails that represent bad solutions that may bias ants to search to the non-promising areas of the search space. In this paper, the impact of the pheromone evaporation rate

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2013, LNCS 7835, pp. 606–615, 2013.

is examined on DOPs, and an adaptive scheme is designed for ACO. Adaptive methods have been successfully applied for different parameters of ACO, including the evaporation rate [10,12]. However, these methods have been investigated on static optimization problems.

The rest of the paper is organized as follows. Section 2 describes the generation of dynamic routing DOPs. Section 3 describes an ACO algorithm and gives details for its adaptation capabilities in DOPs. Section 4 describes the proposed scheme where the evaporation rate in ACO is adapted. Section 5 describes the experiments carried out on a series of different DOPs. Finally, Section 6 concludes this paper with directions for future work.

2 Generating Dynamic Routing Environments

Routing problems are usually illustrated using weighted graphs. Let $G = (V, E)$ be a weighted graph where V is a set of n nodes and E is a set of links. Each node i has a location defined by (x, y) and each link (i, j) is associated with a non-negative distance d_{ij}. Usually, the distance matrix of a problem instance is defined as $\boldsymbol{D} = (d_{ij})_{n \times n}$.

In order to generate dynamic routing problems, the dynamic benchmark generator for permutation-encoded problems (DBGP) [9] is used, which converts any static problem instance to a dynamic environment. In case the optimum of the static problem instance is known, then it will remain known during the environmental changes, because DBGP biases algorithms to search to a new location in the fitness landscape, instead of modifying the fitness landscape.

Every f iterations a random vector $\boldsymbol{r}(T)$ is generated that contains all the objects of a problem instance of size n, where $T = \lceil t/f \rceil$ is the index of the period of change, t is the iteration count of the algorithm, and f determines the frequency of change. The magnitude m of change depends on the number of swapped locations of objects. More precisely, $m \in [0.0, 1.0]$ defines the degree of change, in which only the first $m \times n$ of $\boldsymbol{r}(T)$ object locations are swapped. Then a randomly re-ordered vector $\boldsymbol{r'}(T)$ is generated that contains only the first $m \times n$ objects of $\boldsymbol{r}(T)$. Therefore, exactly $m \times n$ pairwise swaps are performed using the two random vectors.

3 ACO in Dynamic Environments

3.1 $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System

The ACO metaheuristic consists of a population of μ ants where they construct solutions and share their information among each other via their pheromone trails. The first ACO algorithm developed is the Ant System (AS) [1]. Many variations of the AS have been applied to difficult optimization problems [2].

One of the best performing ACO algorithm is the $\mathcal{MAX}$-$\mathcal{MIN}$ AS ($\mathcal{MM}$AS) [11]. Ants read and write pheromones in order to construct their solutions.

Each ant k uses a probabilistic rule to choose the next city to visit. The decision rule an ant k uses to move from city i to city j is defined as follows:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \text{if } j \in \mathcal{N}_i^k, \tag{1}$$

where τ_{ij} is the existing pheromone trail between cities i and j, $\eta_{ij} = 1/d_{ij}$ is the heuristic information available a priori, where d_{ij} is the distance between cities i and j. $\mathcal{N}_i^k$ denotes the neighbourhood of cities for ant k when the current city is i. α and β are the two parameters which determine the relative influence of τ and η, respectively. The pheromone trails in $\mathcal{MM}$AS are updated by applying evaporation as follows:

$$\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij}, \forall (i,j), \tag{2}$$

where ρ is the evaporation rate which satisfies $0 < \rho \leq 1$, and τ_{ij} is the existing pheromone value. After evaporation the best ant deposits pheromone as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{best}, \forall (i,j) \in T^{best}, \tag{3}$$

where $\Delta\tau_{ij}^{best} = 1/C^{best}$ is the amount of pheromone that the best ant deposits and C^{best} defines the solution quality of tour T^{best}. Since only the best ant deposits pheromone, the algorithm will quickly converge towards the best solution of the first iteration. Therefore, pheromone trail limits are imposed in order to avoid this behaviour.

3.2 Response to Dynamic Changes

ACO algorithms are able to use knowledge from previous environments using the pheromone trails generated in the previous iterations. For example, when the changing environments are similar, the pheromone trails of the previous environment may provide knowledge to speed up the optimization process to the new environment. However, the algorithm needs to be flexible enough to accept the knowledge transferred from the pheromone trails, or eliminate the pheromone trails, in order to adapt well to the new environment.

ACO algorithms can be applied directly to DOPs without any modifications due to the pheromone evaporation. Lowering the pheromone values enables the algorithm to forget bad decisions made in previous iterations. When a dynamic change occurs, evaporation eliminates the pheromone trails of the previous environment from areas that are not visited frequently and may bias ants not to adapt well to the new environment.

The adaptation via pheromone evaporation may be a sufficient choice when the changing environments are similar, otherwise a complete re-initialization of the pheromone trails after a dynamic change occurs may be a better choice. However, such action is available only in DOPs where the frequency of change is available beforehand or in DOPs where the dynamic changes can be detected. In our case, the dynamic changes can be detect by re-evaluating some stored solutions, used as detectors, in every iteration [8].

4 ACO with Adaptive Evaporation Rate

4.1 Effect of the Pheromone Evaporation Rate

Although ACO has adaptation capabilities due to the pheromone evaporation, the time required to adapt to the new environment may depend on the problem size and the magnitude of change. When the environmental change is severe then it may take longer to eliminate unused pheromone trails, therefore a high evaporation rate may be more suitable. More precisely, a high evaporation rate will eliminate the high intensity of pheromone trails that are usually concentrated to the optimum of the previous environment that is caused by stagnation behaviour. On the other hand, a high pheromone evaporation rate may destroy information that can be used on further environments, since any bad solution in the current environment may be good in the next environment.

In traditional ACO algorithms the evaporation rate is usually fixed. A low evaporation rate corresponds to slow adaptation, whereas a high evaporation rate corresponds to fast adaptation. However, we believe that a fixed evaporation rate is not the best choice when addressing DOPs, since at different stages of the optimization process for different optimization problems and under different dynamic environments, the most appropriate evaporation rate varies.

4.2 Detect Stagnation Behaviour

In order to adapt the value of the pheromone evaporation during the search process, the exploration of the algorithm is measured in order to detect stagnation behaviour. A direct way that can give an indication of exploration is to measure the diversity of the solutions. The measurement for routing problems is usually based on the common edges between the solutions [8]. Such a measure may be computational expensive since there are $\mathcal{O}(n^2)$ possible pairs to be compared and each single comparison has a complexity of $\mathcal{O}(n)$.

A more efficient measurement is the λ-branching factor [4], which measures the distribution of the pheromone trail values. The idea of λ-branching is described as follows: If for a given object $i \in V$, the concentration of pheromone trails on almost all the incident arcs becomes very small but is large for a few others, then the freedom of exploring other paths from object i is very limited. Therefore, if this situation arises simultaneously for all objects of graph G, the search space that is searched by ants becomes relatively small.

The average $\bar{\lambda}(t)$ branching factor at iteration t is defined as follows:

$$\bar{\lambda}(t) = \frac{1}{2n} \sum_{i=1}^{n} \lambda^i, \tag{4}$$

where n is the number of objects in the corresponding graph and λ^i is the λ-branching factor for object i, which is defined as follows:

$$\lambda^i = \sum_{j=1}^{d} L_{ij} \tag{5}$$

where d is the number of available arcs incident to object i and L_{ij} is defined as follows:

$$L_{ij} = \begin{cases} 1, & \text{if } (\tau^i_{min} + \lambda(\tau^i_{max} - \tau^i_{min})) \leq \tau_{ij}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where λ is a constant parameter ($\lambda = 0.05$ by default [4]), τ^i_{min} and τ^i_{max} are the minimum and maximum pheromone trail values on the arcs incident to object i, respectively. A value of $\bar{\lambda}(t)$ close to 1 indicates stagnation behaviour.

4.3 Adapting Pheromone Evaporation Rate

Considering the statements above, if the algorithm reaches stagnation behaviour, the evaporation rate needs to be increased in order to eliminate the high intensity of pheromone trails in some areas and increase exploration. However, very high exploration may disturb the optimization process because of randomization [8].

According to the behaviour of the algorithm in terms of searching, we have the following pheromone evaporation rate update rule:

$$\rho(t) = \begin{cases} \rho(t-1) - \sigma, & \text{if } \bar{\lambda}(t) > 1, \\ \rho(t-1) + \sigma, & \text{otherwise.} \end{cases} \tag{7}$$

where $\bar{\lambda}(t)$ is defined in Eq. (4) and σ is the step size of varying the evaporation rate ρ at iteration t. A good value of σ was found to be 0.001 because a higher value may quickly increase ρ to an extreme evaporation rate and destroy information and a smaller value may not have any effect to the performance of ACO.

5 Experimental Study

5.1 Experimental Setup

In the experiments, we compare a $\mathcal{MM}$AS with a global re-initialization of the pheromone trails, denoted as $\mathcal{MM}\text{AS}_R$, and a $\mathcal{MM}$AS with the best fixed evaporation rate, denoted as $\mathcal{MM}\text{AS}_B$ against the $\mathcal{MM}$AS with the proposed adaptive pheromone evaporation, denoted as $\mathcal{MM}\text{AS}_A$. For all algorithms, we set $\alpha = 1$, $\beta = 5$, $q_0 = 0.0$, and $\mu = 50$, except for $\mathcal{MM}\text{AS}_R$ where $\mu = 50 - d_T$, where $d_T = 6$ is the number of detectors. The evaporation rate for $\mathcal{MM}\text{AS}_R$ was set to $\rho = 0.4$. For $\mathcal{MM}\text{AS}_B$ the best value from $\rho \in \{0.02, 0.2, 0.4, 0.6, 0.8\}$ was selected, whereas for $\mathcal{MM}\text{AS}_A$ ρ was adapted by Eq. (7).

For each algorithm on a DOP, $N = 30$ independent runs were executed on the same environmental changes. The algorithms were executed for $G = 1000$ iterations and the overall offline performance is calculated as follows:

$$\bar{P}_{offline} = \frac{1}{G} \sum_{i=1}^{G} \left(\frac{1}{N} \sum_{j=1}^{N} P^*_{ij} \right) \tag{8}$$

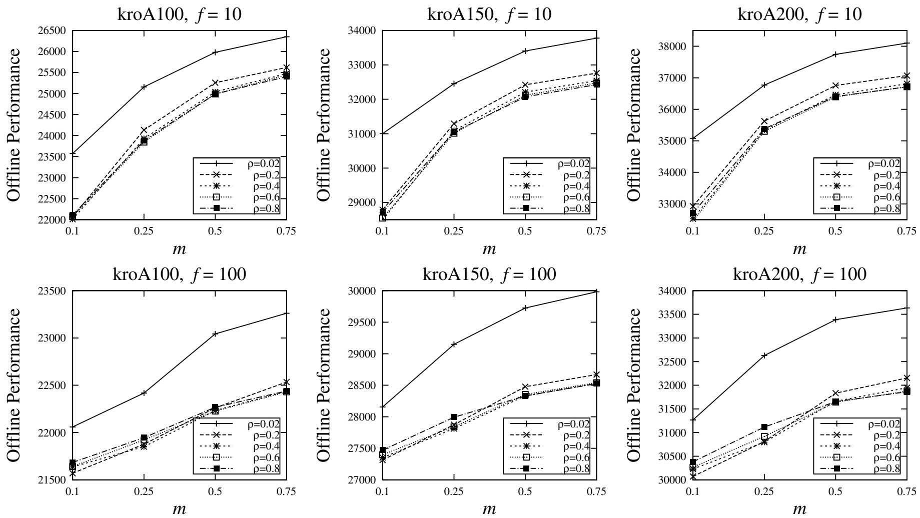


Fig. 1. Impact of the evaporation rate on the offline performance of a conventional $\mathcal{MM}$AS on different DOPs

where P_{ij}^{*} defines the tour cost of the best ant since the last dynamic change of iteration i of run j [7].

We took three travelling salesman problem (TSP) instances[1] and three vehicle routing problem (VRP) instnaces[2] as the base and used the DBGP described in Section 2 to generate DOPs. The value of f was set to 10 and 100, which indicate fast and slowly changing environments, respectively. The value of m was set to 0.1, 0.25, 0.5, and 0.75, which indicate the degree of environmental changes from small, to medium, to large, respectively. As a result, eight dynamic environments, i.e., 2 values of $f \times 4$ values of m, for each problem instance are generated to systematically analyze the adaptation and searching capability of algorithms on the DOPs.

5.2 Experimental Results and Analysis

The experimental results regarding the different ρ values for $\mathcal{MM}\text{AS}_B$ on dynamic TSPs are presented in Fig. 1. Note that the corresponding experimental results for dynamic VRPs show similar observations and are not presented here. The offline performance of the different algorithms on dynamic TSPs and dynamic VRPs and the corresponding statistical results of Wilcoxon rank-sum test, at the 0.05 level of significance are presented in Table 1 and Table 2, respectively. Moreover, the dynamic behaviour of the algorithms is presented in Fig. 2. From the experimental results, several observations can be made by comparing the behaviour of the algorithms.

---

[1] `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`.

[2] `http://neo.lcc.uma.es/vrp/`.

Table 1. Experimental results of the algorithms regarding the offline performance

Travelling Salesman Problem Instances								
	$f = 10$				$f = 100$			
$m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
Alg. & Inst.	`kroA100`(Optimum=21282)							
$\mathcal{MM}$AS$_B$	22010	23844	24989	25401	21570	21850	22227	22430
$\mathcal{MM}$AS$_A$	22069	23542	24448	24800	21683	21819	22049	22468
$\mathcal{MM}$AS$_R$	24576	24580	24583	24588	22244	22252	22224	22212
Alg. & Inst.	`kroA150`(Optimum=26524)							
$\mathcal{MM}$AS$_B$	28488	31013	32070	32426	27315	27814	28330	28526
$\mathcal{MM}$AS$_A$	28690	30507	31444	31778	27299	27726	28140	28262
$\mathcal{MM}$AS$_R$	31520	31526	31515	31516	28208	28204	28215	28198
Alg. & Inst.	`kroA200`(Optimum=29368)							
$\mathcal{MM}$AS$_B$	32454	35300	36394	36711	30071	30796	31644	31863
$\mathcal{MM}$AS$_A$	32353	34524	35560	35872	30167	30645	31245	31506
$\mathcal{MM}$AS$_R$	35375	35368	35362	35375	31282	31368	31338	31380
Vehicle Routing Problem Instances								
	$f = 10$				$f = 100$			
$m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
Alg. & Inst.	`F-n45-k4`(Optimum=724)							
$\mathcal{MM}$AS$_B$	800.42	807.56	816.98	823.20	792.79	796.24	796.33	797.65
$\mathcal{MM}$AS$_A$	800.55	805.22	814.02	820.39	795.63	797.66	797.22	799.04
$\mathcal{MM}$AS$_R$	812.16	812.06	812.34	812.16	799.76	800.46	799.55	799.70
Alg. & Inst.	`F-n72-k4`(Optimum=237)							
$\mathcal{MM}$AS$_B$	268.69	279.67	285.76	288.36	259.95	263.00	265.82	266.81
$\mathcal{MM}$AS$_A$	270.36	281.13	286.47	289.20	261.17	263.76	267.18	267.44
$\mathcal{MM}$AS$_R$	291.09	291.14	291.14	291.23	270.64	270.91	271.11	270.61
Alg. & Inst.	`F-n135-k7`(Optimum=1162)							
$\mathcal{MM}$AS$_B$	1298.71	1339.50	1365.33	1375.41	1255.00	1271.66	1286.62	1291.91
$\mathcal{MM}$AS$_A$	1297.46	1335.77	1363.29	1372.89	1255.21	1269.49	1283.98	1288.14
$\mathcal{MM}$AS$_R$	1348.90	1348.65	1348.87	1348.85	1281.08	1283.68	1283.35	1282.20

First, when the evaporation rate is set to $\rho = 0.02$, which is the recommended value for $\mathcal{MM}$AS on static problems [2, p. 71], has the worst results on DOPs, as observed from Fig. 1. Furthermore, a high evaporation rate, i.e., $\rho \geq 0.4$, often achieves better performance when $f = 10$. This is natural because when the environment changes quickly, a fast adaptation is required. When $f = 100$, the evaporation rate depends on the magnitude of change, e.g., when $m = 0.1$, $\rho = 0.2$ shows better performance. However, as the magnitude of change increases, a higher value of ρ achieves better performance. This validates our claim that the time required for ACO, in which pheromone evaporation is used, to adapt to the new environment depends on the magnitude of change.

Second, $\mathcal{MM}$AS$_B$ is outperformed by $\mathcal{MM}$AS$_R$ on most DOPs with $m = 0.5$ and $m = 0.75$, whereas the former outperforms the latter on all DOPs with $m = 0.1$ and $m = 0.25$; see the comparisons $\mathcal{MM}$AS$_B$ ⇔ $\mathcal{MM}$AS$_R$ in Table 2.

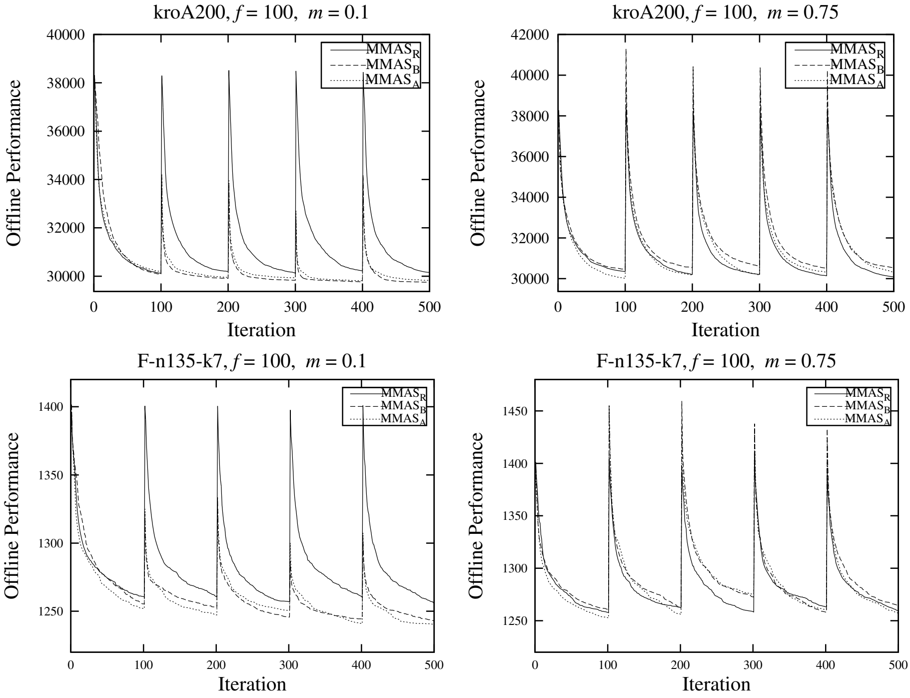
Table 2. Statistical tests of comparing algorithms regarding the offline performance, where "+" or "−" means that the first algorithm is significantly better or the second algorithm is significantly better, respectively, and "∼" means that the algorithms are not significantly different

Travelling Salesman Problem Instances												
Alg. & Inst.	`kroA100`				`kroA150`				`kroA200`			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_B$	−	+	+	+	−	+	+	+	+	+	+	+
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	+	−	+	+	+	−	+	+	−	−
$\mathcal{MM}\text{AS}_B \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	−	−	+	+	−	−	+	+	−	−
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_B$	−	∼	+	∼	∼	+	+	+	∼	+	+	+
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	+	−	+	+	+	∼	+	+	+	−
$\mathcal{MM}\text{AS}_B \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	∼	−	+	+	−	−	+	+	−	−

Vehicle Routing Problem Instances												
Alg. & Inst.	`F-n45-k4`				`F-n72-k4`				`F-n135-k7`			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_B$	∼	+	+	+	−	−	−	−	∼	+	+	+
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	−	−	+	+	+	+	+	+	−	−
$\mathcal{MM}\text{AS}_B \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	−	−	+	+	+	+	+	+	−	−
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_B$	∼	∼	∼	−	−	∼	−	∼	∼	∼	∼	∼
$\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	+	∼	+	+	+	+	+	+	∼	−
$\mathcal{MM}\text{AS}_B \Leftrightarrow \mathcal{MM}\text{AS}_R$	+	+	+	−	+	+	+	+	+	+	∼	−

This is because when the environments are similar, due to a slight change, the pheromone trails of the previous environment help to start the optimization process from a promising area in the search space, whereas when the environments are different, due to a severe change, the pheromone trails of the previous environment mislead the searching to non-promising areas. This validates our claim that the adaptation of pheromone evaporation is useful when the environments are similar and useful knowledge can be transferred.

Finally, the proposed $\mathcal{MM}\text{AS}_A$ outperforms $\mathcal{MM}\text{AS}_B$ on most TSP DOPs when $m = 0.25$, $m = 0.5$ and $m = 0.75$, whereas the former is comparable with the latter when $m = 0.1$; see the comparisons $\mathcal{MM}\text{AS}_A \Leftrightarrow \mathcal{MM}\text{AS}_B$ in Table 2. This is probably because when the evaporation is high, it may destroy useful knowledge from the previous environment after a dynamic change. Therefore, a low evaporation rate sometimes may be a better choice, even when the dynamic change is severe, for the first iterations after a dynamic change to obtain knowledge, and a higher evaporation rate may be a better choice later on to avoid the stagnation behaviour. This can be observed from Fig. 2 where $\mathcal{MM}\text{AS}_A$ converges faster and to a better optimum than $\mathcal{MM}\text{AS}_B$. Fortunately, even if $\mathcal{MM}\text{AS}_A$ does not achieve the best result compared to $\mathcal{MM}\text{AS}_B$, e.g., when $f = 100$ for VRP DOPs, its performance level is still satisfactory since they are usually not significantly different. This can be expected since the


kroA200, $f = 100$, $m = 0.1$ kroA200, $f = 100$, $m = 0.75$

F-n135-k7, $f = 100$, $m = 0.1$ F-n135-k7, $f = 100$, $m = 0.75$

Fig. 2. Dynamic behaviour of the algorithms with respect to offline performance against the iterations in slowly changing environments for the first 500 iterations in different DOPs: 1) `kroA200` for TSP; and 2) `F-n135-k7` for VRP

results of $\mathcal{MM}\mathrm{AS}_{\mathcal{B}}$ are obtained via fine-tuning the evaporation rate. Moreover, $\mathcal{MM}\mathrm{AS}_{A}$ outperforms $\mathcal{MM}\mathrm{AS}_{R}$ in most DOPs, except when $m = 0.75$, because of the same reasons discussed for $\mathcal{MM}\mathrm{AS}_{\mathcal{B}}$ previously.

6 Conclusions

This paper examines the impact of the pheromone evaporation on the performance of ACO algorithms for DOPs. An adaptive evaporation rate is proposed for ACO to deal with DOPs, which is based on the detection of the stagnation behaviour. Experimental studies were performed on a series of DOPs to investigate the performance of the proposed approach. From the experimental results, several conclusions can be drawn. First, pheromone evaporation is important for ACO to address DOPs. Second, the higher the magnitude of the dynamic change, the higher the evaporation rate is needed. Third, the adaptation capabilities of pheromone evaporation perform well only when the environments are similar; otherwise, a re-initialization of the pheromone trails is required. Forth, the proposed adaptive evaporation rate promotes the performance of ACO in many routing DOPs but depends on the dynamics and the type of the DOP.

Finally, compared to the tedious work of fine-tuning the pheromone evaporation rate manually, the proposed adaptive scheme is more convenient and has sufficiently good performance under different conditions. However, the performance is slightly decreased in some cases, for the sake of this convenience.

For future work, it will be interesting to consider other ways for adapting the evaporation rate. Moreover, there are evidence that the more parameters adapted in ACO, the better the performance in optimization problems with static environment [10]. Therefore, another future work is to adapt more ACO parameters, e.g., α and β, in parallel for DOPs.

Acknowledgement. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under Grant EP/K001310/1.

References

1. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. on Syst., Man and Cybern., Part B: Cybern. 26(1), 29–41 (1996)
2. Dorigo, M., Stützle, T.: Ant colony optimization. The MIT Press, London (2004)
3. Eyckelhof, C.J., Snoek, M.: Ant Systems for a Dynamic TSP: Ants Caught in a Traffic Jam. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) ANTS 2002. LNCS, vol. 2463, pp. 88–99. Springer, Heidelberg (2002)
4. Gambardella, M.L., Dorigo, M.: Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: Proc of the 12th Int. Conf. on Machine Learning, pp. 252–260. Morgan Kaufmann (1995)
5. Guntsch, M., Middendorf, M.: Applying Population Based ACO to Dynamic Optimization Problems. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) ANTS 2002. LNCS, vol. 2463, pp. 111–122. Springer, Heidelberg (2002)
6. Guntsch, M., Middendorf, M.: Pheromone Modification Strategies for Ant Algorithms Applied to Dynamic TSP. In: Boers, E.J.W., Gottlieb, J., Lanzi, P.L., Smith, R.E., Cagnoni, S., Hart, E., Raidl, G.R., Tijink, H. (eds.) EvoWorkshop 2001. LNCS, vol. 2037, pp. 213–222. Springer, Heidelberg (2001)
7. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments - a survey. IEEE Trans. on Evol. 9(3), 303–317 (2005)
8. Mavrovouniotis, M., Yang, S.: Ant colony optimization with memory-based immigrants for the dynamic vehicle routing problem. In: Proc. of the 2012 IEEE Congress on Evol. Comput., pp. 2645–2652. IEEE Press (2012)
9. Mavrovouniotis, M., Yang, S., Yao, X.: A Benchmark Generator for Dynamic Permutation-Encoded Problems. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 508–517. Springer, Heidelberg (2012)
10. Pellegrini, P., Stützle, T., Birattari, M.: A critical analysis of parameter adaptation in ant colony optimization. Swarm Intelli. 6(1), 23–48 (2012)
11. Stützle, T., Hoos, H.: The $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System and local search for the traveling salesman problem. In: Proc. of the 1997 IEEE Int. Conf. on Evol. Comput., pp. 309–314. IEEE Press (1997)
12. Stützle, T., López-Ibáñez, M., Pellegrini, P., Maur, M., de Oca, M.M., Birattari, M., Dorigo, M.: Parameter adaptation in ant colony optimization, pp. 191–215. Springer, Heidelberg (2012)