Genetics and Competing Strategies in a Threshold Model for Mail Processing

Harry Goldingay and Jort van Mourik,
Information Engineering, Aston University,
Aston Triangle, Birmingham B4 7ET, UK

May 16, 2008

Abstract

Multi-agent algorithms inspired by the division of labour in social insects are applied to a problem of distributed mail retrieval in which agents must visit mail producing cities and choose between mail types under certain constraints. The efficiency (i.e. the average amount of mail retrieved per time step), and the flexibility (i.e. the capability of the agents to react to changes in the environment) are investigated both in static and dynamic environments.
New rules for mail selection and specialisation are introduced and are shown to to exhibit improved efficiency and flexibility compared to existing ones. We employ a genetic algorithm which allows the various rules to evolve and compete. Apart from obtaining optimised parameters for the various rules for any environment, we also observe extinction and speciation.
From a more theoretical point of view, in order to avoid finite size effects, most results are obtained for large population sizes. However, we do analyse the influence of population size on the performance. Furthermore, we critically analyse the causes of efficiency loss, derive the *exact* dynamics of the model in the large system limit under certain conditions, derive theoretical upper bounds for the efficiency, and compare these with the experimental results.

1 Introduction

In the field of distributed systems, communication costs between elements of a system can significantly limit its performance [1]. As such the idea of self-organising systems is an attractive one because such systems inherently remove the need for the communication necessary for centralised control. Models of social insect behaviour based on the idea of stimergy, developed by Grassé [2], have been a source of inspiration in designing such self-organising systems, providing good solutions with high degrees of flexibility and robustness [3]. These stimergic models have the additional advantage of removing the need for direct communication between the elements of a system as their behaviour is solely dependent on their perception of the environment.

In this paper we extend a social insect inspired method for solution of a problem of distributed mail retrieval based on the problem studied by Price et. al in [4]. This problem involves agents repeatedly travelling to cities which can produce and store a set of different types of mail. Each Agent is associated with a mail processing centre and must choose a piece of mail stored at a city to take for processing under the constraint that switching mail types causes a penalty in processing time.

The model we use to solve this problem, known as the threshold model [5], [6], was inspired by task allocation in social insect colonies. It has been successfully applied to other problems which require robust, decentralised control including the scheduling of truck painting [7], [8], which involves similar constraints, and the real world example of conserving battery life in a remote sensor network [1].

The idea of thresholds as a method for task allocation was developed by Bonabeau et al. in [5] in order to show how the flexibility of insect colonies to different circumstances can be explained by the autonomous flexibility to engage in tasks of the individuals which comprise them. They proposed a model, known as the fixed response threshold (FRT) model, which stated that the tasks that an individual was capable of engaging in could be broken down into types and that each instance of a task has some stimulus associated with it which is indicative of its demand for completion. Each colony member has a set of thresholds which determine their preference for engaging in each type of task. Upon encountering a task an individual will compare the stimulus s of the task with the corresponding threshold θ, and use this to determine the probability of uptake of that task. The probability of uptake should be high for $s >> \theta$, low for $s << \theta$, zero for $s = 0$ (no demand for the task), and $\frac{1}{2}$ for $s = \theta$, and is defined by a *threshold function* $\Theta(s, \theta)$.

In the original formulation of the model it was assumed that each agent had a genetically determined set of thresholds which were constant over an agent's lifetime. However, while this was able to account for such features of social insect colonies as increased uptake of tasks by nonspecialist individuals upon the removal of specialists in these tasks, it was unable to account for the initial distribution of specialisations or the re-specialisation of colony members to meet changes in the distribution of stimulus. In order to account for these features Theraulaz et al., in [6], introduced a process of self reinforcement whereby time spent performing a task would lead to a decrease in an agent's threshold for this task whilst time spent not performing this task would cause the threshold to increase. Individuals possess an update rule, U, which governs the magnitude of these changes. This model is known as the variable response threshold (VRT) model. In this paper a discretised version of this model is used and increases in thresholds at times when the individual is inactive are discounted. As such an agent with thresholds $\theta = (\theta_1, ...\theta_n)$ will, upon completion of a task of type i, update its thresholds as follows:

$$U(\boldsymbol{\theta}, i) = (u(\theta_1, i), ..., u(\theta_n, i)) \tag{1.1}$$

where

$$\begin{cases} u(\theta_j, i) < \theta_j & \text{if } i = j, \\ u(\theta_j, i) > \theta_j & \text{otherwise,} \end{cases} \tag{1.2}$$

with θ_j restricted to some range $[\theta_{max},\theta_{min}]$. These changes are related to both the size of the threshold and the time taken to complete the task.

This paper builds upon work by Bonabeau et al. [9] who use a problem of mail agents serving a zonal demand to demonstrate the applicability of the threshold model to a non-static task allocation problem. This problem was then developed by Price et al. [4], [10], into the constrained, distributed, mail retrieval problem which we shall study in this paper. They showed that the threshold model gives a good solution to the problem when compared to other algorithms, particularly under a change of mail production probabilities.

We investigate the performance of the system with large populations and, in order to make this computationally viable, we propose an alteration to the problem in which each city may have a set of mail to be taken, but the information about this mail is local and cannot be determined from outside the city. The main focus of our work shall be the sources of loss of efficiency in the system and these are statistically analysed from a population dynamics perspective. The flexibility of the system is also tested by the introduction of a dynamic environment in which mail production probabilities are continuously varied.

In order to minimise this loss we introduce a set of new update rules which define agents' task specialisation behaviour. We then use these rules as "species" in a genetic algorithm to determine the best rule in a given circumstance and to optimise the parameters of these rules. A new set of threshold functions better suited to this genetic optimisation are also introduced.

The rest of of the paper is organised as follows:
In the next section 2, we introduce the model and the various strategies to solve it. In section 3, we perform a theoretical analysis that in the large system limit under certain conditions describes the dynamics of the model exactly, and provide a theoretical upper bound for the performance of any algorithm. In section 4, we present and discuss the numerical results and compare them with both the exact dynamics and the theoretical bounds. Finally, in section 5, we summarise our main findings, discuss the limitations of the current setting, and give an outlook to future work.

2 The Model

2.1 The Mail Processing Problem

In order to study the VRT model a problem which can be used to examine its reaction to various profiles of stimulus must be used. A good candidate is the problem of distributed mail retrieval studied by Price et al. in [10],[4] in which agents using VRT rules have been shown to perform well and to exhibit some of the key features model. We study a modified version of this problem in which there are a set of N_c mail producing cities each of which is capable of producing and storing one batch each of N_m different types of mail. There are also a set of N_a mail processing centres whose task it is to process these batches of mail. In order to achieve this, each centre has one associated mail collection agent whose task it is to travel to a city and return with a batch of mail for its centre to process.

Each mail type requires a different processing method and at any point in time the processing centre of agent a is specialised in one specific type σ'_a. When processing a batch of this mail type the centre can do so efficiently, taking a time t_p. However, in order to process a batch of a different mail type m, the centre must undergo alterations. This *changeover* $\sigma'_a \to m$ (including the processing of the batch) takes a time $t_c > t_p$.

In order to reduce the direct impact of these changeovers, each processing centre has a *mail queue* in which it can temporarily store mail while processing other batches. This queue is capable of storing up to L_q batches of mail and, while it has space in it, a centre's agent will continue to collect mail from cities. A processing centre must process the mail in its queue in the order in which it arrived, such that all the freedom in the system is concentrated in the behaviour of the collection agents. Therefore, we define the *effective specialisation* σ_a of the agent as the last collected mail type, because σ'_a will be σ_a by the time the next collected mail is processed.

Time evolves in discrete steps of $\Delta t = 1$ in which:

1. Each centre processes the top batch of mail (if any), thus emptying one space in its queue, or proceeds with its changeover.
2. Each agent a randomly picks one city c to visit, unless its queue is full.
3. Each city now has a set Ψ_c of visiting agents, and the order in which the agents are allowed to act is determined randomly.
4. Each agent acts by examining the (remaining) waiting batches of mail at the city in a random order choosing or rejecting them on an individual basis, until either a batch is chosen or all have been rejected.
5. Then each agent returns to its processing centre and deposits the chosen batch (if any), into the queue.
6. Finally, the cities increase the stimuli of left-over batches and produce new batches.

No centralised control of the agents is permitted and so the aim of the problem is to give agents a set of autonomous rules in order to maximise the amount of mail processed. We refer to this in terms of average mail processed per agent per time step, or *efficiency.* It is clear that this efficiency is limited by several factors. The agent must strike a balance between maximising the proportion of times it takes mail when it visits a city and minimising the amount of time which it spends not visiting cities due to its centre having a full queue. The likelihood of having a full queue increases with the agents likelihood to take mail of a type different from that it took previously. It is clear that in an ideal situation, an agent would always take mail of the type that it took previously. However, this ideal situation is impossible unless the agent rejects all types of mail other than the one previously taken, which in turn is not ideal as it increases the number of times the agent returns empty handed and in extreme cases may even lead to a deadlock situation in which all mail of a certain type is rejected by all agents. In [10], [4] it was shown that agents using rules based on the VRT model exhibit good performance in these tasks when compared with several other algorithms, particularly when flexibility is required.

For each agent a, the uptake of each mail type is defined as a distinct task, and depends on: its threshold function Θ, its vector of thresholds $\boldsymbol{\theta}_a = (\theta_{a,1}, .., \theta_{a,N_m})$, the vector of the queue of its processing centre $\boldsymbol{q}_a = (q_{a,1}, .., q_{a,L_q})$ which can store a backlog of up to L_q batches of mail. The threshold function also requires a stimulus s, and this is taken to be the waiting time of the batch at the city. Therefore, each city c has a vector $\boldsymbol{w}_c = (w_{c,1}, .., w_{c,N_m})$ where $w_{c,m}$ is the waiting time of the batch of mail type m. Note that $w_{c,m} = 0$ indicates that there is no batch of that type of mail present, either because no such batch was produced, or because another agent has already taken that batch. Upon production of a batch of mail type m, its waiting time is initialised to $w_m = 1$, and at the end of each time step the waiting times of remaining batches of mail are increased by 1. The aim of this formulation is that for an appropriately chosen update rule an agent will tend to a low threshold for one mail type only, thus giving it a high probability of taking mail of the same type on consecutive occasions, minimising changeovers and the probability of its queue filling up. On the other hand, if long waiting times for other mail types are encountered, the strong stimuli may still cause changes in agents specialisations, and the population can adapt to meet the current level of demand in the system.

Further, two different types of environment are considered:

- A static environment, in which a city automatically produces a new batch of mail for every mail type that has been taken at the end of each time step.
- A dynamic environment, in which the probability of batches of mail type being produced varies over time.

The dynamic environment is specifically designed to test the flexibility of the system, and we have chosen to vary the probability of taken mail batches being produced in a sinusoidal fashion. All mail types have the same wavelength (e.g. to mimic seasonal variations), but have a different phase. Hence, all mail types have periods of both high and low production with certain mail types being dominant at some times and scarce at others. For both environments the probability of creating a taken batch of type m at the end of cycle t is given by:

$$\pi_m(t) = \begin{cases} 1, & \text{static} \\ \frac{1}{2}[1 + \sin(\frac{t2\pi}{\xi} - \frac{m2\pi}{N_m})], & \text{dynamic} \end{cases} \tag{2.1}$$

where ξ is the wavelength. In the dynamic environment an agent may be forced to compromise in its strategy of specialising in one type of mail as there will be periods during which its preferred mail type is rare.

It is clear that the problem is to minimise the loss of efficiency. Therefore, it is important to identify the different mechanisms that lead to efficiency loss. If an agent of effective specialisation σ_a fails to process mail during an iteration, this can be categorised into four cases:

($\boldsymbol{\ell}.1$) The agent is inactive due to a full mail queue at its processing centre.

($\boldsymbol{\ell}.2$) Mail type σ_a is available at the city, but the agent rejects all mail nonetheless.

($\boldsymbol{\ell}.3$) Mail type σ_a is not available at the city and the agent rejects all other mail.

($\boldsymbol{\ell}.4$) There is no mail at all available by the time of the agent's action.

While it is possible to minimise $\boldsymbol{\ell}.4$ by increasing $\boldsymbol{\ell}.1$ -$\boldsymbol{\ell}.3$ (i.e. by lowering the overall acceptance rate of mail), this is clearly not ideal. In particular, $\boldsymbol{\ell}.3$ and $\boldsymbol{\ell}.4$ are due to the non-uniform number of agents visiting cities. In the current model we have no control over this and focus mainly on the other sources. The most unnecessary source of efficiency loss is clearly $\boldsymbol{\ell}.2$ as the uptake of mail of its own specialisation has no negative consequences for an agent. A good update rule, therefore, should drive an agent's thresholds to a state in which θ_{a,σ_a} is very low. One should note that $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$ are finely balanced against each other as a greater uptake of mail of non-specialised types leads to an increase in agents with full queues. This relationship is clearly non-linear as a increase in the number of inactive agents leads to an increase in average stimulus, hence to an increase in the uptake of non-specialist mail and an even further increase in the number of inactive agents.

2.2 Methodology

As the self organising behaviour exhibited by social insects appears in (large) colonies, it seems natural to consider the performance of the model with a large population of agents. Not only does the large system size have the advantage of removing finite size effects (such as large fluctuations both inside and in between different runs), but Anderson et al. [11], [12] have also shown that a specific form of collective behaviour involving direct cooperation between agents is only efficient in large systems. Furthermore, Dornhaus et al. [13] hypothesise that the simulation of honeybee behaviour that they have investigated, did not produce realistic behaviour because of the small size of the system used. Hence, in this paper we consider the behaviour of the system mostly from a population dynamics perspective where the average behaviour is of greater importance than the individual performance of an agent. Nevertheless, we also investigate the influence of (small) system size on the overall effiency and fluctuations thereof.

2.2.1 Update Rules and Threshold functions

It is clear that for a given threshold function, the efficiency of an agent critically depends on its thresholds, while its flexibility to adapt to new situations critically depends on its ability to modify its thresholds. The strategy used by an agent to modify its thresholds is determined by its so-called *update rule*. One of the main goals of this paper is to investigate what kind of update rule is best suited to the problem at hand, and to investigate whether optimal update rules can be found autonomously by competition between the agents. Therefore, we compare the performance of some existing and some newly introduced update rules. We now proceed with a short overview.

The **Variable Response Threshold (VRT)** rule was proposed in [6] and was applied to the current problem in [4], [10]. The change in threshold $\Delta\theta_m$ over a period of time t, is given by:

$$\Delta\theta_m = -\varepsilon\Delta t_m + \psi(t - \Delta t_m) \tag{2.2}$$

where Δt_m is the time spent performing task m, where ε, ψ are positive constants, and where θ_m is restricted to the interval $[\theta_{min},\theta_{max}]$. For the current model, eq. (2.2) can be discretised, taking into account the fact that thresholds are only changed when a task is performed. Therefore, when the update rule is called, over a single time step, t will be 1 and Δt_m is 1 if mail type m was taken and 0 otherwise. Hence, the VRT rule can be rewritten as

$$u(\theta_m, i) = \begin{cases} \theta_m - \varepsilon & \text{if } i = m, \\ \theta_m + \psi & \text{otherwise.} \end{cases} \tag{2.3}$$

A drawback of the VRT rule is that the thresholds must be artificially restricted to the range $[\theta_{min},\theta_{max}]$. Additionally, in the event of a changeover, for small ε and ψ agents are unlikely to change their thresholds enough to have a high chance of picking the new mail type in the next time step, thus increasing $\boldsymbol{\ell}.2$. In order to overcome these flaws and to see if better efficiency could be obtained, we introduce some new update rules.
The **Switch-Over (SO)** rule restricts the thresholds to $[\theta_{min}, \theta_{max}]$ in a very simple manner by fully specialising in the most recently taken mail type, and fully de-specialising in all other mail types:

$$u(\theta_m, i) = \begin{cases} \theta_{min} & \text{if } i = m, \\ \theta_{max} & \text{otherwise.} \end{cases} \tag{2.4}$$

In some sense the switch-over rule can be seen as an extreme case of the VRT rule with $\varepsilon,\ \psi \geq \theta_{max} - \theta_{min}$. Such values, however, are not in the spirit of the VRT, and so it makes sense to consider them as separate cases. We introduce SO as we expect it to minimise $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.2$ in a static environment, while a drawback is that $\boldsymbol{\ell}.3$ could be maximised.

The **Distance Halving (DH)** rule, is another method for keeping the thresholds restricted to the appropriate range by halving the Euclidean distance between the current threshold and the appropriate limit, θ_{min} if the corresponding mail type has been taken and θ_{max} if it has not.

$$u(\theta_m, i) = \begin{cases} \frac{\theta_m + \theta_{min}}{2} & \text{if } i = m, \\ \frac{\theta_m + \theta_{max}}{2} & \text{otherwise.} \end{cases} \tag{2.5}$$

We introduce DH as we expect the agents to become generalists, thus potentially decreasing $\boldsymbol{\ell}.3$, as it takes several time steps to fully specialise, while it can effectively de-specialise in a single time step.

The **((Modified) Hyperbolic Tangent ((m)tanh)** rule is introduced in a similar spirit to the VRT rule, and allows for a more continuous variation of the thresholds whilst removing the need for artificial limits. Although SO and DH solve the problem of the artificial limitation of the thresholds, they always result in sharp changes in the thresholds. In the (m)tanh rule, the thresholds are a function of some hidden variables h_m:

$$\theta_m = \theta_{min} + (\theta_{max} - \theta_{min})\frac{1}{2}(1 + \tanh h_m) \tag{2.6}$$

By definition the θ_m are restricted to the appropriate range (although we have chosen the $\frac{1}{2}(1 + \tanh)$, it is clear that any sigmoid function would do the trick). The update rule now works on the hidden variables in a manner similar to the VRT:

$$u'(h_m, i) = \begin{cases} h_m - \alpha & \text{if } i = m, \\ h_m + \beta & \text{otherwise,} \end{cases} \tag{2.7}$$

for some positive constants α and β. As for the VRT, low α and β values may suffer from slow re-specialisation, however, the speed of re-specialisation is independent of θ_{min} and θ_{max}. Furthermore, the rule may lead to the problem of saturation. For large hidden variables, the tanh rule produces insignificant changes to the actual thresholds. As the whole basis of VRT-like models is that engagement in a task leads to an increased likelihood of repeating the task, this problem must be addressed.

Therefore, we modify the tanh rule by making a distinction between positive and negative hidden values, and introduce re-specialisation coefficient $\eta \in [-1, 1]$ such that

$$u'(h_m, i) = \begin{cases} h_m - \alpha & \text{if } i = m \text{ and } h_m \leq 0, \\ \eta\, h_m - \alpha & \text{if } i = m \text{ and } h_m > 0, \\ h_m + \beta & \text{if } i \neq m \text{ and } h_m \geq 0, \\ \eta\, h_m + \beta & \text{otherwise.} \end{cases} \tag{2.8}$$

It is identical to the tanh rule for $\eta = 1$, but is more similar to the SO rule for η close to -1.
The **Bienenstock, Cooper and Munro (BCM)** rule [14], proposed as a modification of the Hebbian learning rule for neurons, was designed as a mechanism for selectivity in the increase in synaptic weights. It takes into account the history of stimuli at a neuron, and updates synaptic weights based on both the history and the current stimulus. This method can be modified to an update rule compatible with the mechanism defined by eq. (2.8) which also addresses the problem of saturation. The thresholds are updated as follows:

$$u'(h_m, i, w) = \begin{cases} h_m - \nu(w(t) - H_{BCM}(t))w(t) & \text{if } i = m, \\ h_m - \nu(w(t) + H_{BCM}(t))w(t) & \text{otherwise,} \end{cases} \tag{2.9}$$

where $\nu > 0$ is a constant, $w(t)$ is the current stimulus and where the stimulus history $H_{BCM}(t)$, is defined as some function of the stimuli of recently taken mail averaged over a predefined number of batches (t_h). In this paper, we use the function as proposed in [14]:

$$H_{BCM}(t) = \sum_{i=1}^{t_h} w^{\tau}(t - i) \tag{2.10}$$

for some constant τ. The selectivity, which the system was designed to produce, is not a necessary feature of the update rule such that the constraint $\tau > 1$ can be removed. Initially, when the agent has not taken enough mail to fill the history, it will be assumed that the missing batches have 0 stimulus. Note that BCM does not strictly fit the definition (1.2) as selection of a piece of mail with a low waiting time can lead to an increase in the threshold for the mail type. It is, however, a well established strategy which we have included as an alternative to the mtanh rule. For well chosen parameters, the hidden variables should be kept at levels that are high enough to keep the current thresholds extremised, but low enough to allow mail with sufficient stimuli to cause a switch in specialisation.

It is clear that the overall efficiency and flexibility of an agent not only depends on its (capability to update its) thresholds, but also on threshold function $\Theta(s, \theta)$ itself. Therefore, we compare the performance of the standard and some newly introduced threshold functions. We now proceed with a short overview.

The **Exponential Threshold Function (ETF)**, is the standard threshold function as proposed by [5], and is defined as:

$$\Theta(s, \theta) = \begin{cases} \frac{s^\lambda}{s^\lambda + \theta^\lambda} & \text{if } s \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

which for $\lambda \geq 1$ has all the desired properties. Note that it has an exponential dependence on the parameter λ, which makes it very sensitive to small changes. In order to check whether the obtained results are not (partially) due to this sensitivity, and to see whether efficiency can be improved, we consider two other threshold functions. The **Scaling Threshold Function (STF)** is piecewise linear, and its slope depends on both the value of the threshold, and on $\lambda \in [0, 1]$ (its gradient relative to the threshold):

$$\Theta(s, \theta) = \begin{cases} 0 & \text{if } s \leq \lambda\theta \\ \frac{s - \lambda\theta}{2(1-\lambda)\theta} & \text{if } \lambda\theta < s < (2 - \lambda)\theta \\ 1 & \text{otherwise} \end{cases} \tag{2.12}$$

The **Gradient Threshold Function (GTF)** is also piecewise linear, and makes the slope (governed by the parameter λ) independent of the threshold:

$$\Theta(s, \theta) = \begin{cases} 0 & \text{if } s = 0 \text{ or } s \leq \theta - \frac{1}{2\lambda} \\ \lambda(s - \theta + \frac{1}{2\lambda}) & \text{if } \theta - \frac{1}{2\lambda} < s < \theta + \frac{1}{2\lambda} \\ 1 & \text{otherwise} \end{cases} \tag{2.13}$$

2.2.2 Genetic Algorithms

For any given environment, the behaviour of the system is governed by the update rules and threshold functions of the agents, and the parameters therein. Genetic algorithms (GA) seem a natural way of optimising the model as it is inspired by the behaviour of social insects, and have proven their worth in such settings: e.g. Bonabeau et al. [15] have used a GA to cause a stimergic system to build a structured architecture autonomously (where previously the system required guidance towards particular structures), while Campos et al.

[7] have successfully used a GA to optimise a threshold model on a similar problem, to name but a few.

Our motivation to employ a GA is threefold:
Firstly, rather than optimising all parameters manually (which is a daunting task due to the large number of possible combinations), we have opted to employ a GA. In particular for each individual update rule, we expect the GA to find parameters with the optimal trade-off between $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$.
Secondly, Ga's allow for inter-species (i.e. inter-update-rule) competition which enables us to find an optimal rule-set whilst we optimise the parameters rather than requiring us to optimise each rule-set and then choose the best one.
Finally, a GA can in principle provide novel self-organising behaviour, although completely novel behaviour is not expected due to the relative simplicity of the current model. Riolo [16] has shown how a GA will tend to a peak of fitness in the environment. As the system causes cooperation due to the modification of the environment, it is possible that the existence of one part of a population in one location of the fitness landscape will cause a new peak of fitness to emerge at another point and that a population with two (or more) clusters in these areas will lead to a better mean fitness.

In this paper, we use a similar approach to that of Holland [17]. However, in our case genes are not represented by bit-strings but are the set of parameters associated with each update rule. Hence, the mutation mechanism Holland describes has been replaced by addition of Gaussian noise to the parameter, and the cross-over mechanism merely consists of swapping constants, which is unordered due to the arbitrary arrangement of the genes. As different update rules have different (numbers of) parameters, inter-species cross-over does not occur.

The GA assigns to each agent in the population an update rule and a threshold type which are referred to collectively as an agent's species. Each agent is then assigned the full set of (randomly or otherwise initialised) parameters, which are referred to as its genes. The agent's variables are then initialised and the algorithm will be run for a set number of time steps.

A new generation of agents is bred by selecting an agent from the initial population and choosing to breed it with a probability given by its fitness. This parent agent will either copy itself to the new generation or undergo a cross-over with probability p_c. In the case of a cross-over, a second parent agent (from the same species) is chosen in a similar manner. The first cross-over member of the new generation randomly selects its genes independently from each of its parent agents with probability 0.5, while the second takes the unpicked genes. We also initialise a small proportion of the new generation randomly in order to avoid the persistence of a suboptimal population due to low diversity. Once a new generation of the same size has been created each gene of each new agent mutates with probability p_m by adding some Gaussian noise of zero mean and small variance. This process is repeated until some predefined condition is met.

3 Theoretical Analysis

In general, the mail retrieval problem is hard to solve exactly as it depends on continuous variables (the thresholds), such that the number of micro-states of the agents is infinite (not countable). In other contexts where this problem occurs, such as continuous models on sparse random graphs (see e.g. [18] and references therein), population dynamics can be used for the theoretical analysis. For agent based models, however, this is paramount to simulating the model. On the other hand agent based models are theoretically solvable when the number of discrete micro-states is finite. It turns out that we can analyse some non-trivial cases (combination of update rule and threshold function) of the current model exactly. Furthermore, using similar (but much simpler) techniques we can derive theoretical upper bounds for the efficiency for any update rule/threshold function.

For the SO rule, specialised agents only have thresholds in $\{\theta_{min}, \theta_{max}\}$, and thresholds are entirely determined by the effective specialisation. Therefore, a micro-state $\mathcal{A}$ of an agent is determined by the thresholds $\vec{\theta}$ associated with its specialisation σ, and the state $\vec{q}_L$ of the mail queue at its processing centre:

$$\mathcal{A} \equiv (\vec{\theta}, \vec{q}_L) \tag{3.1}$$

where $L \in \{0, .., L_q\}$ is the length of the queue and $\vec{q}_L = \{q_1, .., q_L\}$ are the remaining processing times. Note that if an agent is specialised in mail type m then $\theta_m = \theta_{min}$ while all other $\theta_n = \theta_{max}$, and that $q_1 \in \{1, .., t_c\}$ and $q_i \in \{t_p, t_c\}$ $(i > 1)$, as their processing has not yet started. Hence, the set $\mathcal{S}_\mathcal{A}$ of all possible agent micro-states has cardinality $|\mathcal{S}_\mathcal{A}| = N_m \; (1 + t_c \sum_{L=1}^{L_q} 2^{L-1}) = N_m \; (1 + t_c(2^{L_q} - 1))$.

The micro-states $\mathcal{C}$ of the city are already discretised, and consist of the waiting times of the mail types:

$$\mathcal{C} = (\vec{w}_{N_m}) = \{w_1, .., w_{N_m}\} \tag{3.2}$$

Although the number of such states is infinite (waiting times have no upper limit), this is not a problem when the threshold function is such that $\Theta(w,\theta) = 1,\ \forall w > \theta_{max}$. Only states in $\{0,..,\theta_{max},(>\theta_{max})\}^{N_m}$ need to be considered, and the set $\mathcal{S}_\mathcal{C}$ of all possible city micro-states has cardinality $|\mathcal{S}_\mathcal{C}| = (\theta_{max}+2)^{N_m}$. For the threshold functions that we have considered, this is the case for sufficiently high λ (e.g. GTF with $\lambda > 0.5$) because then:

$$\Theta(w,\theta) = \begin{cases} 0 & \text{if } w < \theta \text{ or } w = 0, \\ 0.5 & \text{if } w = \theta \text{ and } w \neq 0\ , \\ 1 & \text{if } w > \theta. \end{cases} \tag{3.3}$$

Defining the states of the agents as $\vec{s}(t) = \{s_a(t), a = 1,..,N_a\}$ and the states of the cities as $\vec{S}(t) = \{S_c(t), c = 1,..,N_c\}$, at any time t the global state of the system is completely determined by the agent profile $\vec{\mu}(t) = \{\mu_\mathcal{A}(t),\ \mathcal{A} \in \mathcal{S}_\mathcal{A}\}$ and city profile $\vec{\eta}(t) = \{\eta_\mathcal{C}(t),\ \mathcal{C} \in \mathcal{S}_\mathcal{C}\}$, where

$$\mu_\mathcal{A}(t) \equiv \frac{1}{N_a}\sum_{a=1}^{N_a} \delta_{s_a(t),\mathcal{A}}\ , \qquad \eta_\mathcal{C}(t) \equiv \frac{1}{N_c}\sum_{c=1}^{N_c} \delta_{S_c(t),\mathcal{C}}\ . \tag{3.4}$$

In the large system limit, as a consequence of the Central Limit Theorem, $\vec{\mu}(t)$ and $\vec{\eta}(t)$ become deterministic quantities, for which we can derive the exact time evolution. It is convenient to break up the time evolution into four distinct steps:

a.1 changes to the $\vec{\mu}$ during mail uptake.

c.1 changes to the $\vec{\eta}$ during mail uptake.

a.2 changes to the $\vec{\mu}$ during processing of the queue.

c.2 changes to the $\vec{\eta}$ during mail production.

During the mail uptake the change in the agent profile can be described by multiplication with a matrix $\mathbf{T}(\vec{\mu}(t),\vec{\eta}(t))$ which explicitly depends on both $\vec{\mu}(t)$ and $\vec{\eta}(t)$ due to the competition between agents at the cities. The change to the agent profile during the processing of the queue, can be described by multiplication with a constant matrix $\mathbf{Q}$. The change in city profile during mail uptake can be described by multiplication with a matrix $\mathbf{L}(\vec{\mu}(t))$, which explicitly depends on $\vec{\mu}(t)$ due to competition between the agents. Finally the change in city profile during mail production can be described by multiplication with a matrix $\mathbf{P}(t)$ which is time dependent for the dynamic environment only. Combined, this leads to the following exact time evolution:

$$\begin{cases} \vec{\mu}(t+1) & = \mathbf{Q}\quad \mathbf{T}(\vec{\mu}(t),\vec{\eta}(t))\ \vec{\mu}(t)\ , \\ \vec{\eta}(t+1) & = \mathbf{P}(t)\ \mathbf{L}(\vec{\mu}(t))\qquad \vec{\eta}(t)\ . \end{cases} \tag{3.5}$$

As the derivation and exact expressions of the matrices **T**, **Q**, **L** and **P** are rather involved, we refer those to appendix A. The theoretical time evolution and numerical simulations are compared in the following section, and are in excellent agreement.

The exact solution of the dynamics of the model is only possible for the SO update rule and a threshold function with $\Theta(s, \theta) = 1,\ \forall s > \theta_{max}$. However, following a strategy similar to the one above we can derive update rule- and threshold function- independent theoretical upper bounds for the efficiency of an infinite population in *ideal circumstances*, i.e. when no mail is lost due to $\boldsymbol{\ell}$.1 -$\boldsymbol{\ell}$.3 . This situation would occur when $t \leq L_q$ and when agents *never* reject mail such that the efficiency is only limited by $\boldsymbol{\ell}$.4 . Then, both the agent profile and the mail waiting times become irrelevant and the efficiency is a function of the profile of the following simplified city micro-states:

$$\mathcal{C} = \vec{b}_{N_m} = \{b_1, .., b_{N_m}\}\ , \tag{3.6}$$

where $b_i \in \{0, 1\}$ is the availability of mail type i at the city. The set $\mathcal{S}_\mathcal{C}$ of all possible states has cardinality $|\mathcal{S}_\mathcal{C}| = 2^{N_m}$. Defining the states of the cities as $\vec{S}(t) = \{S_c(t), c = 1, .., N_c\}$, at any time t the global state of the system is completely determined by the city profile $\vec{\chi}(t) = \{\chi_\mathcal{C}(t),\ \mathcal{C} \in \mathcal{S}_\mathcal{C}\}$, where

$$\chi_\mathcal{C}(t) \equiv \frac{1}{N_c} \sum_{c=1}^{N_c} \delta_{S_c(t),\mathcal{C}}\ . \tag{3.7}$$

The change in city profile during mail uptake can be described by multiplication with a matrix $\mathbf{L}'$, while the change in city profile during mail production can be described by multiplication with a matrix $\mathbf{P}'(t)$ which is time dependent for the dynamic environment only. Combined, this gives the following exact time evolution for the city profile:

$$\vec{\chi}(t+1) = \mathbf{P}'(t)\ \mathbf{L}'\ \vec{\chi}(t)\ . \tag{3.8}$$

Then, the efficiency $E(t)$ (the probability that an agent takes mail at time t) is given by

$$E(t) = \sum_{k=1}^{N_m} \chi_k(t) \left(1 - P_{R_{a/c}}(k) + \frac{k - R_{a/c}}{R_{a/c}}(1 - \sum_{i=0}^{k} P_{R_{a/c}}(k))\right)\ , \tag{3.9}$$

where $\chi_k(t) \equiv \sum_{\vec{b} \in \mathcal{S}_\mathcal{C}} \chi_{\vec{b}}(t) \delta_{|\vec{b}|,k}$ is the probability that a city has exactly k pieces of mail available, P_λ is the Poisson distribution with parameter λ, and $R_{a/c}$ is the ratio of agents to cities. The details of these derivations and the exact expressions of the matrices $\mathbf{L}'$ and $\mathbf{P}'$ can be found in appendix A. A comparison between the performance of the various update rule/threshold function combinations with this theoretical upper bound is presented in the following section.

4 Results

In this section we discuss the numerical results. First, we describe the general tendencies of how the efficiency and loss sources depend on the model parameters such as the system size N_a, the agent to mail ratio $R_{a/m}$, the number of mail types N_m, and the wave length ξ for the dynamic environment. It turns out that qualitatively the general tendencies are rather insensitive to the choice of update rule and threshold function, and their respective parameters (as long as these are chosen reasonably). Therefore, we present these for the SO rule and the ETF threshold function, and only mention other rules if they exhibit qualitatively different behaviour.

The second part of this section is dedicated to the optimisation of the parameters, and the selection of the best possible combination of update rules and threshold functions in terms of the overall efficiency.

4.1 General tendencies

As there are many parameters to cover, we have opted to investigate the influence of different factors on the efficiency systematically, by varying one parameter at a time and keeping the rest in *the standard setting*: unless specified otherwise we simulate the system with $N_a = 5 \times 10^4$ agents and $N_m = 2$ mail types, using the ETF threshold function with $\lambda = 2$. In order to have a fair comparison between different environments, we take $R_{a/m} = 1$ in a static environment, while in the dynamic environment we take $R_{a/m} = 0.5$ (as $\overline{\pi_m(t)} = 0.5$ over a period). We fix the various parameters to the following values: $\theta_{min} = 0$, $\theta_{max} = 50$, $\varepsilon = \psi = 5$, $\alpha = \beta = \eta = 0.5$, and for the BCM rule $\nu = \tau = 1$ and $t_h = 20$. A standard run consists of 500 iterations over which the average efficiency per agent is monitored, and the standard dynamic environment has a period $\xi = 50$. Note that all simulations are implemented in C++ and are performed on a linux-PC cluster.

In order to investigate the dependence of our results on the system size, we we vary N_a, while keeping N_m and $R_{a/c}$ fixed. In general we find that the average of any measured quantity quickly ($N_a \simeq 5 \times 10^2$) converges to its asymptotic value (for $N_a = \infty$) with increasing system size, while both inter- and intra-run variance decreases to below the line width of the plots at values $N_a \simeq 10^4$. With these findings in mind, we have decided to fix the system size at $N_a = 5 \times 10^4$, for which simulations can be run in reasonable time, and for which a single run suffices to determine any quantity with sufficient accuracy, omitting the need for error-bars in most figures that follow.

Only for genetic algorithms we use repeat runs when the algorithm is particularly sensitive to the emergence of particular individuals (a sensitivity that is not directly related to the system size). The significant finite size effects for relatively small system sizes ($N_a \leq 10^2$) are illustrated in Figure 1. The increased average efficiency for small values of N_a can be explained by considering that each agent on average competes with $(N_a - 1)/N_c$ agents, which is monotonically increasing with N_a for fixed $R_{a/m}$ and N_m and saturates for high N_a values.

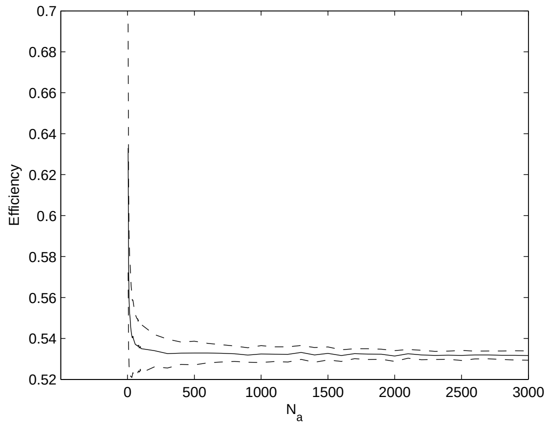

Figure 1. The average efficiency (solid line) as a function of the system size N_a with fixed $R_{a/m} = 1$ and $N_m = 2$, in a static environment, using the tanh update rule, averaged over 1000 runs ($2 \leq N_a \leq 30$), 500 runs ($30 \leq Na \leq 200$), or 50 runs ($N_a > 200$), and error bars of ± 1 s.d. (dashed lines).

As a rule of thumb, we consider an agent to be fully specialised in a mail type if its threshold for this type is less than a distance of 1% of the possible range from θ_{min} while all other thresholds are within 1% of θ_{max}. The qualitative behaviour of the mtanh rule, as shown in Figure 2 (top), is typical for all update rules although the speed of convergence, and the asymptotic values depend on both the update rule and threshold function. We see that the algorithm accounts for the genesis of specialisation. The system tends towards a stable asymptotic regime in which most agents are specialised and the specialists are equally split between mail types. The fact that $\boldsymbol{\ell}.1$

is almost negligible while we still have some $\boldsymbol{\ell}.2$ is indicative of the high value of θ_{max}. With the S0 rule and $\theta_{min} = 0$, $\boldsymbol{\ell}.2$ becomes impossible once an agent has taken a piece of mail. Agents with all initial thresholds close to θ_{max} may never, over the course of a run, encounter a batch of mail with a strong enough stimulus to accept it.

In the dynamic environment (see Figure 2 bottom), we observe variations in efficiency over the course of a wavelength. The positions of the minima and maxima may at first sight seem strange as the total probability for mail production remains static and the points of maximum efficiency occur where a non-uniform distribution of mail is expected. However, at the end of an iteration in which mail type m is predominantly produced it is also more likely for this mail type to be left over. This in turn lowers the probability of this mail type being produced in the next iteration compared to when $\pi_1(t) \approx \pi_2(t)$. Hence, while the a priori total probability of mail production probability is static, the *effective* total probability of mail production is maximal when $\pi_1(t) - \pi_2(t) \to 0$. The qualitative behaviour of efficiency and losses, as shown in Figure 2 for the SO rule, is typical for most update rules. However, the specialisation that drives this behaviour varies between rules. In particular, those rules based on hidden variables ((m)tanh, BCM) tend to specialise in a manner similar to the mtanh rule, while the other algorithms (VRT, SO and DH) behave qualitatively similarly to the SO rule.

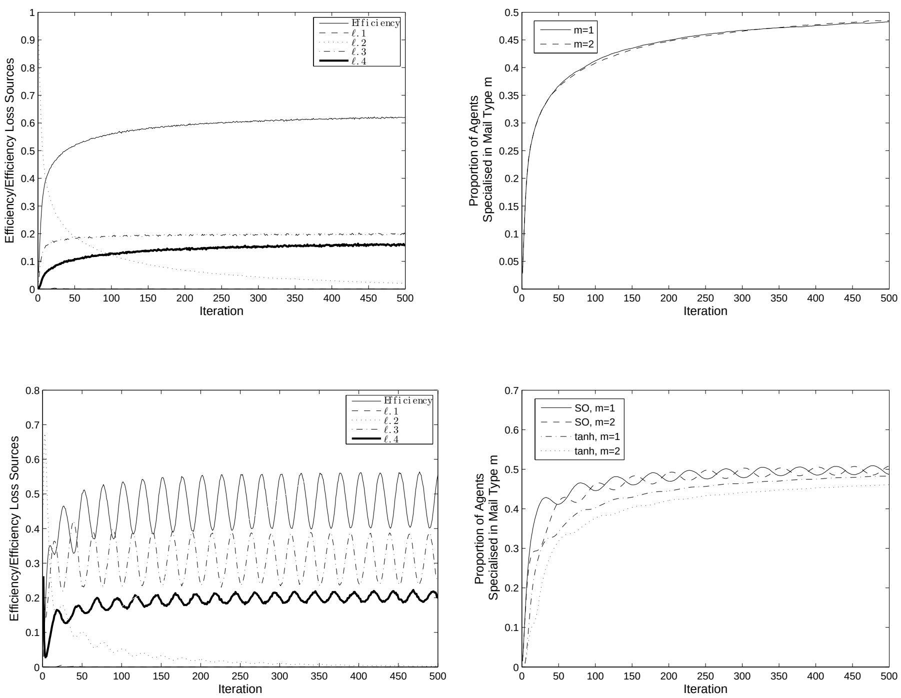

Figure 2. Top left: evolution of the efficiency and loss sources during a single run in the standard static environment using the SO rule. Note that $\boldsymbol{\ell}.1$ is negligible everywhere and $\boldsymbol{\ell}.2$ tends to 0, while $\boldsymbol{\ell}.3$ and $\boldsymbol{\ell}.4$ (and hence the efficiency) quickly tend to their long time values. Top right: the population of agents tends towards an equal split in specialisation with almost all agents specialised. Bottom left: evolution of the efficiency and loss sources during a single run in the standard dynamic environment using the SO rule. The values of the loss sources and the efficiency fluctuate around their *average* values, which are qualitatively similar to those in the static environment. Bottom right: the difference in specialisation behaviour between the SO and the tanh rule. Note that the tanh rule tends to a static, uneven (initial condition dependent) set of specialisations, while the SO rule efficiently adapts to changes in the environment.

The tanh function, like any sigmoid function, is effectively constant (i.e. saturated) for sufficiently large arguments. The saturation region is reached when an agent using the tanh update rule repeatedly takes the same mail type. Once in this region, the update rule becomes incapable of effective self reinforcement on which the VRT model relies, and incapable of reacting to changes in the environment. A similar problem can be encountered in neural networks with sigmoidal nodes, in which Hebbian learning drives synaptic weights into the saturation region of the function rendering the relative sizes of these weights meaningless [19] thus removing the selectivity of the node. The BCM rule, which was designed to deal with this lack of selectivity, attempts to keep the hidden thresholds at relatively low levels and thus avoid saturation. We follow a slightly different strategy with the mtanh rule for which the hidden thresholds are allowed to increase arbitrarily, but can also decrease quickly when necessary, thus rendering self reinforcement practical again.

The tanh rule is the only rule inherently unable to dynamically adapt its thresholds due to the saturation effects described above, while the other rules can do so if given suitable parameters (i.e. a lowering of η in the mtanh rule). To highlight the effects of saturation for the tanh rule, we let the system equilibrate for 1500 iterations in the standard static environment to allow agents to specialise fully. Then we remove that half of the population that is most specialised in e.g. mail type 2, and equilibrate the remaining system (with halved $R_{a/m}$) for a further 1500 iterations. The results, shown in figure 3, show that in contrast to the SO rule, which adapts very quickly to the change, none of the specialised agents using the tanh rule re-specialise.

Stability is reached when the average stimulus of mail type 2 reaches a high enough level to force changeovers a significant proportion of the time, leading to high levels of $\boldsymbol{\ell}.1$. The mtanh rule is able to somewhat adapt to this by lowering the thresholds of agents taking type 2 mail most often, causing some to re-specialise. Once enough agents are re-specialised that the average stimulus of type 2 mail drops to a level where frequent changeovers are unlikely the re-specialisation slows significantly meaning that an optimal set of specialisations will not be regained.

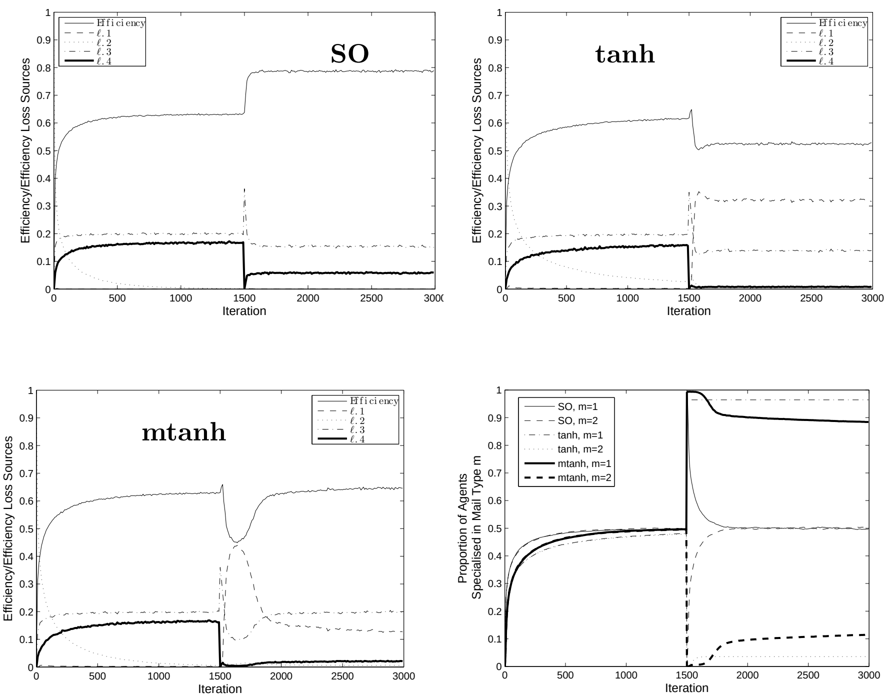

Figure 3. Efficiency and loss sources in a static environment with removal of specialised agents. The SO rule (top left) almost immediately returns to the optimal split in specialisations. Due to saturation, the tanh rule (top right) is unable to re-specialise, resulting in a dramatic increase in consecutive changeovers ($\boldsymbol{\ell}$.1). Although the mtanh rule (bottom left) is capable of re-specialising, it does so far less efficiently than the SO rule and initially reacts similarly to the tanh rule. Bottom right: evolution of the fraction of specialised agents for the various rules.

In figure 4, we compare the upper bound with the actual efficiency and the loss sources of the SO rule (for $N_m = 2$), as a function of $R_{a/m}$. In the static environment (left), the difference in efficiency at low $R_{a/m}$ is due to high average waiting times which become close enough to θ_{max} to overwhelm agents' selectivity and force multiple changeovers and high levels of $\boldsymbol{\ell}$.1 . At high values of $R_{a/m}$ it is clear that θ_{max} is too high as we know that a smaller population could serve the demand, such that a drop in $\boldsymbol{\ell}$.1 would be acceptable in order to decrease $\boldsymbol{\ell}$.2 and $\boldsymbol{\ell}$.3 . The behaviour in the dynamic environment (right) is similar to that of the static environment. However, the variable nature of the environment causes an increase in both $\boldsymbol{\ell}$.1 and $\boldsymbol{\ell}$.3 as the agents' specialisation, being reactive, lags behind the state of the environment. Agents that fail to react cause an increase in $\boldsymbol{\ell}$.3 , while agents that do react, must undergo a changeover thus increasing **ℓ**.1 . The decrease in **ℓ**.2 in the dynamic environment is merely due to an increase in average stimulus as a consequence of the overall decrease in efficiency.

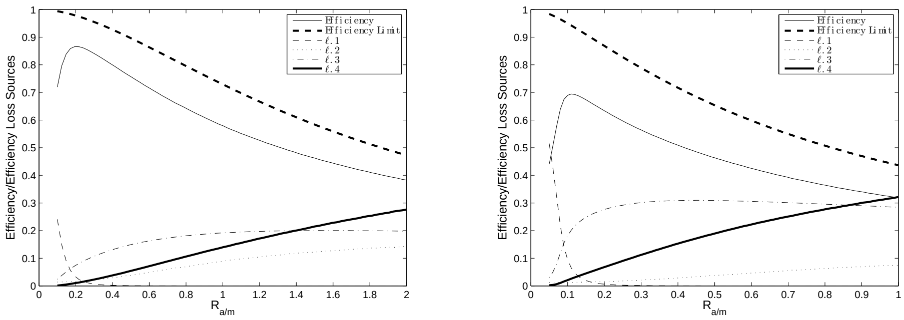

Figure 4. : efficiency and loss sources of the SO rule as a function of $R_{a/m}$ for $N_m = 2$, for the static (left) and dynamic (right) environment. With the exception of very low values of $R_{a/m}$, where **ℓ**.1 dominates, the efficiency follows the same trend as the theoretical upper bound. As expected loss sources **ℓ**.2 -**ℓ**.4 increase with $R_{a/m}$ while **ℓ**.1 becomes negligible. In the dynamic environment, the increases in **ℓ**.1 and **ℓ**.3 are more pronounced.

Figure 5 shows the efficiency as a function of N_m. At low values of N_m the efficiency initially increases due to the distribution of agents between cities becoming more uniform, while at high values of N_m this is offset by an increase in **ℓ**.1 . Agents have to examine more mail before they find their specialised type leading to increased chances of changeovers. Note that we do not compare efficiency with the theoretical limit as the assumptions used to derive the limit (such as a low frequency of switch-overs) are completely unrealistic in this case.

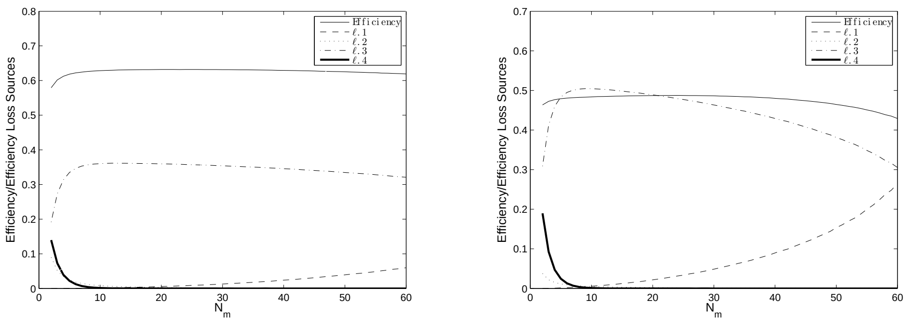

Figure 5. efficiency and loss sources of the SO rule as a function of N_m for $R_{a/m} = 1$ in the static (left), and $R_{a/m} = 0.5$ in the dynamic (right) environment. Loss sources **ℓ**.2 and **ℓ**.4 tend to 0 as N_m increases. Effi-

ciency initially improves before increases in $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$ cause it to decrease. In the dynamic environment, $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$ are increased, while $\boldsymbol{\ell}.2$ is reduced.

Figure 6 shows the average efficiency and loss sources as a function of the wave length for the dynamic environment. The peak in average efficiency at relatively short wavelengths has the same origin as the peaks of instantaneous efficiency seen inside a single run in the dynamic environment: persistence of mail. At short wavelengths the state of the environment changes so quickly that left over mail from the previous iteration is less likely to be of the type that is predominantly currently produced. Hence, the *effective* mail production is increased. The relatively low efficiency at the initial value $\xi = 2$, is caused by the discrete nature of the iterations. The sine wave part of equation 2.1 becomes $\sin((t-m)\pi) = 0$ for $N_m = 2$, such that the mail production probability probability is effectively constant with $\pi_m = 0.5$.

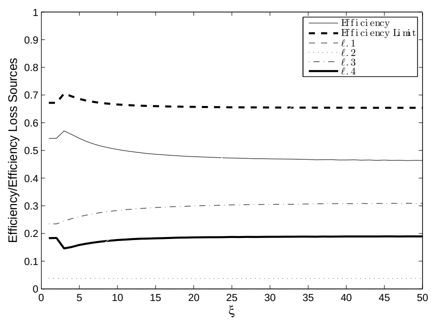

Figure 6. : efficiency and loss sources in the dynamic environment as function of the wavelength ξ. After an initial increase in efficiency (matching that of the theoretical limit) due to a decrease in $\boldsymbol{\ell}.4$, it then gently decreases to the long wave length value, due to a increase in $\boldsymbol{\ell}.3$ and $\boldsymbol{\ell}.4$. Both $\boldsymbol{\ell}.2$ and $\boldsymbol{\ell}.1$ (which is negligible) are virtually independent of ξ.

4.2 Genetic optimisation

As explained in section 2.2.2, we employ a Genetic Algorithm (GA) to obtain the optimal parameters for the various update rules and threshold functions. In our simulations, a standard generation of the GA consists of 500 iterations over which the average efficiency of each agent is monitored which then counts as its fitness. It turns out that for all update rules $\theta_{min} = 0$ is optimal, and that the optimised SO rule outperforms the other update rules in virtually all circumstances. The only update rules that can compete with it are those that can effectively mimic its behaviour by extreme choices of parameters. As this against the spirit of the nature inspired VRT rule, we have limited its parameters to $\varepsilon, \psi < \frac{\theta_{max}}{2}$. Otherwise, we leave all other parameters unconstrained and generate an initial population with equal proportions of all update rules and threshold functions.

The GA then optimises the different species of agents, while at the same time letting them compete with each other. In the static environment (see figure 7 top), the GA quickly finds a good tradeoff between $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$. This is obtained by dropping θ_{max} to a much lower value than intuitively expected (and used in the standard setting). The remaining efficiency gain is mainly a consequence of the increasing fraction of the population with a good rule set. Note that the BCM update rule quickly tends to extinction due to its strong parameter sensitivity, such that so only a small fraction of its starting population has good fitness. Even with continuous re-introduction of BCM agents the population does not recover, which shows that it is unlikely to be the best rule in this environment. The two best update rules are the SO and the mtanh rule (with $\eta < 0$ and large ε, ψ, thus approximating SO). The SO rule, however, has the added advantage of not being able to mutate away from this behaviour. All other rules tend to extinction due to suboptimal efficiency.

The threshold function populations evolve for reasons almost exactly opposite to the extinction of BCM. The STF increases initially due to a higher proportion of its initial population having reasonable fitness as, for this threshold function, high θ_{max} and low λ, minimising $\boldsymbol{\ell}.1$ while having a reasonable chance of low $\boldsymbol{\ell}.2$. This reasonable but suboptimal behaviour leads to less of a drive towards the optimal behaviour seen in the other two rules, with *low* θ_{max} and *high* λ, almost a step function at θ_m. This means that while the subsection of the STF population with optimal behaviour is increasing, the decrease in the suboptimal subsection is shared between each of the functions giving an overall decrease in the proportion. Overall, it is clear that the GTF is most suited to becoming a step function and least suited to suboptimal θ_{max} which explains it having the largest initial decrease followed by the greatest rate of increase.

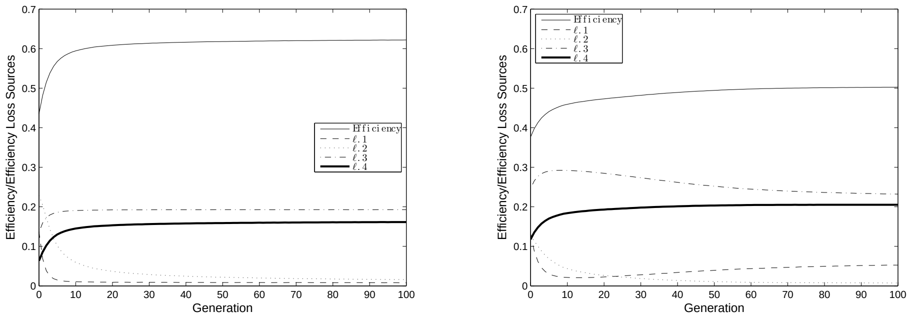

Figure 6. : evolution of the efficiency and loss sources during a GA optimisation of the population of agents in the static (left) and dynamic (right) environment. Efficiency is optimised by balancing the various loss sources.

The setup used for the GA in a dynamic environment was identical to that used in a static environment and the initial generations show almost identical behaviour, albeit with slightly higher θ_{max} due to the higher average waiting times. However, in this environment an improved tradeoff between $\boldsymbol{\ell}.1$ and $\boldsymbol{\ell}.3$ is possible with higher θ_{max} and lower λ, which decreases $\boldsymbol{\ell}.3$ to give a higher chance of keeping up with the state of the environment while keeping a reasonable chance of avoiding unnecessary changeovers and so keeping $\boldsymbol{\ell}.1$ to a reasonable level. The update rules are largely independent of this strategy, such that SO & mtanh with negative η still give the best performance. As the STF is least likely to mutate away from this strategy, it becomes the dominant threshold function.

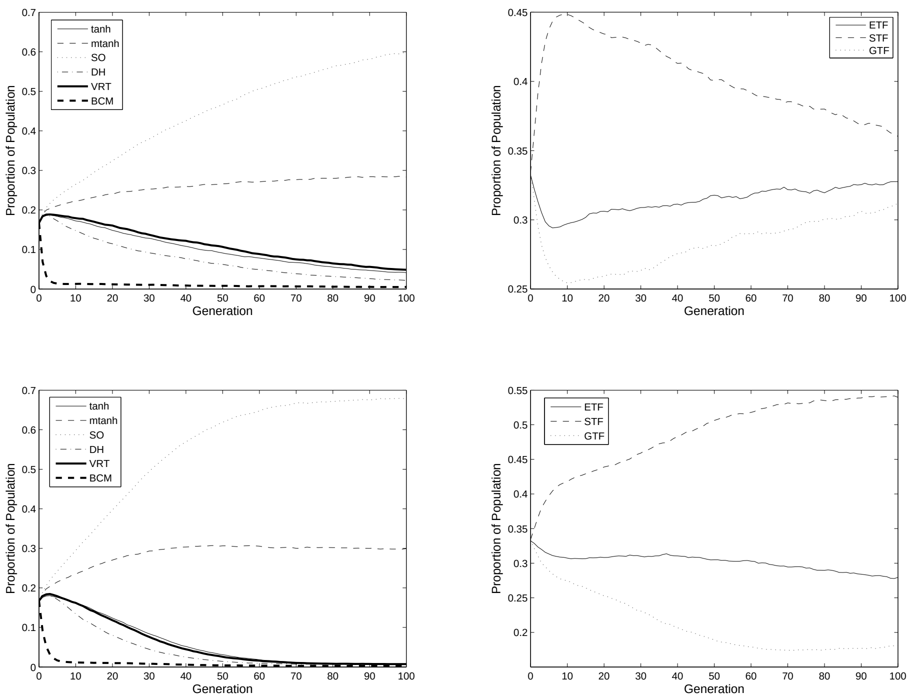

Figure 7. : Evolution of the fractions of agents using the various update rules (left) and threshold functions (right) for the static (top) and dynamic (bottom) environment. In both environments, all update rules eventually tend to extinction, except the mtanh and SO rule which effectively become the same. The relative fractions are determined by the initial conditions and sensitivity to mutations.
In the static environment the optimal threshold function is basically a step function $H(\theta_{max} - s)$, i.e. the Heaviside function. The continuing evolution of the relative fractions is due to a varying sensitivity to mutations. In the dynamic environment the optimal threshold function is the STF with finite slope. The suboptimal ETF and GTF become step functions, and eventually die out.

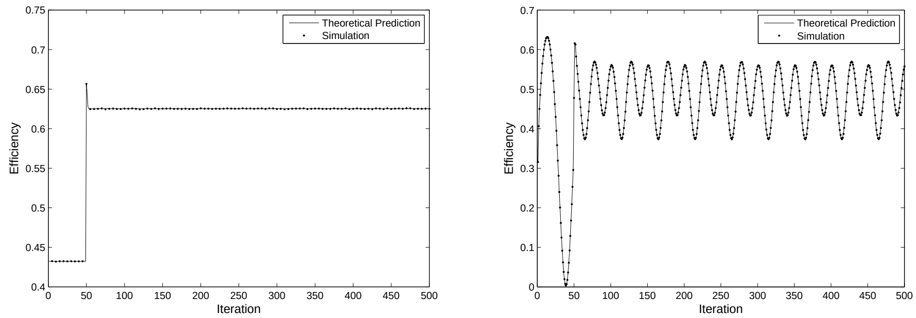

Figure 8. : comparison of the theoretical solution (lines) for an infinite system, with simulations (dots) in the standard static (left), and the dynamic (right) environment. Note that all agents were initialised with specialisation in mail type 1.

In figure 8, we show the excellent agreement between the exact theory for infinite system size and simulations of a large but finite population with corresponding settings. Note that we have opted to show the efficiency, but any other quantities such as fractions of specialised agents can also be calculated and are in equally good agreement.

	static	static final	dynamic	dynamic final
VRT	0.501 (0.687)	0.555 (0.761)	0.412 (0.630)	0.451 (0.690)
SO	0.586 (0.804)	0.623 (0.854)	0.467 (0.714)	0.485 (0.742)
GA	0.626 (0.858)	0.632 (0.867)	0.508 (0.777)	0.509 (0.779)
Theory	N/A	0.633 (0.868)	N/A	0.504 (0.771)

Table 1: The average efficiency of the different methods in the static and dynamic environment and the average efficiency after convergence (final), both in absolute numbers and as a fraction of the theoretical upper bound (in brackets). The SO rule already provides a large improvement over the VRT rule, while the genetically determined rules and parameters increase the speed with which high efficiency is reached but have less effect on the final efficiency. The best results obtained by the theoretical model are very close to the final results of the GA.

In Table 1, we illustrate the effect on the efficiency made by the introduction of new update rules and genetic optimisation in comparison to the original VRT model, which already outperforms a range of other general purpose algorithms [4, 10]. Efficiencies are averaged over 500 iterations (including the initial specialisation period), while final efficiencies are averaged over 100 subsequent iterations. GA results are given using the best performing rule sets at the end of 100 generations (SO update rule with the GTF in the static environment and the STF in the dynamic environment) with parameters taken averaged within their "species" in the final generation. The theoretical results are given for the optimal values of θ_{max}, which were determined by exhaustive search. In particular in the static environment, the comparison between the final GA results and the theory shows that the GA leads to a combination of update rule and threshold function that is approximately that for which theoretical results can be derived. Moreover, it finds approximately the same θ_{max} and has nearly identical efficiency. In the dynamic environment, however, the best GA rule set uses a different threshold function and has slightly improved efficiency.

5 Conclusions and Outlook

In this paper, we have studied an agent based model for distributed mail retrieval. The efficiency and flexibility have been investigated both in static and dynamic environments. We have introduced new rules for mail selection and specialisation and have used a genetic algorithm to optimise these further. We have shown that some of the new rules have improved performance compared to existing ones. The best ones give increased efficiency of 24.8% in a static, and 23.3% in a dynamic environment, compared to a method (VRT) which already outperformed a variety of other algorithms [10]. Nevertheless the performance may still be limited by our choice of the functional forms of the new rules.

We have shown that a nature inspired update rules such as the VRT can be competitive in all environments, especially when used in combination with a genetic algorithm to optimise its parameters. Nevertheless, it can be outperformed by specialist rules in each environment, and exhibits a lack of robustness against random mutations. Similarly, we have shown that the nature inspired ETF is a competitive strategy in all environments but can again be outperformed by specialist rules in each environment.

We have introduced a new dynamical environment to measure the flexibility of the various rules in terms of the efficiency in that environment. Furthermore we have systematically investigated the influence of the various model parameters such as system size, number of mail types, ratio of agents to cities, and wave length. We have identified the various loss sources, and have demonstrated that the random choice of cities to visit by the agents forms the main limitation on the maximal attainable efficiency, and we have derived this limit theoretically. We have demonstrated that a near optimal strategy can be *exactly* analysed theoretically in the large system limit, and we have validated the analytical solution with experimental results.

Although speciation and extinction do occur in the current model using a genetic algorithm, proper self-organising behaviour such as cooperation between the agents is not observed. The main limitation of the current model is again the random choice of cities which does not really allow agents to develop cooperative strategies, and direct competition is the only driving force behind the evolution of species. Therefore, a study of the model in which agents can adapt their preference to return to certain cities (memory effects) is in progress. Also in progress is a version of the model with genetic programming, in which agents are allowed to develop their strategies completely freely only driven by genetics, without us imposing any functional form.

Acknowledgements

The authors would like to thank P. Tiňo for introducing us to the problem, and for stimulating discussions and helpful suggestions, and J. P. Neirotti for stimulating discussions and suggestions.

References

[1] P. Janacik, T. Haimfarth, and F. Rammig. Emergent topology control based on division of labour in ants. In *Proceedings of the IEEE 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, 2006.

[2] P. P. Grassé. La reconstruction du nid et les interactions interindividuelles chez les bellicositermes natalenis et cubitermes sp. la théorie de la stigmergie: essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6:41–83, 1959.

[3] E. Bonabeau, M. Dorigo, and G Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 1999.

[4] R. Price and P. Tiňo. *Parallel Problem Solving from Nature VIII*, chapter Evaluaton of Adaptive Nature Inspired Task Allocation Against Alternate Decentralised Multiagent Strategies, pages 982–990. Springer Berlin / Heidelberg, 2004.

[5] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg. Fixed response thresholds and the regulation of division of labour in insect societies. *Bull. Math. Biol.*, 60:753–807, 1998.

[6] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg. Response threshold reinforcement and division of labour in insect societies. In *Proc. Roy. Soc. London B 265*, pages 327–332, 1998.

[7] M. Campos, E. Bonabeau, G. Thraulaz, and J.-L. Deneubourg. Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, 8(2):83–92, 2001.

[8] S. Nouyan, R. Ghizzioli, M. Birratari, and M. Dorigo. An insect-based algorithm for the dynamic task allocation problem. Technical Report TR/IRIDIA/205-031., Universit Libre de Bruxelles, 2005.

[9] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. In *Biocomputing and Emergent Computation*, pages 36–45, 1997.

[10] R. Price. Evaluaton of adaptive nature inspired task allocation against alternate decentralised multiagent strategies, 2004.

[11] C. Anderson and F. Ratnieks. Task partitioning in insect societies. i. effect of colony size on queueing delay and colony ergonomic efficiency. *The American Naturalist*, 154(5), 1999.

[12] C. Anderson and F. Ratnieks. Task partitioning in insect societies. ii. use of queueing delay information in recruitment. *The American Naturalist*, 154(5), 1999.

[13] A. Dornhaus, F. Klügl, F. Puppe, and J. Tautz. Task selection in honeybees - experiments using multi-agent simulation. In *Proc of GWAL'98, Bochum*, 1998.

[14] E. L. Bienenestock, L. N. Cooper, and P. W. Munro. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Nuroscience*, 2(1):32–48, 1982.

[15] E. Bonabeau, S. Gueŕin, D. Snyers, P. Kuntz, and G. Theraulaz. Three-dimensional architectures grown by simple stigmergic agents. *Biosystems*, 56(1):13–32, 2000.

[16] R. L. Riolo. Survival of the fittest bits. *Sci. Am.*, 267:114–116, 1992.

[17] J. Holland. Genetic algorithms. *Sci. Am.*, 267(1):66–72, 1992.

[18] Reimer Kühn, Jort van Mourik, Martin Weigt, and Annette Zippelius. Finitely coordinated models for low-temperature phases of amorphous systems. *J. Phys. A: Math. Theor.*, 40:9227–9252, 2007.

[19] N. N. Schraudolf and T. J. Sejnowski. Unsupervised discrimination of clustered data via optimization of binary information gain. In *Advances in Neural Information Processing Systems.*, volume 5, pages 499–506, 1993.

A Details of the Theoretical Analysis

Exact time evolution

As discussed in section 3, the exact time evolution of the algorithm can be calculated when the total number of states is finite, and can be divided into four distinct phases. For the mail uptake stage, we note that agents only visit cities if their mail queue is not full, but that their behaviour at the cities is otherwise only depends on their specialisation $\vec{\theta}$. Therefore, we define the marginal densities of active agents with a given specialisation as:

$$\mu^a_{\vec{\theta}} \equiv \sum_{\vec{q}_L \ (L<L_q)} \mu_{\vec{\theta},\vec{q}_L} \tag{A.1}$$

and the total number of active agents is given by $N^a_a = N_a \sum_{\vec{\theta}} \mu^a_{\vec{\theta}}$. In the current model, agents visit cities randomly such that the probability that a subset of k agents visits any given city, is given by

$$\binom{N^a_a}{k} \left(\frac{1}{N_c}\right)^k \left(\frac{N_c - 1}{N_c}\right)^{N^a_a - k} \simeq P_{R^a_{a/c}}(k), \tag{A.2}$$

where $R^a_{a/c} \equiv N^a_a / N_c$ is the ratio of active agents to cities, and P_λ is the Poisson distribution with parameter λ, which can be truncated to arbitrary precision. We also introduce short-hand notations $\langle n \rangle$ for an arbitrary subset of mail types, $\langle n \rangle_k$ ($\equiv \{n_1..n_k\}$) for a subset with k distinct mail types, and $\langle n \rangle^l_{k-1}$ the corresponding subset with mail type n_l removed.

a.1

Now, we can write down the probability $U_m(\vec{\theta}, \vec{w}, i)$ that an active agent with specialisation $\vec{\theta}$ in position $i > 1$ at a city with initial waiting times $\vec{w}$, takes mail of type m in recursive form:

$$U_m(\vec{w},\vec{\theta},i)=\sum_{\vec{\theta}'}\mu^a_{\vec{\theta}'}\left[U_0(\vec{w},\vec{\theta}')\ U_m(\vec{w},\vec{\theta},i-1)+\sum_{n\neq m}U_n(\vec{w},\vec{\theta}')\ U_m(\vec{w}_n,\vec{\theta},i-1)\right] \quad (A.3)$$

where $\vec{w}_n \equiv \vec{w}|_{w_n\to 0}$. The corresponding probability that the agent takes no mail is given by $U_0(\vec{w},\vec{\theta},i) = 1-\sum_m U_m(\vec{w},\vec{\theta},i)$.
The $U_m(\vec{w},\vec{\theta})$ ($\equiv U_m(\vec{w},\vec{\theta},1)$) are given by:

$$\begin{aligned} U_m(\vec{w},\vec{\theta}) &= \frac{\Theta(w_m,\theta_m)}{N_m}\left(\sum_{k=0}^{N_m-1}\frac{1}{\binom{N_m-1}{k}}\sum_{\langle n\rangle_k(\not\ni m)}\prod_{i=1}^{k}(1-\Theta(w_{n_i},\theta_{n_i}))\right)\ , \\ U_0(\vec{w},\vec{\theta}) &= \prod_{m=1}^{N_m}(1-\Theta(w_m,\theta_m))\ . \end{aligned} \quad (A.4)$$

Using these definitions, we can now easily write down the total probability $U_m(\vec{\theta})$ that an active agent with specialisation $\vec{\theta}$ takes mail of type m:

$$U_m(\vec{\theta}) = \sum_{\vec{w}}\eta_{\vec{w}}\sum_{k=1}\frac{P_{R^a_{a/c}}(k-1)}{k}\sum_{i=1}^{k}U_m(\vec{w},\vec{\theta},i)\ , \quad (A.5)$$

while $U_0(\vec{\theta}) = 1-\sum_m U_m(\vec{\theta})$ is the probability that it takes no mail. Note that the inactive agents all have $U_0(\vec{\theta}) = 1$ irrespective of their specialisation. Then, the matrix $\mathbf{T}$ that describes the change in $\vec{\mu}_{\mathcal{A}}$ during the mail uptake stage is given by:

$$\begin{aligned} T_{(\vec{\theta}_n,\vec{q}_K),(\vec{\theta}_m,\vec{q}_L)} &= U_0(\vec{\theta}_n)\ \delta_{n,m}\ \delta_{\vec{q}_K,\vec{q}_L}+\delta_{K,L+1}\prod_i \delta_{q'_i,q_i}\times \\ &\quad \left(U_n(\vec{\theta}_n)\ \delta_{n,m}\ \delta_{q'_K,t_p}+U_n(\vec{\theta}_m)\ (1-\delta_{n,m})\ \delta_{q'_K,t_c}\right) \end{aligned} \quad (A.6)$$

where $\vec{\theta}_n$ is the set of thresholds ($\theta_i = \theta_{min}$ if $i=n$, and $\theta_i=\theta_{max}$ otherwise) for specialisation n.
The efficiency is given by:

$$E = \sum_{\vec{\theta}}\mu^a_{\vec{\theta}}\sum_{m=1}^{N_m}U_m(\vec{\theta}) \quad (A.7)$$

c.1

Similarly, we can write down the probability $G_{\langle n\rangle_k}(\vec{w},i)$ that exactly the subset $\langle n\rangle_k$ of mail types is given out by a city with waiting times

$\vec{w}$ and i visiting agents, in recursive form:

$$G_{\langle n \rangle_k}(\vec{w}, i) = \sum_{\vec{\theta}} \mu^a_{\vec{\theta}} \left[\sum_{l=1}^{k} U_{n_l}(\vec{w}, \vec{\theta}) \ G_{\langle n \rangle^l_{k-1}}(\vec{w}_{n_l}, i-1) + U_0(\vec{w}, \vec{\theta}) \ G_{\langle n \rangle_k}(\vec{w}, i-1) \right] \tag{A.8}$$

with $G_{\langle n \rangle_k}(\vec{w}, i) = 0$, when $k > i$ or $w_m = 0$ for any $m \in \langle n \rangle_k$, and with $G_{\langle n \rangle_0}(\vec{w}, 0) = 1$. The total probability $G_{\langle n \rangle}(\vec{w})$ that exactly a subset $\langle n \rangle$ of mail types is given out by a city with waiting times $\vec{w}$ is given by:

$$G_{\langle n \rangle}(\vec{w}) = \sum_{i=0} P_{R^a_{a/c}}(i) G_{\langle n \rangle}(\vec{w}, i) \tag{A.9}$$

Then the matrix $\mathbf{L}$ which describes the change in $\vec{\eta_{\mathcal{C}}}$ during the mail uptake stage is given by

$$L_{\eta_{\vec{w}}, \eta_{\vec{w}'}} = G_{\langle n_{\vec{w}' - \vec{w}} \rangle}(\vec{w}') \tag{A.10}$$

where $\langle n_{\vec{w}' - \vec{w}} \rangle$ is the set of indices n for which $w'_n \neq 0$ and $w_n = 0$.

a.2

The matrix $\mathbf{Q}$ that describes the change in $\vec{\mu}_{\mathcal{A}}$ during the queue processing stage is relatively straightforward to write down:

$$Q_{(\vec{\theta}, \vec{q}_L), (\vec{\theta}', \vec{q}'_{L'})} = \delta_{\vec{\theta}, \vec{\theta}'} \times \begin{cases} 1 & , \ L = 0 \\ \delta_{L,L'} \quad \delta_{q_1, q'_1 - 1} \prod_{i=2}^{L} \delta_{q_i, q'_i} & , \ L > 0, \ q'_1 > 1 \\ \delta_{L,L'-1} \quad \prod_{i=1}^{L} \delta_{q_i, q'_{i+1}} & , \ L > 0, \ q'_1 = 1 \end{cases} \tag{A.11}$$

c.2

The matrix $\mathbf{P}$ that describes the change in $\vec{\eta_{\mathcal{C}}}$ during the mail production stage is again relatively straightforward to write down:

$$P_{\vec{w}, \vec{w}'} = \prod_{m=0}^{N_m} \left[\delta_{w_m, w'_m + 1} \left(1 + (\pi_m - 1) \delta_{w'_m, 0} \right) + \delta_{w_m, w'_m} (1 - \pi_m) \delta_{w'_m, 0} \right] \tag{A.12}$$

where the π_m are time dependent for the dynamic environment only.

Efficiency Upper Bound

Now we derive the upper bound for the efficiency in *ideal circumstances*, i.e. no mail is lost due to $\boldsymbol{\ell}.1$ -$\boldsymbol{\ell}.3$, and the efficiency is only limited by $\boldsymbol{\ell}.4$. Note that all agents are active ($N^a_a = N_a$), and have an identical set of thresholds $\vec{\theta} = \{\theta_n (= \theta_{min} = 0), \forall n = 1..N_m\}$.

The city states $\mathcal{C} = \vec{w}$ can now be simplified to $\mathcal{C} = \vec{b}$ where $b_i = 1 - \delta_{w_i,0}$. The matrix $\mathbf{P}'$ that describes the changes to the $\chi_{\vec{b}}(t)$ during mail production phase, is identical to the matrix $\mathbf{P}$ in eq. A.12 with all $\vec{w}$'s replaced by $\vec{b}$'s. Similarly, the matrix $\mathbf{L}'$ that describes the changes to the $\chi_{\vec{b}}(t)$ during the mail uptake phase, can be derived like $\mathbf{L}$ in eq. A.10 with all $\vec{w}$'s replaced by $\vec{b}$'s.

Since agents act in arbitrary order, the probability $U(k,i)$ that an agent takes mail when visiting a city with k available mail types and i visiting agents in total, is given by:

$$U(k,i) = \begin{cases} \frac{k}{i} \ , & \text{if } i > k \ , \\ 1 \ , & \text{if } i \leq k \ . \end{cases} \tag{A.13}$$

Therefore, the total probability that an agent takes mail (i.e. the efficiency) can be expressed as:

$$\begin{aligned} E(t) &= \sum_{k=1}^{N_m} \chi_k(t) \ \sum_j P_{R_{a/c}}(j-1) \ U(k,j) \ , \qquad \text{(A.14)} \\ &= \sum_{k=1}^{N_m} \chi_k(t) \ \left(1 - P_{R_{a/c}}(k) + \frac{k - R_{a/c}}{R_{a/c}} (1 - \sum_{j=0}^{k} P_{R_{a/c}}(j)) \right) \ . \end{aligned}$$