Volume 29 Number 2 June 2005 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Ant Colonies and Multi-Agent Systems

Guest Editors:

Nadia Nedjah

Luiza de Macedo Mourelle

1977

The Slovene Society Informatika, Ljubljana, Slovenia

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Introduction

Instead of designing complex and centralized systems, researchers rather prefer to work with many small and autonomous agents. The agents mimic the ant's behavior within an ant colony. Each one acting on the simplest of rules, these many agents can solve very complex problems known as hard problems. Generally, such multi-agent systems are used as search and optimization tools.

This special issue of the *Informatica - International Journal of Computing and Informatics* is focused on ant colonies and multi-agent systems. It includes seven contributions that describe new methods and experiences for multi-agent implementations of aspects of artificial life, ant colony and swarm intelligence.

The first paper is entitled "Investigating Strategic Inertia Using OrgSwarm" and was proposed by *Anthony Brabazon, Arlindo Silva, Tiago Ferra de Sousa, Michael O'Neill, Robin Matthews and Ernesto Costa*. The study describes a novel simulation model, called *OrgSwarm*, of the process of strategic adaptation. In this paper, strategic adaptation is conceptualized as a process of adaptation or search, on a landscape of strategic possibilities, by a population of profit-seeking organizations.

The second paper is entitled "Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm" and was proposed by *André L. Vizine, Leandro N. de Castro, Eduardo R. Hruschka, Ricardo R. Gudwin*. The paper introduces and discusses both a progressive vision scheme and pheromone heuristics for the standard ant-clustering algorithm, together with a cooling schedule that improves its convergence properties. The proposed algorithm is evaluated in a number of well-known benchmark data sets, as well as in a real-world bioinformatics dataset.

The third paper is entitled "Efficient Pre-Processing for Large Window-Based Modular Exponentiation Using Ant Colony" and was proposed by *Nadia Nedjah and Luiza de Macedo Mourelle*. The paper exploits the ant colony strategy to finding an optimal addition sequence that allows one to perform the pre-computations in window-based methods with a minimal number of modular multiplications and hence, improves the efficiency of modular exponentiation.

The forth paper is entitled "Max Min Ant System and Capacitated *p*-Medians: Extensions and Improved Solutions" and was proposed by *Fabrício Olivetti de França, Fernando J. Von Zuben, Leandro Nunes de Castro*. The work introduces a modified MAX MIN Ant System (MMAS) designed to solve the Capacitated *p*-Medians Problem (CPMP). It presents the most relevant steps towards the implementation of an MMAS to solve the CPMP, including some improvements on the original MMAS algorithm, such as the use of a density model in the information heuristics and a local search adapted from the un-capacitated *p*-medians problem.

The fifth paper is entitled "Application of Ant-based Template Matching for Web Documents Categorization" and was proposed by *Siok Lan Ong, Weng Kin Lai, Tracy S. Y. Tai, Choo Hau Ooi and Kok Meng Hoe*. The paper examines the direct implementation of a template based on a Gaussian Probability Surface to supervise these homogeneous multi-agents to form clusters within a specified dropping zone.

The sixth paper is entitled "Efficient and Scalable Communication in Autonomous Networking using Bio-inspired Mechanisms – An Overview" and was proposed by *Falko Dressler*. In this paper, the author demonstrates the possibilities which evolve by the application of cell biology for computer networking. With the focus on autonomous networking, the combination with methodologies known from swarm intelligence is evaluated. The author shows the capabilities of this combination and derive destinations and goals for self-organization in communication networks showing a more efficient and scalable behavior.

The seventh paper is entitled "Model Checking Multi-Agent Systems" and was proposed by *Mustapha Bourahla and Mohamed Benmohamed*. In this paper, the authors show how a well known and effective verification technique, model checking, can be generalized to deal with multi-agent systems. The paper explores a particular type of multi-agent system, in which each agent is viewed as having the three mental attitudes of belief, desire and intention.

Nadia Nedjah and Luiza de Macedo Mourelle

Investigating Strategic Inertia Using OrgSwarm

Anthony Brabazon, Arlindo Silva, Tiago Ferra de Sousa, Michael O'Neill, Robin Matthews, Ernesto Costa
University College Dublin, Ireland.
anthony.brabazon@ucd.ie

Escola Superior de Tecnologia, Instituto Politecnico de Castelo Branco,Portugal.
arlindo@est.ipcb.pt

University of Limerick, Ireland.
michael.oneill@ul.ie

Kingston University, London.
r.matthews@kingston.ac.uk

Centro de Informatica e Sistemas da Universidade de Coimbra, Portugal.
ernesto@dei.uc.pt

Keywords: Particle swarm, Organizational adaptation, OrgSwarm

Received: July 26, 2004

*This study describes a novel simulation model (**OrgSwarm**) of the process of strategic adaptation. Strategic adaptation is conceptualized as a process of adaptation (search), on a landscape of strategic possibilities, by a population of profit-seeking organizations. Unfortunately, the characteristics that make organizations coherent and viable such as organizational structure and shared organizational culture, also create strategic inertia, potentially limiting the ability of organizations to adapt. This study examines the impact of strategic inertia on the adaptive potential of organizations. The simulation results suggest that a degree of strategic inertia can assist rather than hamper adaptive efforts in static and slowly changing strategic environments.*

Povzetek: Predstavljen je OrgSwarm, nov model procesa strateškega prilagajanja.

1 Introduction

There are parallels between biological and social systems. In both, individuals within a larger population are attempting to appropriate scarce resources, or *to earn a living*, in a dynamic environment. Entities in these systems typically alter their 'strategies' over time in an attempt to improve their success. In an organizational setting, a strategy consists of a choice of what activities the organization will perform, and choices as to how these activities will be performed [36]. These choices define the strategic configuration of the organization. Recent work by [28] and [38] has recognized that strategic configurations consist of interlinked individual elements (decisions), and have applied general models of interconnected systems such as Kauffman's NK model to examine the implications of this for processes of organizational adaptation.

Following a long-established metaphor of adaptation as search [46], strategic adaptation is considered in this study as an attempt to uncover peaks on a high-dimensional strategic landscape. Some strategic configurations produce high profits, others produce poor results. The search for good strategic configurations is difficult due to the vast (combinatorial) number of configurations, uncertainty as to the nature of topology of the strategic landscape faced by an organization, and changes in the topology of this landscape over time. Despite these uncertainties, the search process for good strategies is not blind. Decision-makers receive feedback on the success of their current and historic strategies, and can assess the payoffs received by the strategies of their competitors [26]. Hence, certain areas of the strategic landscape are illuminated.

Organizations do not exist in isolation, but interact with, and receive feedback from, their environment. Their efforts at strategic adaption are guided by social as well as individual learning. Good ideas discovered by one organization disseminate over time. One model combining both individual and social learning which has attracted significant interest in recent years is that of *Particle Swarm Optimization* (PSO) [21], [25]. Particle swarm research has been concentrated in two broad areas, the application and study of PSO as an optimizing tool, and the application of PSO as a model of social and cultural adaptation. This paper adopts the second of these perspectives, and adapts the canonical PSO to create a plausible model of the process of strategic adaptation.

Although the particle swarm model has been applied to a variety of problems in the fields of engineering [1], chemistry [34], medicine and psychology [25], as yet it has not been applied to the domain of organizational science. This paper introduces the model to this domain, and utilizes it to examine the impact of differing degrees of strategic inertia on the adaptive capabilities of a population of organizations.

1.1 Structure of paper

This contribution is organized as follows. Section 2 provides a short discussion of prior literature in the domain of strategic adaptation in order to provide a number of perspectives on strategic inertia. Section 3 incorporates an introduction to the canonical Particle Swarm algorithm (PSA),[1] followed by a description of the simulation model in Section 4. Section 5 outlines the results of the simulations and finally, conclusions and future work are discussed in Section 6.

2 Strategic Adaptation

Strategic adaptation and strategic inertia are closely linked. If strategic adaptation is problematic, inertia is a possible cause. A substantial literature has emerged on strategic adaptation. This, along with its implications for strategic inertia, is outlined below.

Two polar views exist concerning the ability of organizations to adapt their strategic configuration. Adaptationists or advocates of strategic choice [35], [40], [31], broadly consider that managers or dominant coalitions in organizations scan the environment for current and future opportunities and threats, formulate strategic responses and adjust organizational activities and structure appropriately [10]. Therefore, strategic direction and organizational form are determined by managers, and market selection processes act to maintain organizations which are good adaptors. Under this perspective, an organization's fate is largely in its own hands, and hence strategic inertia is considered to represent a challenge rather than a roadblock to strategic adaptation efforts. The adaptationist argument presupposes that organizations are capable of adapting at least as fast as their environment changes [31], [30]. If firms are incapable of responding to environmental changes in a similar timescale, adaptation (or learning) processes will not enhance organizational survival [13]. The current practitioner interest in 'change management' [16] exemplifies the belief that even substantial strategic adaptation is possible.

In contrast, the population ecology school [12] proposes an alternative view on organizational-environment relations. This school of thought allows that organizations have some ability to adapt to environmental change and notes that '*leaders of organizations do formulate strategies and organizations do adapt to environmental contingencies*' [12] (p. 930). However, it is argued that the ability of firms to accurately and consistently adapt in a world of high uncertainty, where connections between means and ends are unclear is doubtful [13], [9]. Although selection processes select the most fit organizations in a given environment for continued survival, population ecologists contend that an organization's fitness primarily arises because of good initial strategic choices, or luck, rather than reflecting post-founding adaptation [2]. Advocates of the population ecology school suggest that the ability of organizations to adapt is highly constrained because of their inherent inertia. This inertia stems from two sources, *imprinting forces*, and as a *consequence of market selection forces*.

2.1 Imprinting Forces

Imprinting forces [4] combine to define and solidify the strategic configuration of a newly formed organization. These forces include the dominant initial strategy pursued by the organization, the skills / prior experience of the management team, and the distribution of decision-making influence in the organization at time of founding [4]. These forces influence the initial choice of organizational strategy. As consensus concerning the strategy emerges, it is imprinted on the organization through resource allocation decisions [42]. The imprinting leads to inertia by creating sunk costs, internal political constraints, and a rigid organizational structure. Over time this inertia intensifies due to the formation of an organizational history which creates barriers to industry exit, and legitimacy issues if adaptation is suggested [12]. The resulting inertia serves to circumscribe the organization's ability to adapt its strategy in the future. The initial imprinting determines the basin of attraction in which the organization is located on the strategic landscape. Imprinting also occurs as relationships are built up with suppliers and customers [43]. The creation of a web of these relationships can serve to constrain the range of strategic alternatives in the future, as strategic moves which dramatically disrupt the web are less likely to be considered.

2.2 Market-Selection Forces

The discussion of strategic inertia was extended by [13] who posited that inertia is also created as a natural *consequence* of the market-selection process, claiming that '*selection processes tend to favor organizations whose structures are difficult to change.*' (p. 149). The basis of this claim is that organizations which can produce a good or service reliably (consistently of a minimum quality standard) are favored for trading purposes by other organizations, and therefore by market selection processes. The routines required to produce a product or service reliably, tend to lead to structural inertia, as the construction of routines to achieve this leads to an increase in the complexity of the patterns of links between organizational subunits [13] & [27]. Building on this point, it can be posited that more efficient organizations are likely to exhibit inertia. As organizations seek better environment-structure congruence,

[1] The term PSA is used in place of PSO (Particle Swarm Optimization) in the remainder of this paper, as the object of the paper is not to develop a tool for optimizing, but to apply the swarm metaphor as a model of organizational adaptation.

their systems become increasingly specialized and interlinked, making changes to their activities become costly and difficult. Structural inertia is rooted in the size, complexity and interdependence of the firm's structures, systems, procedures and processes [45]. Theoretical support for these assertions, that increasing organizational complexity can make adaptation difficult, is found in [19] and [38], as the heightened degree of interconnections between activities within the firm will increase the 'ruggedness' of the strategic landscape faced by an organization.

The arguments that organizations are subject to strategic inertia also finds resonance in the literature concerning organizational learning and organizational memory. The preference of organizations to continue to pursue activities similar to those undertaken in the past has been widely noted [14], [32], as has the cumulative nature of organizational learning [33].

In summary, strategic inertia can arise from a variety of sources, and the general consensus in organizational literature is that its existence poses clear difficulties for strategic adaptation by organizations.

3 Particle Swarm Algorithm

This section provides an introduction to the Particle Swarm algorithm (PSA). A full description of this algorithm and the cultural model which inspired it is provided in [25]. A Swarm can be defined as '... *a population of interacting elements that is able to optimize some global objective through collaborative search of a space.*' [25](p. xxvii). The nature of the interacting elements (particles) depends on the problem domain, in this study they represent organizations. These particles move (fly) in an n-dimensional search space, in an attempt to uncover ever-better solutions to the problem of interest.

Each of the particles has two associated properties, a current position and a velocity. Each particle also has a memory of the best location in the search space that it has found so far (*pbest*), and knows the location of the best location found to date by all the particles in the population (*gbest*). At each step of the algorithm, particles are displaced from their current position by applying a velocity vector to them. The size and direction of this velocity is influenced by the velocity in the previous iteration of the algorithm (simulates momentum), and the current location of a particle relative to its pbest and gbest. Therefore, at each step, the size and direction of each particle's move is a function of its own history (experience), and the social influence of its peer group. A number of variants of the PSA exist. The following paragraphs provide a description of the basic *continuous* version described by [25]. The algorithm is initially described narratively. This is followed by a description of the particle position-update equations.

3.1 The Algorithm

i. Initialize each particle in the population by randomly selecting values for its location and velocity vectors

ii. Calculate the fitness value of each particle. If the current fitness value for a particle is greater than the best fitness value found for the particle so far, then revise *pbest*

iii. Determine the location of the particle with the highest fitness and revise *gbest* if necessary

iv. For each particle, calculate its velocity according to equation (1)

v. Update the location of each particle

vi. Repeat steps ii - v until stopping criteria are met

Each particle i has an associated current position in the search space x_i, a current velocity v_i, and a personal best position in the search space y_i. During each iteration of the algorithm, the location and velocity of each particle is updated using equations (1) - (5).

To provide intuition on the workings of the algorithm, see figure 1. Each particle i has an associated current position in search space $x(t) = (x_{i1}(t), \ldots, x_{in}(t))$ at time t, a current velocity of $v(t) = (v_{i1}(t), \ldots, v_{in}(t))$, and a pbest position of $y_i(t) = (y_{i1}(t), \ldots, y_{in}(t))$. The position of the particle at time $t+1$ is a determined by $x(t) + v(t+1)$, and $v(t+1)$ is obtained by a stochastic blending of $v(t)$, an acceleration towards gbest (v_{gbest}) and an acceleration towards pbest (v_{pbest}).

Assuming a function f is to be maximized, that the swarm consists of m particles, and that r_1, r_2 are drawn from a uniform distribution in the range (0,1), the velocity update for particle i is as follows:

$$v_i(t+1) = W v_i(t) + c_1 r_1 (y_i - x_i(t)) + c_2 r_2 (\hat{y} - x_i(t)) \tag{1}$$

where ŷ is the location of the global-best solution found by all the particles.[2] In every iteration of the algorithm, each particle's velocity is stochastically accelerated towards its previous best position and towards a neighborhood (global) best position. The weight-coefficients c_1 and c_2 control the relative impact of pbest and gbest locations on the velocity of a particle. The parameters r_1 and r_2 ensure that the algorithm is stochastic. A practical effect of the random coefficients r_1 and r_2, is that neither the individual nor the social learning terms are always dominant. Sometimes one or the other will dominate [25]. Although the velocity update has a stochastic component, the search process is not random. It is guided by the memory of past 'good' solutions corresponding to a psychological tendency for individuals to repeat strategies which have worked for them in the past

[2] A variant on the basic algorithm is to use a local rather than a global version of gbest. In the local version, gbest is set independently for each particle, based on the best point found thus far within a *neighborhood* of that particle's current location.

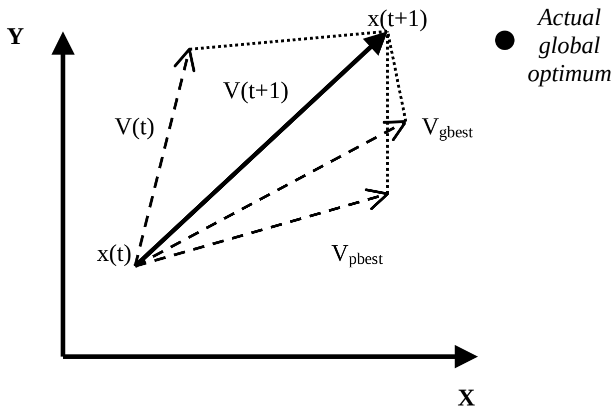

Figure 1: A representation of the particle position-update process.

[22], and by the global best solution found by all particles thus far. W represents a momentum coefficient which controls the impact of a particle's prior-period velocity on its current velocity. Each component of a velocity vector v_i is restricted to a range $[-v_{max}, v_{max}]$ to ensure that individual particles do not leave the search space. The implementation of a v_{max} parameter can also be interpreted as simulating the incremental nature of most learning processes [22]. The value of v_{max} is usually chosen to be $k * x_{max}$, where $0 < k < 1$. Once the velocity update for particle i is determined, its position is updated and pbest is updated if necessary.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

$$y_i(t+1) = y_i(t) \text{ if, } f(x_i(t)) \le f(y_i(t)), \tag{3}$$

$$y_i(t+1) = x_i(t) \text{ if, } f(x_i(t)) > f(y_i(t)) \tag{4}$$

After all m particles have been updated, a check is made to determine whether gbest needs to be updated.

$$\hat{y} \in (y_0, y_1, \ldots, y_m) | f(\hat{y}) = \max\ (f(y_0), f(y_1), \ldots, f(y_m)) \tag{5}$$

3.1.1 PSA vs the Genetic Algorithm

It is noted that the PSA bears similarity to other biologically-inspired optimizing algorithms. Like the Genetic Algorithm (GA), it is a population-based algorithm, is typically initialized with a population (swarm) of random solutions, and search proceeds by updating these solution each generation (iteration). Unlike the GA, the move (update) operators are not direct analogs of the genetic operators of mutation and crossover,[3] there is no explicit selection process, and potential solutions are referred to as particles rather than as chromosomes.

The communication (information-sharing) mechanism of the PSA also differs from that of the GA. In the GA, the communication is between two solutions, in the PSA, the communication is between the current solution, its pbest and the gbest. Hence, candidate solutions can 'see' the global best solution found by all particles thus far. The movement of each particle through the search space is influenced by their own previous experience (history) and a wish to move towards the global, best position found thus far by other particles [39].

3.2 The PSA and Social Learning

Despite its simplicity, the PSA is capable of capturing a surprising level of complexity, as individual particles are capable of both individual and social learning. In social settings, individuals are not '*...isolated information-processing entities ...*' [25] (p. xv), but also learn from the experiences of their peers. Social behavior helps individuals to adapt to their environment, as it ensures that they obtain access to more information than that captured by their own senses. Learning in social species is therefore distributed and parallel.

Communication (interactions) between agents (individuals) in a social system may be direct or indirect. An example of the former could arise when two organizations trade with one another. Examples of the latter include:

i. the observation of the success (or otherwise) of a strategy being pursued by another organization, and

ii. *stigmergy* which arises when an organization modifies the environment, which in turn causes an alteration of the actions of another organization at a later time.

The first of these indirect learning mechanisms is included in the canonical PSA, the second can be included through an adaption of the basic model.

[3]It can be argued that the velocity vector update does bear similarity to a recombination operator, being impacted by the location of pbest and gbest [21].

The mechanisms of the basic Particle Swarm model bear *prima facie* similarities to those of the domain of interest, organizational adaptation. It embeds concepts of a population of entities which are capable of individual and social learning. However, the model requires modification before it can employed as a plausible model of organizational adaptation. These modifications, along with a definition of the strategic landscape used in this study are discussed in the next section.

4 OrgSwarm Model

This section describes the simulation model (OrgSwarm) employed in this study [7], [8]. The model can be classed as a multi-agent system (MAS). MASs focus attention on collective intelligence, and the emergence of behaviors through the interactions between the agents. MAS usually contain a world (environment), agents, relations between the entities, a set of activities that the agents can perform, and changes to the environment as a result of these activities [44]. The key components of the simulation model, the landscape generator (environment), and the adaption of the basic Particle Swarm algorithm to incorporate the activities and interactions of the agents (organizations) are described next.

4.1 Strategic Landscape

In this study, the strategic landscape is defined using the NK model [18], [19]. Application of the NK model to define a strategic landscape is not atypical and has support from existing literature in organizational science [28],[38], [15], and related work on technological innovation [29], [20], [41], [37]. The NK model considers the behavior of systems which are comprised of a configuration (string) of N individual elements. Each of these elements are in turn interconnected to K other of the N elements (K<N). In a general description of such systems, each of the N elements can assume a finite number of states. If the number of states for each element is constant (S), the space of all possible configurations has N dimensions, and contains a total of $\prod_{i=1}^{N} S_i$ possible configurations.

In Kauffman's operationalization of this general framework [19], the number of states for each element is restricted to two (0 or 1). Therefore the configuration of N elements can be represented as a binary string. The parameter K, determines the degree of fitness interconnectedness of each of the N elements and can vary in value from 0 to N-1. In one limiting case where K=0, the contribution of each of the N elements to the overall fitness value (or worth) of the configuration are independent of each other. As K increases, this mapping becomes more complex, until at the upper limit when K=N-1, the fitness contribution of any of the N elements depends both on its own state, and the simultaneous states of all the other N-1 elements, describing a fully-connected graph.

If we let s_i represent the state of an individual element i, the contribution of this element (f_i) to the overall fitness (F) of the entire configuration is given by $f_i(s_i)$ when K=0. When K>0, the contribution of an individual element to overall fitness, depends both on its state, and the states of K other elements to which it is linked ($f_i(s_i : s_{i1}, ..., s_{ik})$). A random fitness function (U(0,1)) is adopted, and the overall fitness of each configuration is calculated as the average of the fitness values of each of its individual elements. Therefore, if the fitness values of the individual elements are $f_1, ..., f_N$, overall fitness (F) is:

$$F=\frac{\sum_{i=1}^{N} f_i}{N} \tag{6}$$

Altering the value of K effects the ruggedness of the described landscape, and consequently impacts on the difficulty of search on this landscape [18], [19]. The strength of the NK model in the context of this study is that by tuning the value of K it can be used to generate strategic landscapes (graphs) of differing degrees of local-fitness correlation (ruggedness). The strategy of an organization is characterized as consisting of N attributes [28]. Each of these attributes represents a strategic decision or policy choice, that an organization faces. Hence, a specific strategic configuration s, is represented as a vector $s_1, ..., s_N$ where each attribute can assume a value of 0 or 1 [38]. The vector of attributes represents an entire organizational form, hence it embeds a choice of markets, products, method of competing in a chosen market, and method of internally structuring the organization [38]. Good consistent sets of strategic decisions (configurations), correspond to peaks on the strategic landscape.

The definition of an organization as a vector of strategic attributes finds resonance in the work of Porter [35], [36], where organizations are conceptualized as a series of activities forming a value-chain.[4] The choice of what activities to perform, and subsequent decisions as to how to perform these activities, defines the strategy of the organization. The individual attributes of an organization's strategy interact. For example, the value of an efficient manufacturing process is enhanced when combined with a high-quality sales force. Differing values for K correspond to varying degrees of payoff-interaction among elements of the organization's strategy [38]. As K increases, the difficulty of the task facing strategic decision makers is magnified. Local-search attempts to improve an organization's position on the strategic landscape become ensnared in a web of conflicting constraints.

It is acknowledged that there are limitations to using the NK model as a strategic landscape generator. The model produces a finite graph and presupposes the existence of a strategy space, albeit one which may be poorly understood by strategists. This implies that it is inappropriate to apply the NK model to examine very long run adaptive processes, where organizational fitness is not clearly bounded, and

[4] This activity-based conceptualization has spread beyond studies of strategy to encompass new methods of costing products/services [17].

where the dimensionality of the strategy space itself could change. It is also noted that the NK model assumes a constant value of K for all elements. In reality, the value of K is likely to differ for varying elements of a strategy vector. In the work of [37], a distinction is drawn between *generic activities* which are likely to have an optimal configuration for many firms, for example, the possession of an accounting system. Generic activities (or 'table-stakes'), whilst important for the successful operation of the firm, are usually not strongly interconnected with the non-generic activities of the firm [37]. In contrast, the firm-specific element of strategy are typically highly interconnected, as they embed choices involving trade-offs between alternative strategic configurations [36], [37]. Hence, the NK landscape can be considered to represent these non-generic, interconnected, elements of the strategy vector, rendering the assumption of a constant value of K more plausible.

4.2 The Algorithm

Five characteristics of the problem domain which impact on the design of a simulation model are:

i. the environment is dynamic,

ii. organizations are prone to strategic inertia,

iii. organizations do not knowingly select poorer strategies than the one they already have (election operator),

iv. organizations make errorful *ex-ante* assessments of fitness, and

v. organizations co-evolve.

Each of these factors is embedded in our simulation model. In this study we report results which consider the first three of these factors. Future work will extend this to incorporate the latter two. We note that this model bears passing resemblance to the eleMentals model of [24], which combined a swarm algorithm and an NK landscape, to investigate the development of culture and intelligence in a population of hypothetical beings called eleMentals. However, the strategic model developed in this study is differentiated from the eleMental model, not just on grounds of application domain, but because of the inclusion of an inertia operator, and also by the investigation of both static and dynamic environments.

4.2.1 Dynamic environment

Organizations do not compete in a static environment. Rather they can individually and collectively alter their environment. The environment may also be altered as a result of exogenous events. The second of these factors is implemented in this study by allowing the landscape itself to be respecified. During the course of a simulation run, the strategic landscape can be stochastically subject to minor or major respecification, mimicking a *regime change*, such as the emergence of a new technology, or a change in customer preferences. These respecifications simulate a dynamic environment, and a change in the environment may at least partially negate the value of past learning (adaptation) by organizations.[5] Minor respecifications are simulated by altering the fitness values associated with one of the N dimensions in the NK model, whereas in major changes, the fitness of the entire NK landscape is redefined. The probability that a minor or major respecification occurs is controlled by the modeler.

4.2.2 Inertia

Organizations do not have complete freedom to alter their current strategy. Their adaptive processes are subject to conservatism arising from inertia. Inertia springs from the organization's culture, history, and the mental models of its management [4]. In the simulation strategic inertia is mimicked by implementing a 'strategic anchor'. The degree of inertia can be varied in the simulations from zero to high. In the latter case, the organization is highly constrained from altering its strategic stance. By allowing the weight of this anchor to vary, adaptation processes corresponding to different industries, each with different levels of inertia, can be simulated. Inertia could be incorporated into the PSA in a variety of ways. We have chosen to incorporate it into the velocity update equation, so that the velocity and direction of the particle at each iteration is also a function of the location of its 'strategic anchor'. Therefore for the simulations, equation 1 is altered by adding an additional inertia term

$$v_i(t+1)=v_i(t)+R_1(y_i-x_i(t))+R_2(\hat{y}-x_i(t)+R_3(a_i-x_i(t)) \qquad (7)$$

where a_i represents the value of the anchor on dimension i (a full description of the other terms such as R_1 is provided in the pseudo-code below). This anchor can be fixed at the initial position of the particle at the start of the algorithm, or it can be allowed to 'drag', thereby being responsive to the recent adaptive history of the particle. Both the weight attached to the anchor parameter (relative to those attached to pbest and gbest), and in the case of a dragging anchor, the number of periods over which the anchor can drag, can be altered by the modeler.

It is noted that the concept of inertia developed in this paper is not limited to organizations, but is plausibly a general feature of social systems. Hence, the extension of the social swarm model to incorporate inertia may prove useful beyond this study.

4.2.3 Election operator

Real-world organizations do not usually intentionally move to poorer strategies. Hence, an election operator is im-

[5] As noted by [11] (p. xxvii), '*the very processes and values that constitute an organization's capabilities in one context, define its disabilities in another.*'.

plemented, whereby position updates which would worsen an organization's strategic fitness are discarded. In these cases, an organization remains at its current location. One economic interpretation of the election operator, is that strategists carry out a mental simulation or *thought experiment*. If the expected fitness of the proposed strategy appears unattractive, the 'bad idea' is discarded [6], [25]. The simulation therefore incorporates a 'rachet' operator option, which if turned on, ensures that an organization only updates (alters) its strategy if the new strategy being considered is better than its current strategy. By permitting strategists to conduct thought experiment during each iteration of the algorithm, strategists are given a *look-ahead* capability. They can direct their adaptive efforts to the area of the strategic landscape which offer potential.

4.2.4 Outline of algorithm

A number of further modifications to the basic PSA are required. As the strategic landscape is defined using a binary representation, the canonical PSA described above is adapted for the binary case using the *BinPSO* version of the algorithm [23]. The binary version of the PSA is inspired by the idea that an agent's probability of making a binary decision (yes/no, true/false) is a function of both personal and social factors **Eq.** 8.

$$P(\mathbf{x_i(t+1)=1})=f(\mathbf{x_i(t)},\mathbf{v_i(t)},\mathbf{pbest},\mathbf{gbest},\mathbf{anchor}) \quad (8)$$

The vector $\mathbf{v_i}$ is now interpreted as organization i's predisposition to set each of the n binary strategic choices that they face to one. The higher the value of v_i^j for an individual decision j, the more likely that organization i will choose to set decision $j = 1$, with lower values of v_i^j favoring the choice of decision $j = 0$.

In order to model the tendency of organizations to repeat historically good strategies, values for each dimension of $\mathbf{x_i}$, which match those of **pbest**, should become more probable in the future, and the $Prob(x_i^j = 1)$ should be adjusted towards $pbest_i^j$ on each dimension j. Adding the difference between $pbest_i^j$ and x_i^j for organization i to v_i^j will move the probability thresholds towards 1.0, if the distance is positive ($pbest_i^j = 1$ and $x_i^j = 0$). If the difference between $pbest_i^j$ and x_i^j for organization i is negative ($pbest_i i^j = 0$), and $x_i^j = 1$, adding the difference to v_i^j will move it towards 0.0. The difference in each case is weighted by a random number drawn from U(0,1).[6]

In order to ensure that the vector $\mathbf{v_i(t+1)}$ is mapped into (0,1), a sigmoid transformation is performed on each element j of $\mathbf{v_i(t+1)}$ (**Eq.** 9), and each element of $Sig(\mathbf{v_i(t)})$ is mapped to either 0 or 1 by comparing it with a vector of random numbers $\mathbf{prob_i(t+1)}$ drawn from $U(0,1)$ (**Eq.** 10) .

[6]Similarly, each organization has a tendency to match the values for each dimension of $\mathbf{x_i}$ to those of **gbest**, and its anchor. Therefore, the resulting value of $v_i^j(t+1)$, is influenced by $v_i^j(t)$, and the position of gbest, pbest, and anchor.

$$Sig(v_i^j(t+1))=\frac{1}{1+exp(-v_i^j(t+1))} \quad (9)$$

$$prob_i^j(t+1)<Sig(v_i^j(t+1))\ then\ x_i^j(t+1)=1;\ else\ x_i^j(t+1)=0 \quad (10)$$

The pseudo-code for the algorithm is as follows:

```
For each dimension n
  v[n]=v[n]+R1*(g[n]-x[n])+R2*(p[n]-x[n])+R3*(a[n]-x[n])
  If(v[n]>Max) v[n]=Vmax
  If(v[n]<-Vmax) v[n]=-Vmax
  If(Pr<S(v[n]))t[n]=1
    Else t[n]=0
UpdateAnchor(a)       //if iteratively update anchor
                      //option is selected
```

$R1$, $R2$ and $R3$ are random weights drawn from a uniform distribution ranging from 0 to $R1_{max}$, $R2_{max}$ and $R3_{max}$ respectively, and they weight the importance attached to the gbest, pbest and anchor in each iteration of the algorithm. $R1_{max}$, $R2_{max}$ and $R3_{max}$ are constrained to sum up to 4.0. x is the particle's actual position, g is the global best position, p each particle's personal best position and a is the position of the particle's anchor. V_{max} is set to 4.0. Pr is a probability value drawn from a uniform distribution ranging from 0 to 1, and S is the sigmoid function: $S(x) = \frac{1}{1+exp(-x)}$, which squashes v into a 0 to 1 range. t is a temporary record which is used in order to implement conditional moving. If the new strategy is accepted, t is copied into x, otherwise t is discarded and x remains unchanged.

5 Results

This section provides the results from our simulation study. As the adaptive process is stochastic, and as the initialization of the position and velocity for each organization is random, each simulation run describes a single sample-path through time. There are many possible sample-paths, so the results of the simulations are averaged over multiple (30) runs in an attempt to uncover prevalent characteristics of the sample paths which the system can give rise to. All simulations were run for 5,000 iterations, and all reported fitnesses are the average population fitnesses, and average environment best fitnesses, across 30 separate simulation runs. On each of the simulation runs, the NK landscape is specified anew, and the positions and velocities of particles are randomly initialized at the start of each run. A population of 20 particles is employed, with a neighborhood of size 18. The choice of a high value for the neighborhood, relative to the size of the population, arises from the observation that real-world organizations know the profitability of their competitors.

Tables (1, 2 and 3) provide the results for each of fourteen distinct PSA variants, at the end of 5,000 iterations, across a number of static and dynamic NK landscape scenarios. In each scenario, the same series of simulations are undertaken. Initially, a basic PSA is employed, without an anchor or a rachet (conditional move) operator. This simulates a population of organizations searching a strategic landscape, where the population has no strategic inertia, and where organizations do not utilize a rachet operator in deciding whether to alter their position on the strategic landscape.

The basic PSA is then supplemented by a series of strategic anchor formulations, ranging from an anchor which does not change position during the simulation (initial anchor) to one which can adapt after a time-lag (moving anchor). Two lag periods are examined, a 20 and a 50 iteration lag. Differing weights can be attached to the inertia term in the velocity equation, ranging from 0 (inertia is turned off) to a maximum of 4. To determine whether the weight factor has a critical impact on the results, results are reported for weight values of both 1 and 3. Next, to isolate the effect of the rachet, the conditional move operator is implemented, and inertia is turned off. Finally, to ascertain the dual effect of both rachet and inertia when they are combined, the inertia simulations outlined above are repeated with the rachet operator turned on.

Real world strategy vectors consist of a large array of strategic decisions. A value of N=96 was chosen in defining the landscapes in this simulation. It is noted that there is no unique value of N that could have been selected, but the selection of very large values are not feasible due to computational limitations. However, a binary string of 96 bits provides 2^{96}, or approximately 10^{28}, distinct choices of strategy. It is also noted that we would expect the dimensionality of the strategy vector to exceed the number of organizations in the population, hence the size of the population is kept below 96, and a value of 20 is chosen. A series of landscapes of differing K values (0,4 and 10), representing differing degrees of fitness inter-connectivity, were used in the simulations.

5.1 Static Landscape

Table 1 and figures 2 and 3, provide the results for a static NK landscape.[7] Examining these results suggests that the basic PSA, without inertia or rachet operators, performs poorly, even on a static landscape. The average of the average batch populational fitnesses obtained after 5,000 iterations is not better than random search (the expected value of a random point on the landscape is 0.50), suggesting that unfettered adaptive efforts, based on communication between organizations (gbest), and a memory of good past strategies (pbest) is not sufficient to achieve high levels of populational fitness. When a series of anchor mechanisms simulating strategic inertia are added to the basic PSA, the results are not qualitatively altered from those of the basic PSA. This suggests that communication and inertia alone, are not sufficient for the attainment of high levels of populational strategic fitness.

When a rachet operator is added to the basic PSA, a significant improvement in both average populational, and average environment best fitness is obtained across landscapes of all K values, suggesting that the simple decision heuristic of *only abandon a current strategy for a better one* can lead to notable increases in populational fitness. Finally, the results of a series of combination anchor and rachet mechanisms are reported. Virtually all of these combinations lead to significantly (at the 5% level) enhanced levels of populational fitness (against the rachet-only PSA), suggesting that inertia can be beneficial, when combined with a rachet mechanism. Examining the combined rachet and anchor results in more detail, the best results are obtained when the anchor is not fixed at the initial location of each particle on the landscape, but when it is allowed to 'drag' or adapt, over time. It is also noted that the results are not qualitatively sensitive to the weight value (1 or 3).

5.2 Dynamic Landscape

The real world is rarely static, and changes in the environment can trigger adaptive behavior by agents in a system [3]. In this simulation, the landscape can change at a variety of time scales, and the size of the relocation 'jump' of the optimum position on the landscape can vary. Therefore, the environment can be changed with varying temporal, and spatial severity [3]. Two specific scenarios are examined. Table 2 and figures 4 and 5, provides the results for the case where a single dimension of the NK landscape is respecified in each iteration of the algorithm with a probability of P=0.00025. Table 3 and figures 6 and 7, provides the results for the case where the entire NK landscape is respecified with the same probability. When the landscape is wholly or partially respecified, the benefits of past strategic learning by organizations is eroded.

Qualitatively, the results in both scenarios are similar to those obtained on the static landscape. The basic PSA, even if supplemented by an anchor mechanism, does not perform any better than random search. Supplementing the basic PSA with the rachet mechanism leads to a significant improvement in populational fitness, with a further improvement in fitness occurring when the rachet is combined with an anchor. In the latter case, an adaptive or dragging anchor gives better results than a fixed anchor, but the results between differing forms of dragging anchor do not show a clear dominance for any particular form. As for the static landscape case, the results for the combined rachet / anchor, are relatively insensitive to the weight value (1 or 3).

6 Conclusions

The objective of this study has been to examine the impact of strategic inertia on the dynamic adaptation of a population of organizations. A novel synthesis of a strategic landscape defined using the NK model, and a Particle Swarm metaphor to model the adaption of organizations

[7]These simulations were also undertaken with a neighborhood size of four, to determine whether the results were sensitive to neighborhood size. No significant differences in the results between the two neighborhood sizes was noted. As a result, the remaining simulations were run with a neighborhood of size 18.

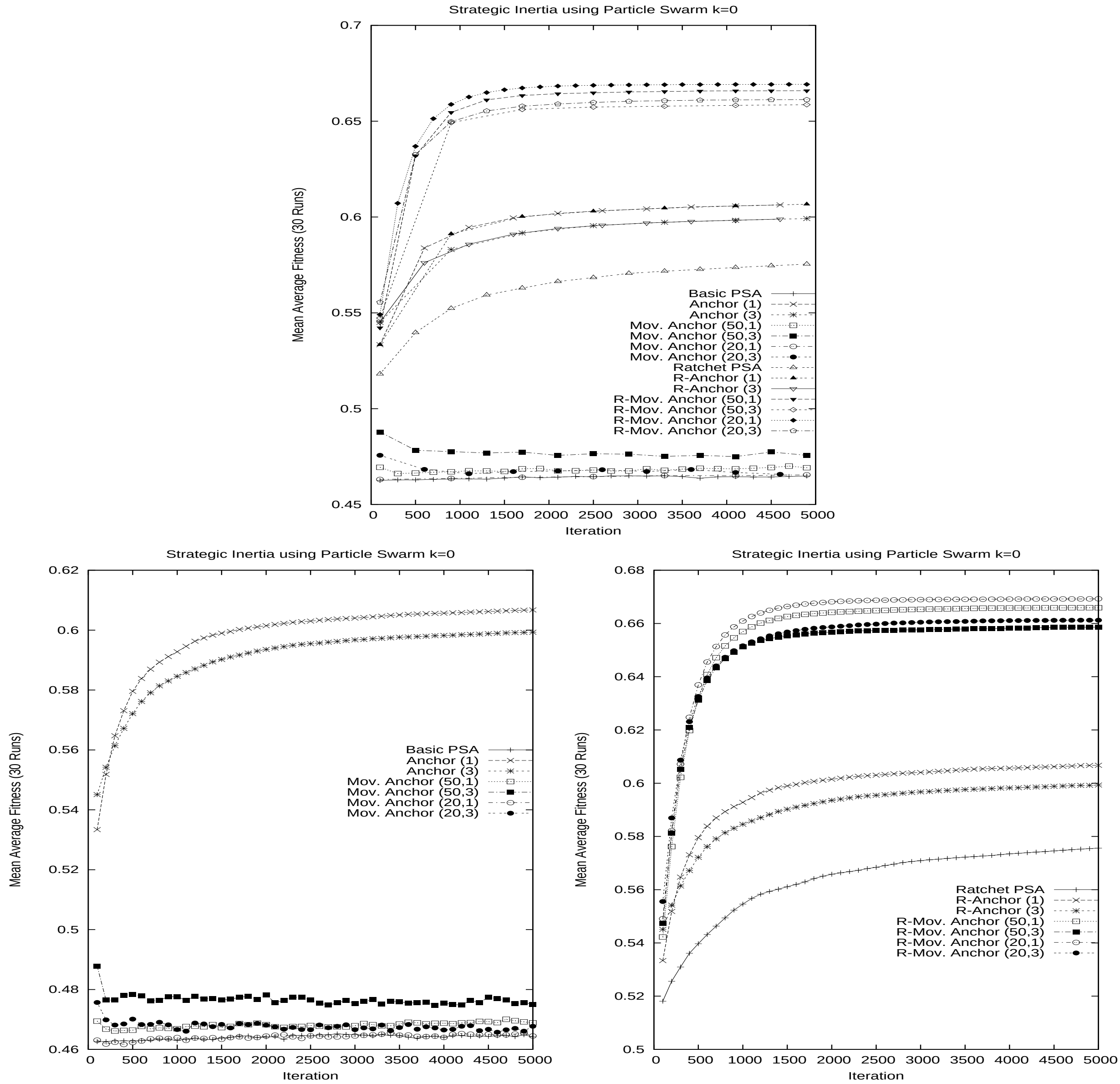

Figure 2: Plot of the mean average fitness on the static landscape where k=0.

on this landscape, is used to construct a simulation model. Adoption of the swarm metaphor allows the incorporation of both social and individual learning mechanisms, and the basic algorithm can be easily adapted to include other search heuristics such as election and inertia.

The results suggest that a degree of strategic inertia, in the presence of an election operator, can assist rather than hamper the adaptive efforts of populations of organizations in static and slowly changing strategic environments. The results also provide an interesting perspective on the claim by [13] that inertia may be a consequence of market-selection processes. The results indicate that there may be good reasons, from a populational perspective, for market selection processes to encourage populations of organizations which exhibit a degree of inertia. Despite the claim for the importance of social learning in populations, the results suggest that social learning alone is of limited benefit, unless supported by an election mechanism.

In the construction of any simulation model, aspects of the real-world system of interest must be omitted. In this study, we omit the cost of making a strategic adjustment,[8] and we omit an explicit birth-death process for the population of organizations.[9] We note that the effect of the gbest, pbest and inertia anchors, is to pin each organization on the landscape. To the extent that the entire collection of organizations have converged to a relatively small region of the landscape, they may find it impossible to migrate

[8]Although we note that incorporating such costs would likely enhance the value of inertia.

[9]It could be argued that although there is no explicit selection process, the effect of including a gbest term in the model is to incorporate an implicit form of selection, in that organizations with poor strategies are drawn towards the location of gbest, mimicking a selection process.

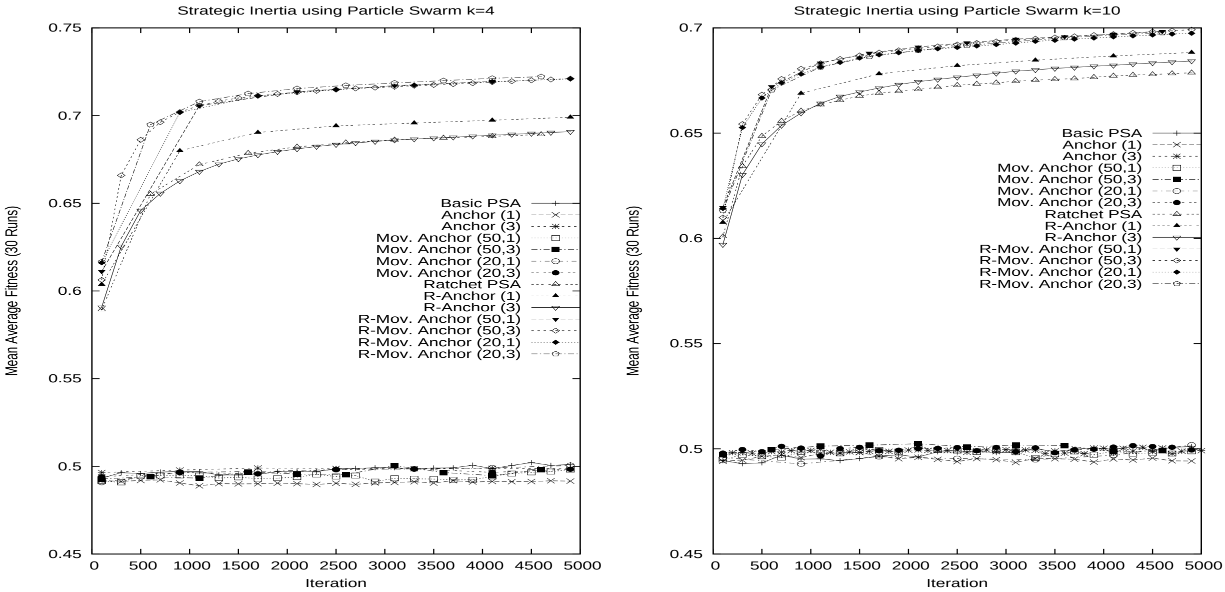

Figure 3: Plot of the mean average fitness on the static landscape where k=4 (left), and where k=10 (right).

to a new high-fitness region of the landscape if that region moves far away from their current location. In real-world environments, this is compensated for by the birth of new organizations.

This study describes the *OrgSwarm* simulator, and reports the results of initial simulations using this model. Future work will extend the range of strategic scenarios, and parameter settings considered. In particular we intend to examine the process of strategic adaptation when strategists make errorful assessments of the fitness of proposed strategies. We also intend to incorporate a co-evolutionary aspect into the model (mimicking direct competition between organizations), wherein the fitness of a strategy is partially determined by the number of organizations which are pursing similar strategies.

References

[1] Abido, M. (2002). Optimal power flow using particle swarm optimization, *Electrical power & Energy Systems*, 24:563-571.

[2] Barnett, W. and Hansen, M. (1996). The Red Queen in Organizational Evolution, *Strategic Management Journal*, 17:139-157.

[3] Blackwell, T. (2003). Swarms in Dynamic Environments, *in Proceedings of GECCO 2003*, Lecture Notes in Computer Science (2723), Springer-Verlag, Berlin, pp. 1-12.

[4] Boeker, W. (1989). Strategic Change: The Effects of Founding and History, *Academy of Management Journal*, 32(3):489-515.

[5] Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). *Swarm Intelligence: From natural to artificial systems*, Oxford: Oxford University Press.

[6] Birchenhall, C. (1995). Modular Technical Change and Genetic Algorithms, *Computational Economics*, 8:233-253.

[7] Brabazon, A., Silva, A., Ferra de Sousa, T., O'Neill, M. and Matthews, R. (2004). A Particle Swarm Model of Organizational Adaptation, *in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), Lecture Notes in Computer Science (3102)*, Deb et. al. Eds., Seattle, USA, June 26-30, 2004, 1:12-23, Berlin: Springer-Verlag.

[8] Brabazon, A., Silva, A., Ferra de Sousa, T., O'Neill, M. and Matthews, R. (2004). Investigating Organizational Strategic Inertia Using a Particle Swarm Model, *in Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 1:652-659, IEEE Press: New Jersey.

[9] Carroll, G. and Hannan, T. (1995). *Organizations in Industry: strategy, structure and selection*, New York: Oxford University Press.

[10] Child, J. (1972). Organizational Structure, Environment and Performance: The Role of Strategic Choice, *Sociology*, 6:2-22.

[11] Christensen, C. (1997). *The Innovator's Dilemma*, (HarperBusiness Essentials, 2003 edition), New York: HarperBusiness Essentials.

[12] Hannan, M. and Freeman, J. (1977). The Populational Ecology of Organizations, *American Journal of Sociology*, 82(5): 929-964.

[13] Hannan, M. and Freeman, J. (1984). Structural Inertia and Organizational Change, *American Sociological Review*, 49:149-164.

[14] Helfat, C. (1994). Evolutionary Trajectories in Petroleum Firm R&D, *Management Science*, 40(12):1720-1747.

[15] Gavetti, G. and Levinthal, D. (2000). Looking Forward and Looking Backward: Cognitive and Experiential Search, *Administrative Science Quarterly*, 45:113- 137.

[16] Hammer, M. and Champy, J. (2001). Reengineering the corporation (revised edition): A manifesto for business revolution, HarperBusiness: New York.

[17] Kaplan, R. and Cooper, R. (1998). *Cost and effect: Using integrated cost systems to drive profitability and performance*, Boston, Massachusetts: Harvard Business School Press.

[18] Kauffman, S. and Levin, S. (1987). Towards a General Theory of Adaptive Walks on Rugged Landscapes, *Journal of Theoretical Biology*, 128:11-45.

[19] Kauffman, S. (1993). *The Origins of Order*, Oxford,England: Oxford University Press.

[20] Kauffman, S., Lobo, J. and MacReady, W. (1998). Optimal Search on a Technology Landscape, *Santa Fe Institute Working Paper 98-10-091*.

[21] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, December 1995, pp. 1942-1948.

[22] Kennedy, J. (1997). The particle swarm: Social adaptation of knowledge, *in Proceedings of the International Conference on Evolutionary Computation*, pp. 303-308, Piscataway, New Jersey:IEEE Press.

[23] Kennedy, J. and Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm, *Proceedings of the Conference on Systems, Man and Cybernetics*, pp. 4104-4109, Piscataway, New Jersey: IEEE Press.

[24] Kennedy, J. (1999). Minds and Cultures: Particle Swam Implications for Beings in Sociocognitive Space, *Adaptive Behavior*, 7(3/4):269-288.

[25] Kennedy, J., Eberhart, R. and Shi, Y. (2001). *Swarm Intelligence*, San Mateo, California: Morgan Kauffman.

[26] Kitts, B., Edvinsson, L. and Beding, T. (2001). Intellectual capital: from intangible assets to fitness landscapes, *Expert Systems with Applications*, 20:35-50.

[27] Levinthal, D. (1991). Random Walks and Organisational Mortality, *Administrative Science Quarterly*, 36:397-420.

[28] Levinthal, D. (1997). Adaptation on Rugged Landscapes, *Management Science*, 43(7):934-950.

[29] Lobo, J. and MacReady, W. (1999). Landscapes: A Natural Extension of Search Theory, *Santa Fe Institute Working Paper 99-05-037*.

[30] Makadok, R. and Walker, G. (1996). Search and Selection in the Money Market Fund Industry, *Strategic Management Journal*, 17:39-54.

[31] March, J. (1981). Footnotes to Organizational Change, *Administrative Science Quarterly*, 26:563-577.

[32] March, J. (1991). Exploration and Exploitation in Organisational Learning, *Organization Science*, 2(1):71-87 .

[33] Nelson, R. and Winter, S. (1982). *An Evolutionary Theory of Economic Change*, Cambridge, Massachusetts, Harvard University Press.

[34] Ourique, C., Biscaia, E. and Pinto, J. (2002). The use of particale swarm optimization for dynamical analysis in chemical processes, *Computers and Chemical Engineering*, 26:1783-1793.

[35] Porter, M. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*, New York, The Free Press.

[36] Porter, M. (1996). What is Strategy?, *Harvard Business Review*, Nov-Dec, 61-78.

[37] Porter, M. and Siggelkow, N. (2001). Contextuality within Activity Systems,*Harvard Business School Working Paper Series*, No. 01-053, 2001.

[38] Rivkin, J. (2000). Imitation of Complex Strategies, *Management Science*, 46(6):824-844.

[39] Silva, A., Neves, A. and Costa, E. (2002). An empirical comparision of particle swarm and predator prey optimisation, in *Lecture Notes in Artificial Intelligence (2464) - Proceedings of AICS 2002*, edited by M. O'Neill, R.F.E. Sutcliffe, C. Ryan, M. Eaton, N.J.L. Griffith Springer-Verlag, Berlin, pp. 103-110.

[40] Simon, H. (1993). Strategy and Organizational Evolution, *Strategic Management Journal*, 14:131-142.

[41] Strumsky, D. and Lobo, J. (2002). If it isn't broken, don't fix it: Extremal search on a technology landscape, *Santa Fe Institute Working Paper 03-02-003*.

[42] Stuart, T. and Podolny, J. (1996). Local Search and the Evolution of Technological Capabilities, *Strategic Management Journal*, 17:21-38.

[43] Sull, D. (1999). Why Good Companies Go Bad, *Harvard Business Review*, 77(4):42-52.

[44] Tanev, I. and Shimohara, K. (2003). On Role of Implicit Interaction and Explicit Communications in Emergence of Social Behaviour in Continuous Predators-Prey Pursuit Problem, *in Proceedings of GECCO 2003, Lecture Notes in Computer Science (2723)*, Springer-Verlag, Berlin, pp. 74-85.

[45] Tushman, M. and O'Reilly, C. (1996). Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change, *California Management Review*, 38(4):8-30.

[46] Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution, *Proceedings of the Sixth International Congress on Genetics*, 1:356-366.

Algorithm	**Fitness** (N=96, K=0)	**(N=96, K=4)**	**(N=96, K=10)**
Basic PSA	0.4641 (0.5457)	0.5002 (0.6000)	0.4991 (0.6143)
Initial Anchor, w=1	0.4699 (0.5484)	0.4921 (0.5967)	0.4956 (0.6102)
Initial Anchor, w=3	0.4943 (0.5591)	0.4994 (0.5979)	0.4991 (0.6103)
Mov. Anchor (50,1)	0.4688 (0.5500)	0.4960 (0.6003)	0.4983 (0.6145)
Mov. Anchor (50,3)	0.4750 (0.5631)	0.4962 (0.6122)	0.5003 (0.6215)
Mov. Anchor (20,1)	0.4644 (0.5475)	0.4986 (0.6018)	0.5001 (0.6120)
Mov. Anchor (20,3)	0.4677 (0.5492)	0.4994 (0.6156)	0.4994 (0.6229)
Rachet PSA	0.5756 (0.6021)	0.6896 (0.7143)	0.6789 (0.7035)
R-Initial Anchor, w=1	0.6067 (0.6416)	0.6991 (0.7261)	0.6884 (0.7167)
R-Initial Anchor, w=3	0.5993 (0.6361)	0.6910 (0.7213)	0.6844 (0.7099)
R-Mov. Anchor (50,1)	0.6659 (0.6659)	0.7213 (0.7456)	0.6990 (0.7256)
R-Mov. Anchor (50,3)	0.6586 (0.6601)	0.7211 (0.7469)	0.6992 (0.7270)
R-Mov. Anchor (20,1)	0.6692 (0.6695)	0.7211 (0.7441)	0.6976 (0.7243)
R-Mov. Anchor (20,3)	0.6612 (0.6627)	0.7228 (0.7462)	0.6984 (0.7251)

Table 1: Average (environment best) fitnesses after 5,000 iterations, static landscape.

Algorithm	**Fitness** (N=96, K=0)	**(N=96, K=4)**	**(N=96, K=10)**
Basic PSA	0.4667 (0.5245)	0.4987 (0.5915)	0.4955 (0.6065)
Initial Anchor, w=1	0.4658 (0.5293)	0.4908 (0.5840)	0.4957 (0.6038)
Initial Anchor, w=3	0.4922 (0.5513)	0.4992 (0.5953)	0.5001 (0.60852)
Mov. Anchor (50,1)	0.4614 (0.5200)	0.4975 (0.5927)	0.5008 (0.6044)
Mov. Anchor (50,3)	0.4691 (0.5400)	0.4975 (0.6040)	0.4987 (0.6174)
Mov. Anchor (20,1)	0.4686 (0.5315)	0.5010 (0.6002)	0.4958 (0.6099)
Mov. Anchor (20,3)	0.4661(0.5434)	0.4964(0.6084)	0.4988 (0.6137)
Rachet PSA	0.5783 (0.6056)	0.6859 (0.7096)	0.6808 (0.7066)
R-Initial Anchor, w=1	0.6207 (0.6553)	0.6994 (0.7330)	0.6895 (0.7142)
R-Initial Anchor, w=3	0.5927 (0.6239)	0.6900 (0.7182)	0.6850 (0.7140)
R-Mov. Anchor (50,1)	0.6676 (0.6688)	0.7187 (0.7438)	0.6987 (0.7241)
R-Mov. Anchor (50,3)	0.6696 (0.6696)	0.7203 (0.7462)	0.6989 (0.7264)
R-Mov. Anchor (20,1)	0.6689 (0.6694)	0.7193 (0.7426)	0.6974 (0.7251)
R-Mov. Anchor (20,3)	0.6594 (0.6622)	0.7221 (0.7450)	0.6987 (0.7280)

Table 2: Average (environment best) fitnesses after 5,000 iterations, 1 dimension respecified periodically.

Algorithm	**Fitness** (N=96, K=0)	**(N=96, K=4)**	**(N=96, K=10)**
Basic PSA	0.4761 (0.5428)	0.4886 (0.5891)	0.4961 (0.6019)
Initial Anchor, w=1	0.4819 (0.5524)	0.4883 (0.5822)	0.4982 (0.6075)
Initial Anchor, w=3	0.5021 (0.5623)	0.4967 (0.5931)	0.4998 (0.6047)
Mov. Anchor (50,1)	0.4705 (0.5450)	0.4894 (0.5863)	0.4974 (0.6008)
Mov. Anchor (50,3)	0.4800 (0.5612)	0.4966 (0.6053)	0.5010 (0.6187)
Mov. Anchor (20,1)	0.4757 (0.5520)	0.4926 (0.5867)	0.4985 (0.6097)
Mov. Anchor (20,3)	0.4824 (0.5632)	0.4986 (0.6041)	0.5004 (0.6163)
Rachet PSA	0.5877 (0.6131)	0.6802 (0.7092)	0.6754 (0.7015)
R-Initial Anchor, w=1	0.6187 (0.6508)	0.6874 (0.7180)	0.6764 (0.7070)
R-Initial Anchor, w=3	0.6075 (0.6377)	0.6841 (0.7130)	0.6738 (0.7017)
R-Mov. Anchor (50,1)	0.6517 (0.6561)	0.7134 (0.7387)	0.6840 (0.7141)
R-Mov. Anchor (50,3)	0.6597 (0.6637)	0.7049 (0.7304)	0.6925 (0.7225)
R-Mov. Anchor (20,1)	0.6575 (0.6593)	0.7152 (0.7419)	0.6819 (0.7094)
R-Mov. Anchor (20,3)	0.6689 (0.6700)	0.7158 (0.7429)	0.6860 (0.7147)

Table 3: Average (environment best)fitnesses after 5,000 iterations, entire landscape respecified periodically.

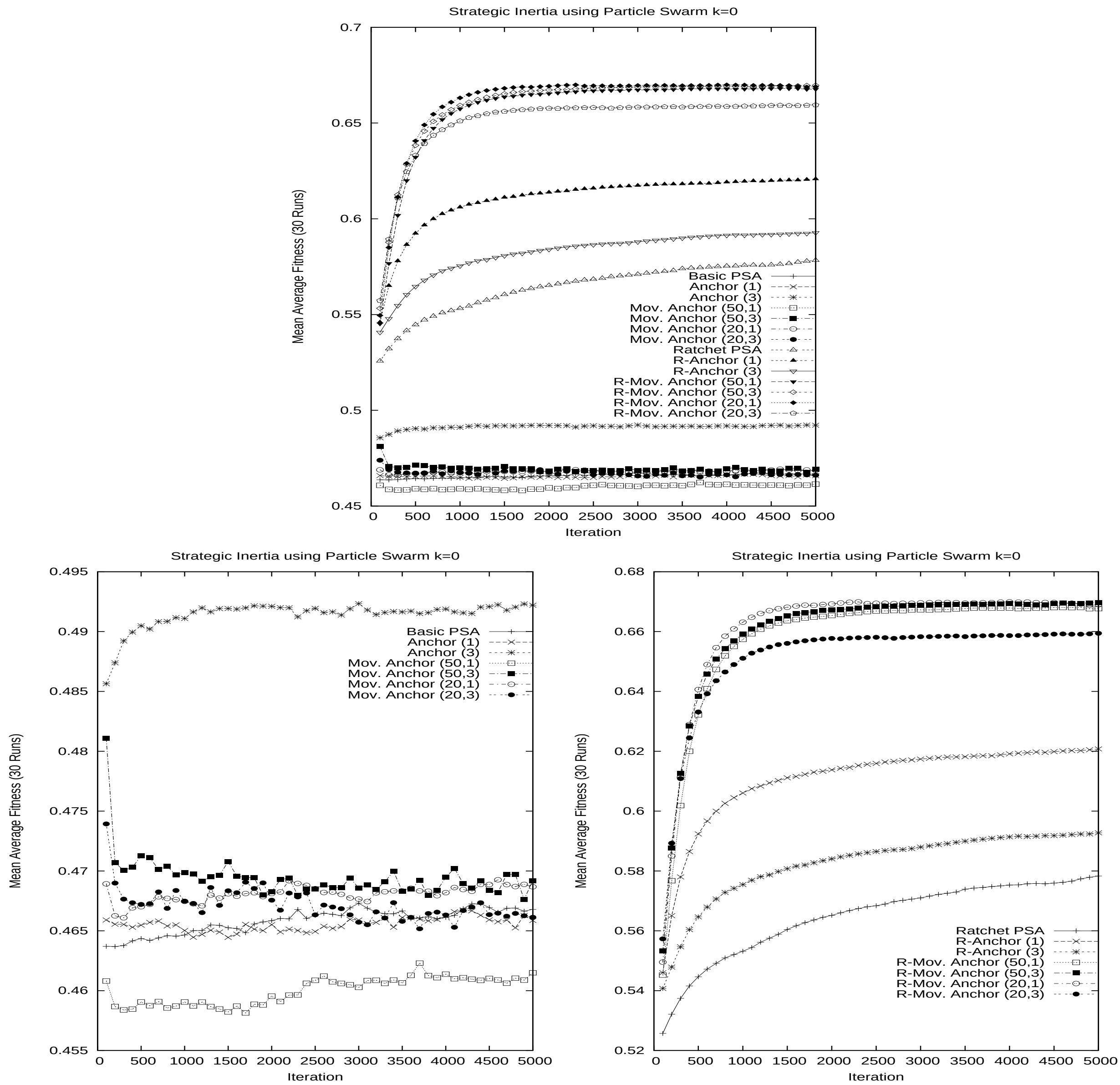

Figure 4: Plot of the mean average fitness on the dynamic landscape (one dimension of the landscape is respecified periodically) where k=0.

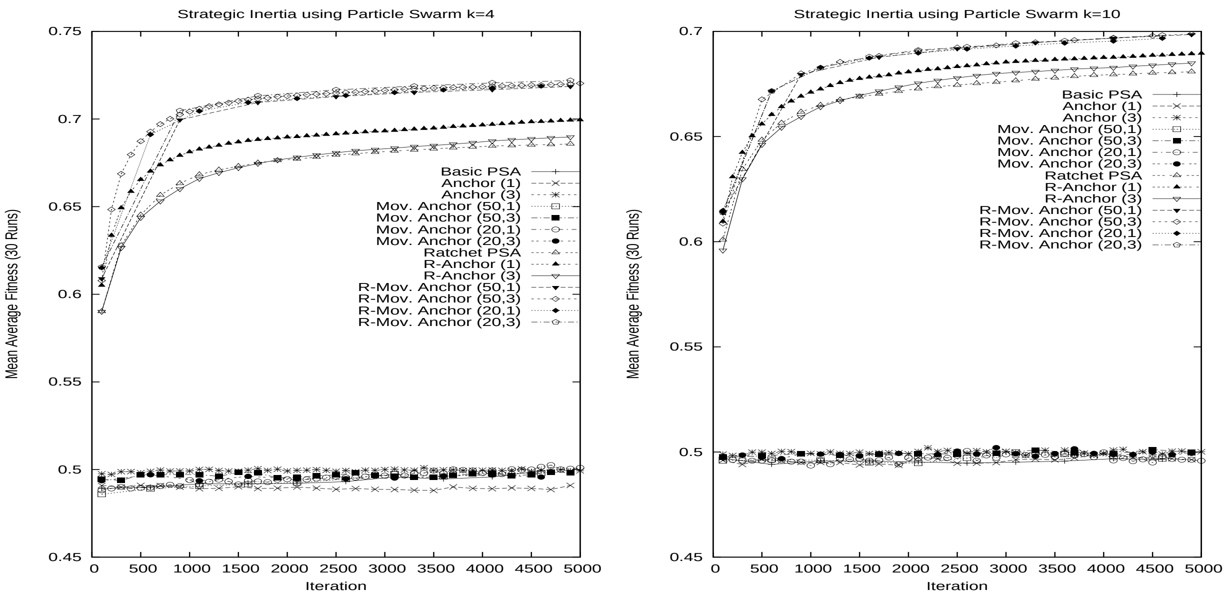

Figure 5: Plot of the mean average fitness on the dynamic landscape (one dimension of the landscape is respecified periodically) where k=4 (left), and where k=10 (right).

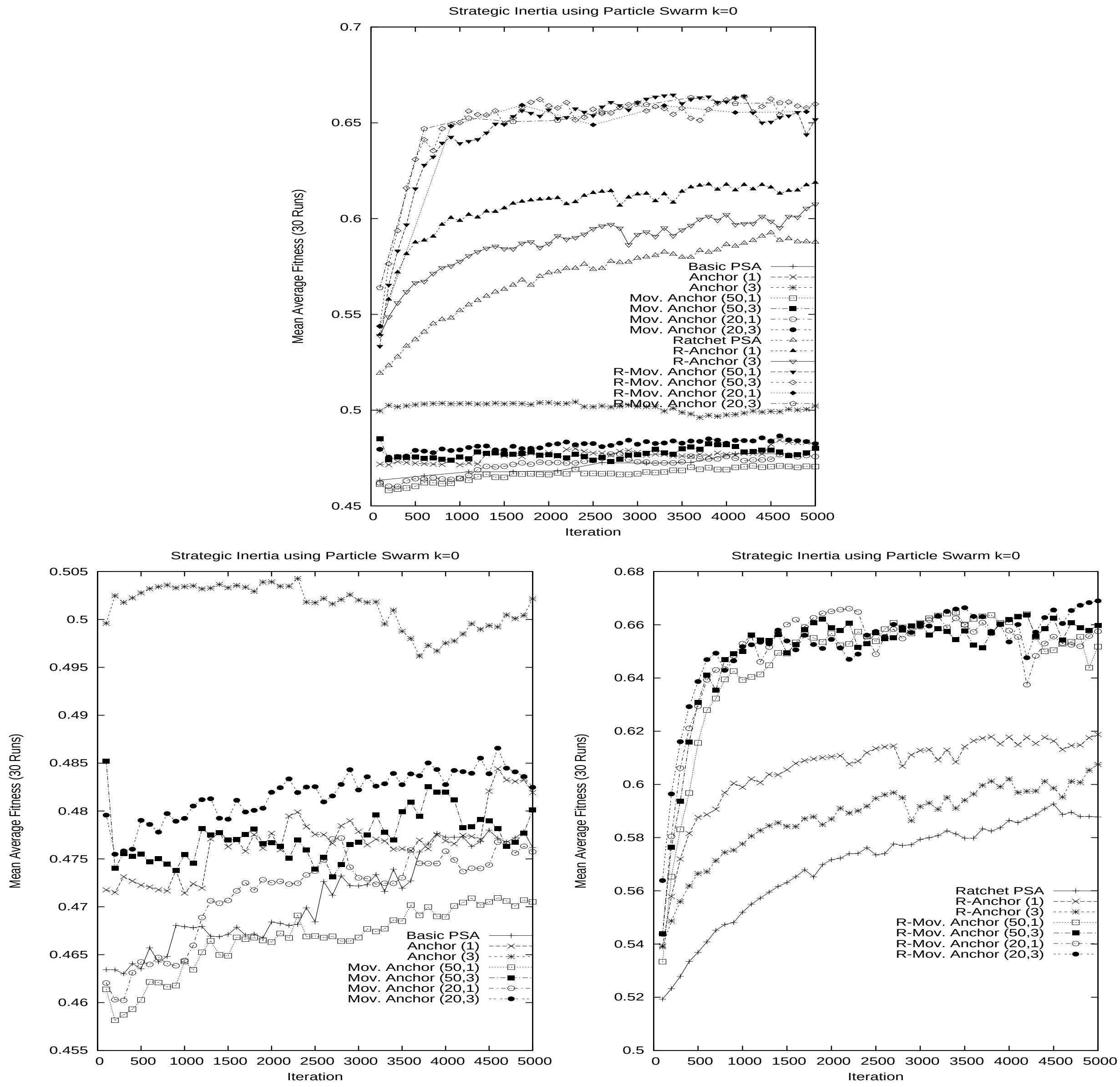

Figure 6: Plot of the mean average fitness on the dynamic landscape (entire landscape respecified periodically) where k=0.

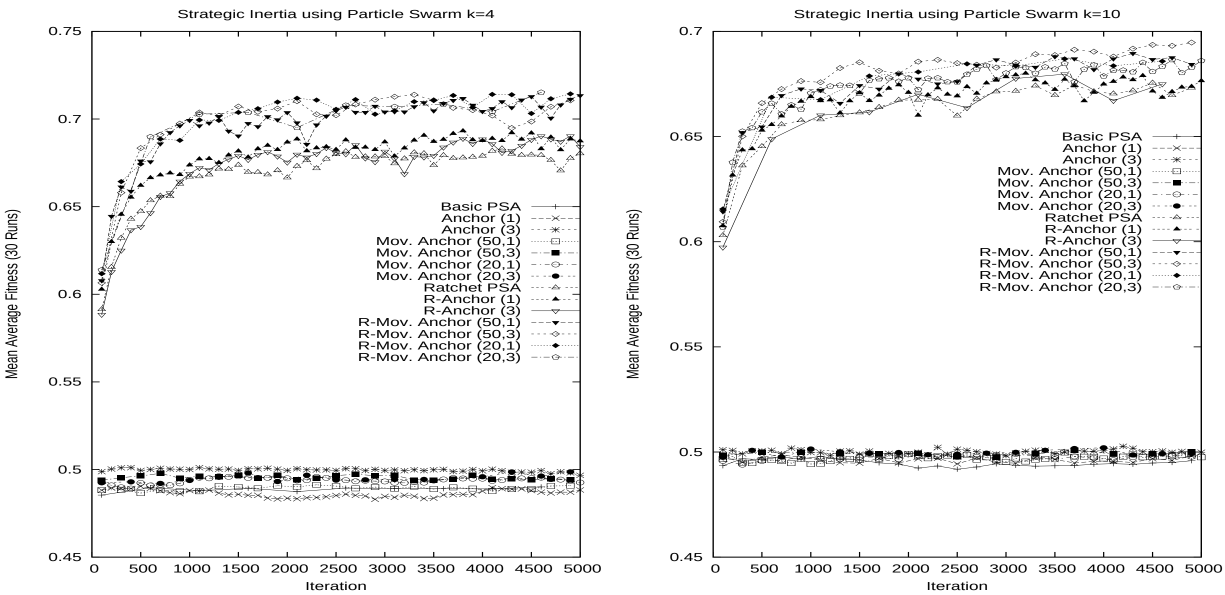

Figure 7: Plot of the mean average fitness on the dynamic landscape (entire landscape respecified periodically) where k=4 (left), and where k=10 (right).

Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm

André L. Vizine[1,2], Leandro N. de Castro[1,2], Eduardo R. Hruschka[1], Ricardo R. Gudwin[2]

[1]Catholic University of Santos (UniSantos)
R. Carvalho de Mendonça, 144, 11070-906, Santos/SP, Brasil
{vizine,lnunes,erh}@unisantos.br

[2]State University of Campinas (Unicamp)
DCA–FEEC–UNICAMP, Cx. Postal 6101, 13083-852, Campinas /SP, Brazil.
gudwin@dca.fee.unicamp.br

Keywords: Ant clustering algorithm, Data clustering, Visual data mining

Received: July 15, 2004

Among the many bio-inspired techniques, ant-based clustering algorithms have received special attention from the community over the past few years for two main reasons. First, they are particularly suitable to perform exploratory data analysis and, second, they still require much investigation to improve performance, stability, convergence, and other key features that would make such algorithms mature tools for diverse applications. Under this perspective, this paper proposes both a progressive vision scheme and pheromone heuristics for the standard ant-clustering algorithm, together with a cooling schedule that improves its convergence properties. The proposed algorithm is evaluated in a number of well-known benchmark data sets, as well as in a real-world bioinformatics dataset. The achieved results are compared to those obtained by the standard ant clustering algorithm, showing that significant improvements are obtained by means of the proposed modifications. As an additional contribution, this work also provides a brief review of ant-based clustering algorithms.

Povzetek: Članek opisuje izboljšan algoritem grupiranja na osnovi pristopa kolonij mravelj.

1 Introduction

Over the past few years, several different types of biologically inspired algorithms have been proposed in the literature (Paton, 1994; de Castro & Von Zuben, 2004). Among these, some have obtained special attention from the scientific community, such as those based on swarm systems (Bonabeau et al., 1999; Kennedy et al., 2001), which are inspired by the social behavior of living organisms. This relatively new field of investigation has originated different types of algorithms for the solution of complex problems in many different domains. Under this perspective, the problems usually tackled involve search, optimization, and data analysis tasks. The main reasons by which swarm based approaches are useful for solving such problems are (Bonabeau et al., 1999; Kennedy et al., 2001): (i) they require little information about the problem at hand (e.g. in clustering problems a data set to be grouped); and (ii) they usually can perform both broad and parallel searches over the space of potential solutions by means of a population (swarm) of candidate solutions.

Despite the broad usefulness of current bio-inspired algorithms, most of them can be further improved, mainly to enhance performance and applicability. In this sense, this work focuses on ant-based clustering algorithms, whose main underlying concepts are based on the way real ants clean their nests and organize dead bodies in their colonies. Considering a more practical computational perspective, these algorithms are basically designed by considering the concept of a 2D grid where objects (data) are laid at random and then automatically organized. A set of ant-like agents is allowed to move throughout the grid, picking up and dropping objects (data) based on their similarity degree within a certain neighborhood.

One difficulty in applying ant-clustering algorithms to solve complex problems comes from the fact that, in most cases, they generate a number of clusters that is much larger than the *natural* number of clusters. Furthermore, these algorithms usually do not stabilize in a particular clustering solution; that is, they constantly construct and deconstruct clusters during the iterative procedure of adaptation. In order to overcome the aforementioned difficulties and, consequently, improve the quality of the results obtained, we propose an Adaptive Ant-Clustering Algorithm (A^2CA), which is more robust in terms of the number of clusters found and tends to converge into good solutions while the clustering process evolves. To achieve these goals, three main modifications are introduced in the standard ant-clustering algorithm proposed by Lumer and Faieta (1994): (i) a cooling schedule for the parameter that controls the probability of ants picking up objects from the grid; (ii) a progressive vision field that allows ants to 'see' over a wider area; and (iii) the use of a pheromone function added to the grid as a way to promote reinforcement for the dropping of objects at more dense regions of the grid. These modifications favor an adaptive clustering process, in the sense that the proposed algorithm tends to converge to stable clusters. In addition to the contributions to the algorithm itself, this paper also brings a brief historical review of ant-based clustering algorithms, emphasizing their main features when compared with the standard ant-clustering algorithm proposed by Lumer and Faieta (1994).

The paper is organized as follows. Section 2 provides a brief review of the standard ant-clustering algorithm (Lumer & Faieta, 1994), which, for the sake of brevity, is referred to as SACA in this work. In Section 3, we present our proposed algorithm (A^{2}CA), which, in Section 4 is experimentally compared to the SACA in three synthetic and one real-world dataset. Section 5 provides a brief survey of related works, whereas Section 6 concludes the paper and points out some avenues for future work.

2 Standard Ant Clustering Algorithm: SACA

The Standard Ant Clustering Algorithm (SACA), introduced by Lumer and Faieta (1994), assumes that ants perform random walks on a two-dimensional grid on which objects (data) are laid down at random. Independently of the dimension of the input data, each datum is randomly projected onto a cell of the grid. A grid cell (or patch) is thus responsible for hosting the index of a specific input pattern, indicating the relative position of the datum in the two-dimensional grid. The general idea is to have items, which are similar in their original N-dimensional space, in neighboring regions of the grid. In other words, data indices that are neighbors in the grid indicate patterns that are similar in their original space of attributes. In this context, it is assumed that each site or cell on the grid can be occupied by at most one object, and one of the two following situations may occur: (i) one ant holds an object i and evaluates the probability of dropping it in its current position; (ii) an ant is unloaded and evaluates the probability of picking up an object. At each discrete time step, an ant is selected at random and can either pick up or drop an object at its current location.

The probability of picking up an object increases with low-density neighborhoods and decreases with high similarity among objects in the surrounding area. The probability of dropping an object, by contrast, increases with high densities of similar objects in the neighborhood. More specifically, assume that $d(i,j)$ is the Euclidean distance between objects i and j in their N-dimensional space. The density dependent function for object i, at a particular grid location, is defined by the following expression:

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_j (1 - d(i,j)/\alpha) & \text{if } f(i) > 0 \\ 0 & \text{otherwise.} \end{cases}, \qquad (1)$$

where s^2 is the number of cells in the surrounding area of i, and α is a constant that scales the dissimilarities among objects. The maximum value for $f(i)$ is obtained if, and only if, all the sites in the neighborhood are occupied by equal objects. Assuming the density dependent function presented in Eq. (1), the probability of picking up and dropping an object i is given by Eqs. (2) and (3), respectively:

$$P_{pick}(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2, \qquad (2)$$

$$P_{drop}(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d; \\ 1 & \text{otherwise.} \end{cases}, \qquad (3)$$

where the parameters k_p and k_d are threshold constants equal to 0.1 and 0.15, respectively. Note that $f(i) \in [0,1]$. Thus, if $f(i) << k_p$, then $P_{pick} \approx 1$, leading to high probabilities of picking up objects in low density regions. Similarly, $P_{pick} \approx 0$ if $f(i) >> k_p$, meaning that objects are unlikely to be removed from dense regions. In the case of P_{drop}, it is also possible to observe that if $f(i) << k_d$, $P_{drop} \approx 0$, whereas if $f(i) \geq k_d$ the ant drops the object.

Whenever a loaded ant decides to drop the object it is carrying, it looks for the first empty cell in its vicinity in which to do so (its current position can be already occupied by another object). A time step finishes with the selected ant moving to one of its four adjacent nodes, each direction of motion being equally likely.

3 Adaptive Ant Clustering Algorithm: A^{2}CA

The Adaptive Ant Clustering Algorithm (A^{2}CA) was developed by taking further inspiration from biological systems. In particular, A^{2}CA was inspired by the fact that termites, while building their nests, deposit pheromone on soil pellets and this serves as a reinforcement signal to other termites placing more pellets on the same region of the space (Camazine et al., 2001). Another biological observation taken into account while developing A^{2}CA was the fact that ants can sense not only its immediate neighborhood environment, but a broader range that may vary from ant to ant and with time. Therefore, A^{2}CA has two main modifications in relation to SACA: (i) a progressive vision scheme, and (ii) the inclusion of pheromone on the grid cells. In addition, we adopt a cooling schedule for the parameter that drives the picking probability (k_p).

3.1 Cooling Schedule for k_p

In addition to the modifications that led to the development of A^{2}CA, one simple modification was previously introduced in SACA so as to improve its convergence properties (Vizine et al., 2005) and it is also adopted in our proposed approach (A^{2}CA). In a nutshell, a cooling schedule for the parameter that drives the picking probability k_p – Eq. (2) – is employed. The adopted scheme is simple: after one cycle (10,000 ant steps) has passed, the value of the parameter k_p starts being geometrically decreased, at each cycle, until it reaches a minimal allowed value, $k_{p\min}$, which corresponds to the stopping criterion for the algorithm. In the current implementation, k_p is cooled based on a geometric scheme presented in Eq. (4). It is important to emphasize that the SACA implementation used in this work also incorporates this *extra feature*, leading to the so-called SACA*. By doing so, more suitable and fair comparisons can be performed, in the sense that SACA* will also tend to converge to better clustering solutions.

$$k_p \leftarrow k_p \times 0.98, \quad k_{p\min} = 0.001. \tag{4}$$

3.2 Progressive Vision

In SACA, the value of the density function, $f(i)$, given by Eq. (1), depends on the vision field, s^2, of each ant. The definition of a fixed value for s^2 may sometimes cause inappropriate behaviors, because a fixed perceptual area does not allow distinguishing between clusters of different sizes. A small area of vision implies a small perception of the cluster at a global level. Thus, small clusters and large clusters are all the same in this sense, for the agent only perceives a limited area of the environment. In some problems, the use of a too restrictive perception field may be limiting, whereas a too broad vision may cause undesirable merging of groups. On the one hand, even if a cluster is perfectly homogeneous (with identical elements) and sufficiently large, there still exists a small probability that an agent picks up a datum from the cluster and drops it somewhere else. On the other hand, a large vision field may be inefficient in the initial iterations, when the data elements are scattered at random on the grid, because analyzing a broad area may imply in analyzing a large number of small clusters simultaneously.

In order to overcome this difficulty, a progressive vision scheme was proposed for SACA as follows (Sherafat et al., 2004a). When an ant perceives a ‘big’ cluster, it increments its perception field (s_i^2) up to a maximal size. Now, s_i^2 is a specific parameter for each ant that will be dynamically and independently updated while running the algorithm. The question that remains is: ‘How can an ant agent detect the size of a cluster so as to control the size of its vision field?’

We tackled this problem by using the density dependent function $f(i)$ as a control parameter. There is a relationship between the size of a cluster and its density dependent function: the average value of $f(i)$ increases as the clustering proceeds, and this happens because larger clusters tend to be formed. When $f(i)$ achieves a value greater than a pre-specified threshold θ, the parameter s^2 is incremented by n_s units until it reaches its maximum value.

$$\text{If } f(i) > \theta \text{ and } s^2 \le s^2_{\max}, \text{ then } s^2 \leftarrow s^2 + n_s. \tag{5}$$

where $s^2_{\max} = 7 \times 7$ and $\theta = 0.6$ in our implementation.

3.3 Pheromone Heuristics

In order to perform data clustering, the SACA takes into account the relative distance among all objects within the vision field of the ant. A problem with this approach is that it does not account for the work in progress at a global level. One form of overcoming this difficulty was proposed by Sherafat et al. (2004a,b). The method is based on the introduction of a local variable $\phi(i)$ associated with each bi-dimensional position, i, on the grid, such that the quantity of pheromone in that exact position becomes a function of the presence or absence of an object at i. Inspired by the way termites use pheromone to build their nests, the artificial agents in the modified ant clustering algorithm will add some pheromone to the objects they carry and this pheromone will be transferred to the grid when an object is deposited. During each iteration, the artificial pheromone $\phi(i)$ at each cell of the grid evaporates at a fixed rate.

Sherafat et al. (2004a,b) introduced a pheromone function, $Phe(\phi_{\max},\phi_{\min},P,\phi(i))$, given by Eq. (6), that influences the probability of picking up and dropping off objects from and on the grid. The proposed pheromone function varies linearly with the pheromone level at each grid position, $\phi(i)$, and depends on a number of user-defined parameters, such as the $\phi_{\max}$ and $\phi_{\min}$ values of pheromone perceived by the agent, and the maximal influence of pheromone allowed, P.

$$Phe(.) = \frac{2.P}{\phi_{max} - \phi_{min}}\phi(i) - \frac{2.P.\phi_{max}}{\phi_{max} - \phi_{min}} + P, \tag{6}$$

To accommodate the addition of pheromone on the grid, some variations on the picking and dropping probability functions of SACA were proposed in (Sherafat et al., 2004a,b), as described in Eqs. (7) and (8), respectively:

$$P_{pick}(i) = (1 - Phe(\phi_{\min}, \phi_{\max}, P, \phi(i))) \times \left(\frac{k_p}{k_p + f(i)} \right)^2. \tag{7}$$

$$P_{drop}(i) = (1 + Phe(\phi_{\min}, \phi_{\max}, P, \phi(i))) \times \left(\frac{f(i)}{k_d + f(i)} \right)^2. \tag{8}$$

where $\phi_{\max}$ represents the current largest amount of pheromone perceived by this agent; $\phi_{\min}$ corresponds to the current smallest amount of pheromone perceived by this agent; P is the maximum influence of the pheromone in changing the probability of picking and dropping data elements; and $\phi(i)$ is the quantity of pheromone in the current position i.

Note that in Eq. (8) the dropping probability originally derived from the model of Deneubourg et al. (1991) was employed. Basically, this choice was made because the algorithm presented superior performance when using the function proposed by Deneubourg et al. (1991) – given by Eq. (9) - instead of Eq. (3) for the dropping probability. This was also the case for SACA. Therefore,