Path Planning Strategies Inspired by Swarm Behaviour of Plant Root Apexes

Final Report

Authors: Luís F. Simões, Cristina Cruz, Rita A. Ribeiro, Luís Correia, Tobias Seidl, Christos Ampatzis, Dario Izzo

Affiliation: UNINOVA-CA3, Computational Intelligence Research Group, New University of Lisbon, Faculty of Sciences, University of Lisbon, Advanced Concepts Team, European Space Agency

Date: 31/01/2011

Contacts:

Rita A. Ribeiro
Tel: +351 212949625
Fax: N/A
e-mail: rar@uninova.pt

Advanced Concepts Team
Tel: +31(0)715655174
Fax: +31(0)715658018
e-mail: act@esa.int

Ariadna ID: 09/6401
Contract Number: 22710

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Distributed control

Exploring unknown environments and identifying potentially interesting or hazardous areas is a challenging task for an autonomous agent. In the absence of a priori provided maps or landmarks guiding navigation, researchers are considering multi-agent systems trying to exploit the inherent parallelism of such systems. Many scientific research works following this direction draw inspiration from biological swarm models. In such models, self-organised exploration strategies emerge at the collective level as a result of simple rules followed by individual agents. To produce the global behaviour, individuals interact by using simple (often indirect or stigmergic) and mostly local communication protocols. Social insects are a good biological example of organisms collectively exploring an unknown environment, and they have often served as a source of direct inspiration for research on self-organized cooperative robotic exploration and path formation in groups of robots using swarm intelligence techniques (e.g. [Payton et al., 2003; Svennebring and Koenig, 2004; Schmickl et al., 2009; Nouyan et al., 2008]). The benefit of such distributed techniques lies in the fact that they produce robust and scalable systems, contrary to traditional approaches often based on centralised architectures and map-like representation of the environment.

1.2 Animal swarms

Swarms are formed when individuals collocate to a higher order by using simple rules (in fish e.g.: align with next fish, keep speed and distance). Swarm formation is ubiquitous in nature and it is observed not only in highly developed vertebrate animals, but also in insects (ants: [Hölldobler and Wilson, 2009], wasps: [Gadagkar, 2001]). From an engineering point of view, it is beneficial to analyse the global patterns observed and subsequently to break them down into a set of simple rules governing individual agents, generating the complex global behaviour. Recently, [Couzin et al., 2005; Couzin, 2007] proposed a model for the flocking behaviour of fish schools by essentially identifying a minimal amount of information that allows biological complex behaviours at the collective level to be digitally reproduced. The a) repulsion, b) attraction, c) heading alignment laws do not have a proven biological origin, yet they are incredibly successful in generating (in digital simulations) behavioural patterns similar to the ones of real swarms. Thus, it makes sense to consider this set of basic behaviours as a starting point to design artificial multi-agent systems displaying flocking properties (e.g [Izzo and Pettazzi, 2007]).

1.3 Bottom-up and top-down generation of swarm controllers

The behavior-based control mentioned above is a bottom-up approach to the design of swarm controllers which exploits a deep engineering knowledge of the problem to construct the global behavior. The local interaction rules among agents are engineered directly as to obtain a predefined collective behavior (in this case, formation control). On the other hand, the Evolutionary Robotics (ER) methodology [Nolfi and Floreano, 2000] allows for an implementation of a top-down approach. ER is constantly gaining momentum in the collective robotics community as it aims at a completely automated design of controllers. The main tool of ER is artificial neural networks' (ANNs) reinforcement learning via evolutionary optimisation techniques, however, other control structures, such as rules bases can be used. The assessment of the system's performance does not take place via a decomposition of the collective behaviour into individual behaviors that are subsequently evaluated in isolation; instead, the system is evaluated as a whole, without reference to how individuals perform. Also, ER does not need assumptions about behavioural mechanisms agents should use, as those are shaped by artificial evolution [Nolfi and Floreano, 2000; Ampatzis et al., 2009]. Overall, we can say that this technique complements (and contrasts) approaches based on the engineering of the local rules at the level of the individual, relying on principles of stochastic optimization to obtain local rules which self-organize into a global structure.

Arguably, the complexity, flexibility and adaptation demonstrated by natural collective systems is unmatched by any man-made system. In consequence, it may make sense to venture into the analysis of real biological swarm models, beyond the obvious analogies and beyond simply drawing inspiration by biological phenomena such as natural selection or flocking. The rationale behind the research work outlined in this report is to combine the properties of automatic design techniques for swarm behaviour (ER) with the imposition on the swarm of biologically realistic behavioural patterns.

1.4 Do only animals form swarms?

A generalized definition considers swarms as large groups of simple autonomous agents interacting locally and hence may in principle include less developed organisms such as bacteria, see e.g., [Atkinson and Williams, 2009]. From here, it is only a small step to consider a plant or parts of a plant, like the root system, as a swarm of autonomously acting units. Decisions of directional growth are taken in the apexes, the tips of a root (or shoot). Indeed, in earlier times philosophers like Erasmus Darwin, the grandfather of Charles Darwin, referred to plants as swarms ([Darwin, 1800] cited in [White, 1979]), describing the almost autonomous behavior of single roots and shoots forming the plant as a whole. The physiological union of the tips of the roots of a plant (apexes) serves the greater task of supporting and nurturing the plant (among others). In fact, all directional growth decisions and a majority of environmental sensing are made in the apex. Growth patterns of roots are basically influenced by gravity, genetics, soil conditions, and distribution of nutrients (H_2O, CO_2, minerals, etc.). Since there is no anatomic evidence for a central sensing and decision unit and considering the rather low computational capacity of a plant cell (compared to neuronal systems of animals, for example), it appears meaningful to consider the apex as a simple autonomous unit taking decisions on own account [Baluška et al., 2004]. There is some evidence of communication between apexes ([Davies and Zhang, 1991; Ali et al., 1999]), but a higher, centralized brain has never been observed in plants. Yet, when looking at the root as a collective, growth patterns are not chaotic, but seem to follow a higher order, and emerge as a result of the individual decision-making of the apexes.

1.5 Complexity of a root–soil system

A root swarm is situated in a fundamentally different setting than fish, birds or insects. The sensory capabilities as well as the computational powers are tremendously reduced and on top of that the mediums properties puts additional constraints. Therefore, a simple rule like align with your neighbor which may be in place in fish schools, may be difficult to follow for an apex in the absence of elaborated sensors and being situated in the sensing-inhibiting soil. This study thus challenges the general applicability of the hitherto known rule-sets for swarm modeling and focuses on delivering strategies for diffuse sensing capabilities and coping with unreliable and heterogeneous substrates.

Trying to infer basic operational principles from plants, and in this case root swarms, for implementation on engineering the design of efficient exploration algorithms has the advantage that the exploration strategys blueprint is imprinted on the root and directly observable. Contrary to other biological systems for which the experimental trials have to be observed live in order to deduce patterns in the exploration strategies, in the case of roots these strategies are available and at our disposal right from the start. Even if the social insect metaphor is straightforward for implementation on engineering the design of efficient exploration algorithms, it suffers from an inherent disadvantage: the exploration of an unknown terrain (or volume) is done before the discovery of food sites, etc. and hence is very difficult to systematically observe. If we look at the example of an ant colony optimisation – here the exploitation of known food sites is regulated in a simple and elegant manner. However, little is known how the first ants actually located the food sites in the first spot. Obviously the "first" ant could not make use of pre-existing marked trails. Here, only the research on individually foraging desert ants of the genus *Cataglyphis* might shed light onto this fascinating problem from an insect point of view [Seidl, 2009; Wehner, 2008]. However, also here each single ant leaving the nest has to be traced individually and so only little data is available. On top of that, *Cataglyphis* continues to exploit sites individually with only little communication. The root-system combines an easily observable foraging behaviour as well as a complex communication structure during exploration and exploitation.

1.6 Overview of the study

In this research, we modeled a root swarm system based on simple agents and used its principles to design a swarm robotic controller. The root model was optimised as a multi-agent system with several goals and tasks. In our case, these tasks are the simultaneous exploration and exploitation of the resources present in the soil where the root lives and grows. Using a bottom-up approach we obtained control structures for individual apexes, that when cloned on all apexes can reproduce biologically plausible global root patterns. Finally, we directly employed these control structures optimised in the context of biological systems to implement the exploratory behaviour of a swarm of robots. In this sense, our study goes beyond vague biological inspiration into direct application of the exploration strategy used by the biological system to the engineering application. The inherent risk of failure of such a direct transfer is theoretically mitigated by millions of years of evolution of the root exploration strategies; in practice, the results from this study confirm the applicability and efficiency of this algorithm.

Chapter 2

Soil Exploration & Exploitation by a Root System

As previously mentioned, the physiological union of the tips of the roots of a plant (apexes) serves the greater task of supporting and nurturing the plant (among others). All directional growth decisions and a majority of environmental sensing are made in the apex. Growth patterns of roots are basically influenced by gravity, genetics, soil conditions, and distribution of nutrients (H_2O, CO_2, minerals, etc.). Since there is no anatomic evidence for a central sensing and decision unit and considering the rather low computational capacity of a plant cell (compared to neuronal systems of animals, for example), it appears meaningful to consider the apex as a simple autonomous unit taking decisions on its own account [Baluška et al., 2004]. There is some evidence of communication between apexes [Davies and Zhang, 1991; Ali et al., 1999], but a higher, centralized brain has never been observed in plants.

Considering a plant as a swarm of individuals is not a new concept, as it was firstly described in 1800 by Erasmus Darwin [Darwin, 1800]. At that time, plant-philosophers discussed the individual 'minds' of plant apexes (mostly those of the sprout) and their power to turn into an entire plant when cut off and put into soil as joining a greater organism and functioning similar to a swarm of individual animals. In later discussions this swarm concept was dismissed as a philosophical concept but still the absence of a central master mind and the distribution of decision loci led to the formulation of meta-population to characterize plants [White, 1979].

Analyzing and subsequently simulating root growth has been in the focus of previous research [Pagès, 2002] (root analyses, e.g.: [Berntson, 1994; Coutts, 1983; Doussan et al., 1998; Ozier-Lafontaine et al., 1999; Lynch, 1995; Pregitzer et al., 2002]; growth and branching simulations e.g.: [Pagès et al., 1989; Lynch et al., 1997]). The major justification for these analyses derives from agricultural/physiological questions on root efficiency, soil exploitation, nutrient uptake per volume root, etc. The main means used are fractal methods, i.e., describing root architecture as a fractal. This work nicely models root architectures, also able to incorporate lack of nutrients, CO_2, water etc. However, this technique involves recursive formulation and hierarchical levels and although the simulations of roots match quite well the observed growth patterns in real plants, it does not reflect the decision processes actually going on during root growth.

Trying to infer basic operational principles from plants, and in this case root swarms, to design efficient exploration algorithms has the advantage that the exploration strategy's blueprint is imprinted on the root and thus directly observable. Contrary to other biological systems for which thousands of experimental trials have to be observed in order to deduce patterns in the exploration strategies, in the case of roots these strategies are available and at our disposal right from the start. Even if the social-insect metaphor is straightforward for implementation on engineering the design of efficient exploration algorithms, it suffers from an inherent disadvantage: the exploration of an unknown terrain (or volume) is done before the discovery of food sites, etc. and hence is very difficult to systematically observe.

2.1 The root system

The ability of the root system to perform its key roles, the capture of water and nutrients from the soil and providing anchorage for the shoot, is strongly dependent on its root architecture, i.e. the spatial distribution of roots within the soil. It is estimated that, globally, only 30-50% of the applied nitrogen fertilizer and ∼45% of the phosphorus fertilizer is taken up by crops [Tilman et al., 2002] with the losses contributing to greenhouse gas emissions and diffuse pollution of the aquatic ecosystems, as well as representing enormous economic wastage. Unfortunately, root traits are notoriously difficult to select for in breeding programmes but there is now considerable interest in the opportunities for improving crop root architecture through new approaches [Lynch, 2007]. However for this to be an achievable goal, it is crucial that we start with an understanding of the complexity of the processes that contribute to building a root system that is as complete as possible.

The ability to occupy space depends on several root characteristics, including relative growth rate, biomass, fine root density, and total surface area. A unique property of plants is their lifelong ability to grow and to continuously develop, elaborating on the basic body plan of the embryo. Therefore plants depend on the incessant activity of confined populations of stem cells located at opposite ends of the apical-basal body axis.

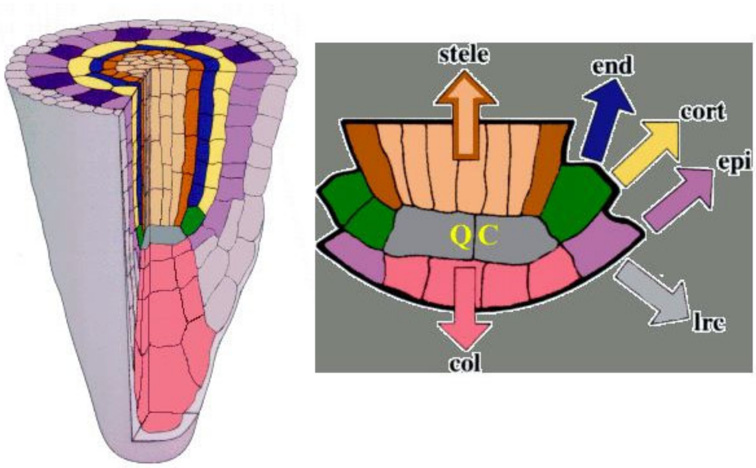


Figure 2.1: Schematic representation of the root structure

The first step towards root formation is the establishment of the apical-basal axis of the developing zygote [Waisel et al., 2002]. In the root of the herbaceous dycotiledonous model plant *Arabidopsis thaliana*, a small amount of stem cells generate all the different tissues that can be distinguished along the symmetrical radical axis. Because of their rigid cell walls, the stereotype division pattern of the root cells organize the separate tissues in concentric columns of cell files. From outside to inside these layers are designated by lateral root cap, epidermis, cortex, endodermis, and pericycle as the cell files that surround the central vascular tissue. Clonal analyses and ablation studies indicate that cell lineage does not necessarily determine the cell fate and pattern formation, but that plant cells are flexible and rather rely on positional information for adopting their final fate. At the basal end a set of stem cells gives rise to the central portion of the root cap, known as columella. Internal to, and contacting all the stem cells ,is a small number of mitotically less active cells, the Quiescent center (QC, Figure 2.1).

Along the apical-basis of the root, stem cells daughters continuously travel through time, crossing the zone of cell division (meristematic zone), the zone of cell expansion and elongation (elongation zone), and ultimately meet their destiny in the differentiation zone.

2.1.1 Root architecture

How is the root system architecture generated though the distribution and activity of root meristems? The term "meristematic wave" describes the propagation of root meristem activity through the soil [Dupuy et al., 2010a]. The meristematic wave may be a fundamental trait of plant root systems. The growth of an individual root occurs through cell division within the apical meristem and cell expansion close to the apical meristem. Newly created tissue eventually becomes rigid as cells differentiate. These developmental processes constrain the direction of advancing root growth and the permanent location of the root produced. Single meristems are pushed by this indeterminate self-organized process, and the advancing meristems of several roots combine to constitute the simplest form of a meristematic wave. The barley root system produces meristematic waves traveling through the soil. The models and simulations suggest that meristematic waves can be produced by the repetition of simple developmental rules in individual roots. Plant roots must acquire resources distributed heterogeneously in the soil volume. Plant species imply diverse root system architectures to explore space and optimize resource acquisition. Plants with highly tropic roots or with a smaller branching angle, explore space locally and can be more efficient in exploiting localized patches of resources in the soil. By contrast, the root systems of plants whose roots vary in their expansion rates (for example, through changes in diameter or in the branching order), or of plants that produce adventitious roots, proliferate more diffusely through the soil. Such spatial and temporal limitations to the generation of root system architectures are manifested in the contrasting strategies of plant species to intercept and exploit soil resources.

The region of highest root activity defines the envelope of the soil volume being exploited intensively by a root meristem. This volume is of fundamental importance to resource acquisition. The ability of the plant to sense the availability of water, nitrogen and phosphorus resides close to the root apical meristem, which allows the development of lateral roots into resource-rich patches [Hodge and Fitter, 2010]. The availability of mineral nutrients is usually higher at the root apex, and living root hairs, which contribute greatly to mineral uptake, are present predominantly at this region. The root apex is also the site of higher exudation of organic compounds [Badri and Vivanco, 2009], enzymes and mucilage, increased microbial activity and the aoplastic uptake of calcium and zinc.

Most architectural models predict root system development by simulating the incremental growth of independent tissues over time. In such models, the architecture of roots is explicit, and this allows complex analysis to be undertaken. For example graph theory concepts can be used to analyze uptake efficiency, and physical models can be developed to study local interactions between roosts and the soil [Dupuy et al., 2010a].

While explicit descriptions of root architectures are convenient to dissect biophysical, physiological and developmental processes, they have certain limitations. Architectural models require accurate measurements of single organ properties, which make them difficult objects to parameterize. The computational time required to generate the architecture of a plant root system depends on its size, and therefore applications have been restricted to a single plant level. Architectural models also generate complex structures and understanding their emergent properties can be difficult. Finally, architectural models produce a single and unique description of a root system for each simulation, which makes them difficult for studying the general properties on plant populations.

The root distribution models of [Hackett and Rose, 1972] and subsequently the reaction-diffusion models [Heinen et al., 2003], attempted to address these issues. However none of these models explicitly incorporated root developmental processes. Describing root architecture in terms of diffusion incorrectly assumes that root expansion rate is proportional to gradients in root length density, and neither the root length density branching density, expansion rate and branching rate generate root system architectures.

The success of models incorporating root architecture have also created a need for different types of models and approaches, and also for realistic parameter values for a range of crops. For example models assisting breeding strategies, and models of population dynamics involve simulation of large number of single plants simultaneously. This is still a huge computational challenge for architecture models because the functioning of all organs is computed explicitly. Effective model of plant soil interactions also requires coupling discrete structures, roots to continuous descriptions of the environments. Other forms of spatial models that could provide efficient solution of these problems, continuous-based approaches in particular, have not developed at a comparable rate.

2.1.2 Lateral root development

Lateral root development encompasses the formation of lateral roots from the cells of a parent root, and the regulation of the respective steps. Are the sites of lateral root founder cells predetermined, or are certain cells triggered by signals that occur during plant growth, or both? What determines the spacing of the lateral roots? What regulates the distribution of overall root mass in the soil? Each of the above questions must take into account plants' response to the environment, as we know that even genetically identical plants make very different developmental decisions when grown under different conditions. This developmental plasticity is a mark of plant development and is clearly seen in the regulation of lateral root formation [Malamy, 2005]. A role for environmental signaling in regulating lateral root formation makes intuitive sense, as this allows plants to optimize the placement of the roots in accordance with the complex and frequently changing soil environment. Hence to truly comprehend the lateral root formation it is necessary to understand the development and environmental cues that contribute to the regulation of this process and the way in which these cues are integrated.

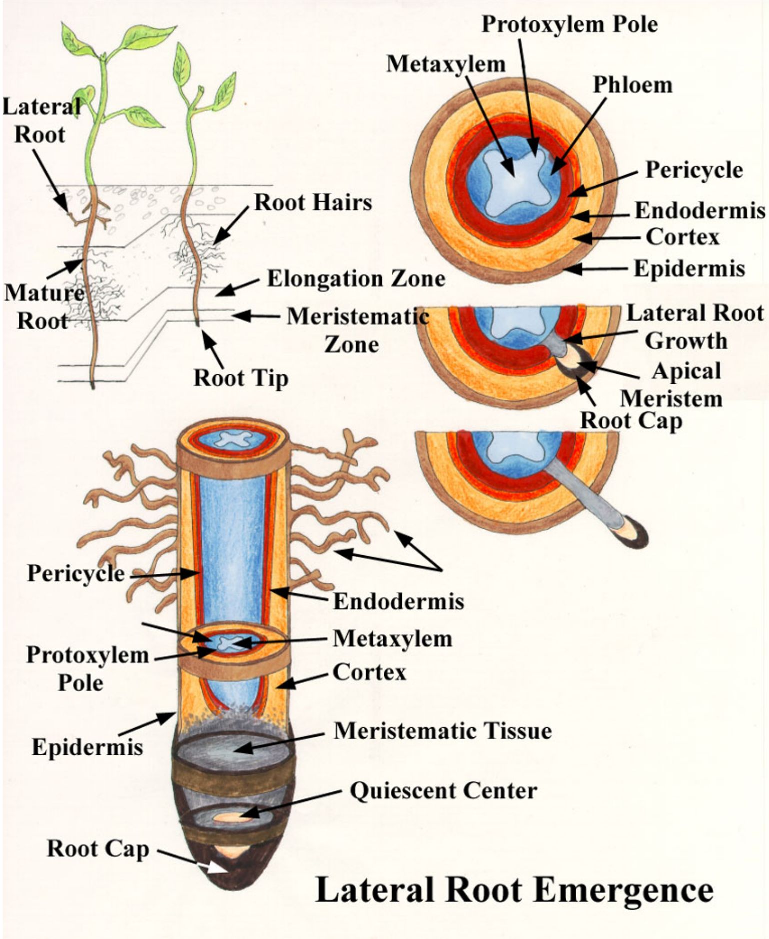
A model for developmental plasticity in the root system, or indeed in any other plastic organ, has been proposed by [Malamy, 2005]. First, developmental signaling pathways can be considered to be hard-wired into the plant. They determine the possible root system phenotypes for that plant genotype, and therefore should be consistent among genetically similar individuals. The existence of such pathways defines, for example the maximum size of branchness that can be attained by the root system of a given plant species. The hard wired pathways are referred to as intrinsic pathways. In contrast environmental pathways might influence lateral root formation by modulating components of the intrinsic pathways. In the model, these components act as nodes to integrate environmental signals with intrinsic developmental programmes and to coordinate root system morphology with growth conditions.

2.2 Effect of nutrient availability on root development

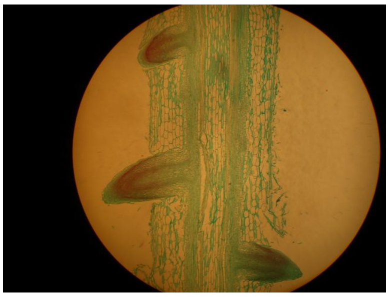
The degree to which root development is responsive to a wide range of environmental factors, including water and nutrient availability, is regarded as a primary example of the phenomenon of phenotypic plasticity [Malamy, 2005]. Phenotypic plasticity, defined as variation that is due to environmental effects, is conventionally considered to be the only nongenetic component of phenotypic variation.

Land plants grow in soil where water and mineral nutrients are heterogeneously distributed. Plant survival, growth and fecundity are largely conditioned by the ability to acquire these resources effectively. The architecture of a plant root system affects its ability to access these resources, and there is considerable evidence linking root architecture properties with the efficient acquisition of water and nutrients. However the fundamental mechanisms controlling root architecture and acclimation to the prevailing environmental conditions is complex and poorly understood.

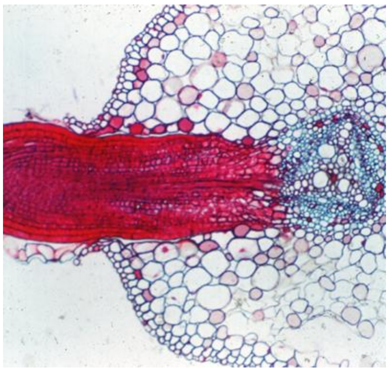
Root architecture results from the activity of apical meristems and is produced through a sequence of expansion and lateral initiation events at the proximity of root apices. Newly created roots are placed rigidly in the soil and the final form of the root system is a direct consequence of the patterns of root expansion and lateral root initiation in the proximity of root tips. Because mature roots are immobile, it is essential that meristematic activity is controlled and coordinated in conjunction with local soil properties. For example, the efficient acquisition of water and essential mineral nutrients requires the ability to detect resource-rich patches and concentrate growth within these patches [Hodge and Fitter, 2010]. How the plant controls the behavior of its meristems is therefore crucial for understanding the plasticity of root architecture. The detailed mechanisms by which water and the availability of the mineral nutrients are sensed by the plants remain poorly understood. However there is increasing evidence that the sensing



(a) Distinct parts of the root involved in the formation of lateral roots



(b) Longitudinal section of a main root with several lateral roots



(c) Detail of the lateral root formation with the origin in the pericycle

Figure 2.2: Lateral root development

mechanisms are located at the root apices [Dupuy et al., 2010a].

2.2.1 Root system and water

Evolutionary adaptation is an intrinsic response to environmental change. The environments in which plants grow have changed and are changing continuously. These conditions may even vary locally in very heterogenous soils. Under such unstable and unpredictable conditions plants have evolved and developed phenotypical plasticity as a valuable capability that allows adaptation to adverse physical and chemical environments they face.

Root system evolution is a notable process that has lead to a progressive transformation from the very simple root systems of early land plants to the diverse and complex root systems of the modern plants. Ancient plants did not face difficulties in getting water or nutrients let alone to efficiently anchor to the substrate as long as they kept growing in humid habitats. Modern day plant root systems have been modulated from distinct ancestors and by different evolutionary pathways influenced by environmental cues, leading to varied root systems architectures. For example the root system of most gymnosperms developed a tap root in which the embryonic root matures and becomes a primary root. From this primary or tap root, lateral roots emerge and the root system is formed. Under optimal environmental conditions, when nutrients and water are not limiting, the primary root grows continuously downward reaching deeper than the lateral roots. This growth pattern is called indeterminate and can be drastically altered by the scarcity of water and nutrients.

In contrast, the tap roots of monocotyledons have short life spans and the root system develops from post-embryonic stem-born roots that grow above or below ground, giving rise to branches that build a root system, called fibrous, in which no specific root grows longer than the others. Aerial roots stabilize the main stem and are also capable of branching and absorbing nutrients and water.

Increased root system size through increased lateral root formation and or growth extends the area explored by the root system. Increased density of shallow roots allows the plant to absorb surface water that is subject to rapid changes in availability through rain and evaporation, while the root system that penetrates to greater depths in the soil profile can only absorb “stationary” water. The correlations between root system traits and plant yield are highly dependent on the environment.

Although many published studies deal with manipulating water availability to plants, we still have a poor understanding of how and under what circumstances competition for water occurs. Paying greater attention to the mechanisms of competition for water and measuring the strength of belowground competitive interactions under different conditions of water availability should determine whether competition really does increase across spatial or temporal gradients in soil moisture and the extent to which the increase is explained by correlated changes in plant growth or biomass. It is important to separate the phenomenon of water availability from plant competition for water; that water limitations may be greatest in arid systems does not necessarily mean that competition for water is greatest there.

Deep roots may allow plants access to a water source available after upper soil layers dried out, enabling them to decouple the timing of growth from rainfall events, persisting after neighboring species have died or become dormant. Examples of temporal partitioning include early and late season annuals in the Mediterranean climate of California, shrub species of the Great Basin, and various trees [Dupuy et al., 2010b].

2.2.2 Root system and nitrogen

Nitrogen is one of the most important elements for plant mineral nutrition and is mainly present in soils in the organic form and as nitrate and ammonium. However, the availability of these compounds in the soil is poor because of the microbial community. Nitrogen limitation affects vital metabolic processes related to energy in plants, such as photosynthesis and respiration.

In order to deal with N deficiency plants establish symbiotic and non-symbiotic relations with soil microorganisms, such as bacteria and fungi. Another general response of plant root systems to N deficiency is the increase in root surface and size (total mass, length and area) and root depth, both important factors that allow the interception of nitrate leached from the soils.

The availability of N sources depends on ecological factors such as soil composition and environmental conditions. For example in soils under anaerobic and humid conditions and with low pH and temperature, nitrification is inhibited and soil ammonium concentrations increase [Miller et al., 2008]. Nevertheless, the concentration of nitrogen in the rhizosphere is heterogeneous in space and time [Cruz et al., 2007]. Nitrate is localized in randomly distributed patches. When a mature lateral root meets a nitrate rich patch, an increase in lateral root elongation occurs, suggesting a positive regulation of root meristem proliferation, whereas the rest of the root axis does not change dramatically. In contrast when plants are grown under a low, uniform nitrate supply, an increase in lateral root proliferation in the entire root axis is observed. Thus N availability regulates the root system architecture at least via two distinct pathways: one triggered by local sensing of the N status and one regulated by systemic processes.

The first steps to understand root developmental response to N availability have involved the study of lateral root formation at the tissue level. It is known that root systems responds not only to N supply but also to the molecular N source [Cruz et al., 2008]. Root density and extension of the root system of maize seedlings are larger in nutrient solutions containing ammonium than in those containing nitrate as the sole nitrogen source suggesting that cell division at the root apical meristem might be more rapid when ammonium is used as the nitrogen source, perhaps because ammonium assimilation is less energy demanding than nitrate [Domínguez-Valdivia et al., 2008]. The responses in root development in response to differential supplies of nitrogen has been proposed to be caused by the altered redox potential or pH during N uptake rather than by a direct influence of either nitrate or ammonium [Cruz et al., 2006].

Although future research in this area is needed, recent reports have contributed to the understanding of the complex and coordinated response of plant root system to N availability mainly in relation to the regulation of lateral root initiation, meristem activation and root elongation. Strong evidences point towards nitrate eliciting a systemic signal to regulate root growth.

In *Arabidopsis* and tobacco the growth rate of the primary root is almost insensitive to a uniform nitrate supply. However, relatively to shoot dry weight, primary root length slightly decreases as the uniform nitrate availability increases. In contrast in roots that encounter a high nitrate patch, primary root growth does not change. Therefore lateral roots might have a lateral nitrate sensory mechanism that enable them to modulate meristematic activity in response to localized sources of nitrate while the primary root tip might lack one or several components of this regulatory pathway. However effects of glutamate on root development, including lateral and primary root growth inhibition resulting in a short and highly branched root system [Filleur et al., 2005]. Root tip cells might be able to sense extra cellular glutamate that triggers a reduction in the rate of cell production and/or cell expansion and, therefore, promotes rapid colonization of the soil patches with high nutrient concentrations. Besides these advances, a detailed analysis of the developmental changes in primary root length as a consequence in modification of cell size, or cell number remains to be carried out.

In the case of maize, an increase in dry mater accumulation has been observed when one root axis has been supplied with nitrate in split root experiments. This dry matter accumulation was not attributable to growth of the primary axis but to increase of lateral root growth. Also, the maize primary seminal root showed a greater extension rate in relation with both nitrate and ammonium supplies, in contrast with the elongation rate in roots without nitrate, suggesting that the elongation is nitrogen dependent and that this nutrient is acquired from the growth medium and not from internal sources, because in this region of the root the phloem system is not fully developed to supply the necessary nitrogen from mature tissues. Furthermore, [Tian et al., 2005] reported a decrease in root length for primary seminal and crown roots when the N supply increased from 0.05 to 20 mM for two contrasting genotypes. A positive correlation between the quantity of nitrate applied and the internal cytokine concentration has also been observed, establishing a possible role of cytokine in the nitrate-mediated root growth found for the most responsive genotype.

Changes in lateral root growth in nitrate patches has been reported in several plant species, including *Arabidopsis* and maize, barley legumes, citrus, rice and tobacco [Hodge, 2004] and some components of the signaling and regulation pathways have been described. Through split root system experiments, nitrate riched patches and homogeneous nitrogen supplied to one axis of the root were both found to produce an increase in lateral root formation and elongation. In barley this has also been observed in response to ammonium. In *Arabidopsis* the response has been proposed to be specific to nitrate, because neither ammonium nor glutamine stimulates lateral root growth [Tranbarger et al., 2003]. Nitrate might be a key signal molecule capable of inducing changes in the developmental programme of plants when they are grown in heterogenous soils because of the relative mobility of this molecule when compared to the low mobility of ammonium or glutamate. This response might facilitate the uptake of nitrate from the soils where nitrate is produced from immobile organic compounds and/or ammonium as the result of bacterial activity and chemical reactions, directing lateral growth to these patches [Miller et al., 2008]. Furthermore, when *Arabidopsis* seedlings are grown in homogeneous nitrate conditions, particular responses depending on the level of nitrate limitations have been reported. Transfer of *Arabidopsis* seedlings from high to very low nitrate concentrations triggers a response in lateral root length, whereas transfer to medium nitrate increases lateral root initiation, reflecting the existence of levels of regulation related to homogeneous low nitrate availability. Regarding lateral primordia initiation, the histology of lateral root formation and pericycle cell differentiation has been studied. It is known that high nitrate patches increase lateral root density.

The role of nitrate as a signal molecule is supported by the response of tobacco- and Arabidopis-lines with very low nitrate reductase (NR) activity, and thus with limited capacity to metabolize nitrate. The higher accumulation of free nitrate in shoots of transgenic lines with low nitrate reductase activity growing under both low and high nitrate conditions inhibited lateral root growth similarly to that observed in wild-type plants growing under high nitrate conditions. This points to nitrate as a signal molecule capable of developmental responses in roots. Such a role in long distance signaling has been demonstrated in split root system experiments in which the accumulation of nitrate in the shoots, but not in the roots, triggered lateral root development initiation over all the root system. Therefore a dual system has been proposed for controlling the developmental changes in nitrate availability: one locally induces lateral root elongation by high nitrate patches and one involves a systemic signaling that mediates repression of the meristematic activation of lateral roots that depends on the nitrate levels in the shoots. To date the regulation of lateral root development by internal and external nitrate levels has been reported at three stages: lateral root primordia initiation, emergence of meristem activation, and lateral root elongation.

[Malamy and Ryan, 2001] reported that when *Arabidopsis* seedlings are growing on media with a high sucrose to nitrogen ration, lateral root initiation is repressed, thus implying a possible role of sugars or sugar-n balance in root responses to nitrate availability.

A possible role for ABA in nitrogen-mediated root formation and growth has been proposed. Lateral root development is less repressed by high nitrate in the ABA-sensitive *Arabidopsis* mutants abi4 and abi5. This de-repression occurs when plants are grown in concentrations under 1 mM nitrate, whereas under low concentrations, the phenotype of both mutants and wild-type is similar. These results also relate nitrate responses with sugar signaling because ABI4 seems to be involved in sugar signaling.

Also a role of the hormone auxin in lateral root development in function of nitrate availability has been demonstrated by showing that AtNRT1.1 possesses a functional auxin response element and is transcriptionally induced by exogenous auxin. Also the axr4 mutant of *Arabidopsis* does not respond to a localized supply of nitrate, suggesting a role for auxin in lateral root development in response to the nitrate supply [Zhang et al., 1999]. Additionally an increase in IAA levels has been observed when roots of *Arabidopsis* seedlings are transferred from high nitrate to low nitrate concentrations when compared with those maintained in high levels implying that auxin is necessary at some checkpoint for lateral root elongation [Walch-Liu et al., 2006]. However these results are in contrast to those obtained by [Linkohr et al., 2002] who observed that axr4 seedlings respond to a localized nitrate supply in the same way as the wild-type.

The sensing of external nitrate concentrations, has for example been found to be regulated by the ANR1 gene, which is expressed in the root tip [Forde and Walch-Liu, 2009]. Similarly a series of experiments by [Svistoonoff et al., 2007] has demonstrated that physical contact of the *Arabidopsis* root tip with phosphate is necessary and sufficient to are the growth of primary roots, and that low phosphate root (LPR) genes are involved in this response.

2.2.3 Root system and phosphorus

Phosphorus has a fundamental role in most developmental and metabolic processes in plants. It is not only a constituent of key cell molecules, but it is also an essential metabolic regulator of several processes (protein activation, energy transfer and carbon and nitrogen metabolism). Moreover P availability represents one of the major constrains for growth and development of terrestrial plants in both natural and agricultural ecosystems due to its low mobility and high absorption capacity by the soil. The environmental challenge imposed by P availability in the soil was a major selective pressure for plants to evolve a range of developmental, biochemical and symbiotic strategies to adapt to P deprivation. In both mono and dicotyledonous plants, a general strategy to cope with low P availability has been described [Wissuwa et al., 2005] that involves three fundamental mechanisms:

1. Release and uptake of phosphate from organic and inorganic sources;
2. Optimization of phosphate utilization by a wide range of metabolic alterations and mobilization of internal P;
3. Increase of the root's exploratory capacity through an increased root area.

The first mechanisms involve biochemical responses directed to augment soil phosphate availability by increasing P uptake capacity through the induction of high affinity phosphate transporters, P recycling and P mobilization through the synthesis and excretion of phosphatases, RNAses and excretion of organic acids. The second mechanism involves the utilization of alternative glycolytic pathways that involve phosphate-independent enzymes, changes in carbohydrate metabolism, and the hydrolises of phospholipids to release phosphate for other metabolic processes and their replacement for non-phospholipids such as sulfolipds and galactolipids. Recently microarray analysis in *Arabidopsis* confirmed that genes involved in several of the previous processes are transcriptionally up-regulated by phosphate starvation [Amor et al., 2009]. From 732 differentially regulated genes, 501 are up-regulated and 231 are down-regulated. Genes involved in the lipidic hydrolysis and galacto- and sulfolipid synthesis are up-regulated by phosphate deprivation.

To face phosphate stress, many plants adapt their root system developmental programme towards the formation of shallow and highly branched root system that increases the soil exploratory capacity of the plants. Phosphate dependent root architecture alterations have been studied in diverse crop systems and despite discrepancies most root systems experience an increase in adventitious root and lateral root density under P limiting conditions [Eissenstat et al., 2000]. Under natural conditions these alterations are though to be directed to maximize P acquisition as it becomes limited in the soil.

2.3 Root behaviours

2.3.1 Apex–soil interaction

Modelling fish or bird swarms only mildly requires considerations of the properties of the substrates the animals move in (like fluid density and sensory range). In running insects, the only boundary conditions are that the animals can only locomote on rigid surfaces but not freely within space; stigmergic olfactory cues are always located on the ground, visual senses are not restricted by blurring effects. What might sound trivial turns into a huge problem under ground: The 3D space shows different levels of accessibilities due to soil compactness, sensors can not reach very far as a result of the high density of the medium and signals might be diffusing inhomogenously. For example, we know that gravity is a downward driving force for birds but is almost perfectly balanced for fish and irrelevant for ants. Roots however, use gravity as a primary navigational cue as they emerge from the air-soil interface and head into the soil. The soil, however, is a much more complex medium than those mentioned before and hence requires full attention. Soils are not homogeneous, displaying increased compactness with increased depth, but also contain patches or layers of different mechanical properties. Different soils and soil-states additionally exhibit different capacity for nutrients and water. Without an appropriate representation of the soil, it will be impossible to model and understand root growth. Understanding the mechanical properties of root growth are key to their reaction to soil mechanics. A rather recent approach to the modelling of soil biophysics was undertaken by Pierret and co-workers [Pierret et al., 2007]. Finally, each root system must maintain a static equilibrium balancing the weight of the plant and additional dynamic loads the plant is exposed to. The area may have a certain topology which makes access to certain places easier than to others. The fine structure of the substrate may allow easy locomotion in one place but inhibit it in another one - despite similar topology. Similar, but situated at the extreme, are obstacles which are not accessible to the plant at all, as outlined in the point below. Taking our goal of controlling e.g. rover motion we see great similarities in the way that also a rover locomoting on a surface may encounter variations in the ease with which a certain type of substrate can be driven on and hence need to be taken into account and plan a detour.

2.3.2 Negotiating obstacles

Obstacles are an extreme of compact soil patches. In the worst case, rocks display a region of in-accessibility to the apex which needs to be circumvented to continue growth and soil exploitation. Coping with obstacles is hence critical for the overall performance. The effect has been known for over a hundred years but the mechanisms were only briefly described. Falik and co-workers [Falik et al., 2005] demonstrated that roots of *Pisum sativum* detect and avoid obstacles by a mechanisms of self inhibition. Allelopathic exudates accumulate in the vicinity of obstacles (as their diffusion is obstructed) and growth toward the obstacle is in consequence inhibited. Pierret and co-workers [Pierret et al., 2007] have implemented a similar mechanism in their simulation mentioned above.

2.3.3 Exploiting nutrient patches

Finding nutrients is a major task of a root and important for the success of a plant, exploiting scientifically interesting sites is the major task of any exploratory mission. Every mission tries to answer more than just one scientific question, but the sites relevant to find answers might be different. The root's strategy to find and exploit nutrients in a balanced way is crucial to its overall performance.

Accessing the nutrients distributed within the soil is actually one of the major tasks a root has. Indeed, nutrients do not behave homogeneously and make foraging a challenge. Basic nutrients are Nitrogen and Phosphate, but also the availability of Water and Oxygen play a role. Some of these are volatile and get distributed and subsequently washed out by water, others are rather equally distributed and do not follow water dynamics. Other nutrients appear in patches or layers and need to be precisely located [López-Bucio et al., 2003]. In summary, each component requires its own foraging strategy. Patches with increased nutrient concentration trigger roots to proliferate compared to roots of the same plant outside that zone [Hodge, 2006, 2004]. A similar behaviour is observed with toxic or poisonous substances. Although they chemically belong to the group of nutrients, toxic substances trigger similar avoidance behaviors as solid obstacles do, as is the case with, e.g., Aluminum [Miyasaka and Hawes, 2001].

2.3.4 Self/non-self recognition

A root system of a plant needs to coordinate the growth of the individual apexes - when and where to proliferate, how to balance growth between apexes, which apex to 'send' into a certain direction. All these decisions require that a root is capable of recognizing the presence of other roots around it. These roots may be of the same plant or of foreign plants, these plants again may be of a close kin or a potential aggressor. Self recognition of roots is mediated internally, i.e., via the direct connection of the roots upstream. As soon as this connection is lost, the recognition as self fails [Biedrzycki et al., 2010]. Recognition of foreign roots happens through exudates, messenger molecules, in the soil. Young seedlings of *Aradbidopsis thaliana* were confronted with exudates from (i) siblings, (ii) strangers, and (iii) own exudates. Roots encountering the stranger's exudates showed increased formation of lateral roots in comparison to those encountering sibling's exudates [Biedrzycki et al., 2010]. These exudates are actively secreted and can be deactivated by inhibitor substances. The reaction of roots when interacting with self, kin or strangers is seen as a strategy to increase individual fitness by maximizing resource exploitation in a competitive environment despite the fact that reactions differ between species [Callaway and Mahall, 2007; Lynch, 1995]. Do robots of a collective mission seek each other to support each other, or shall they exhibit avoidance behaviour and literally not disturb their colleagues and re-screen a previously analysed area? These questions can only be answered by the type of mission that is intended and similar is the plants' behavioural repertoire. We will need to analyse reasons for different behaviours of apexes in sight of their kin.

2.3.5 Root–shoot interaction

Root and shot equally form the entity of the plant, standing in direct and constant bi-directional interaction. Downstream, the major signalling happens in form of the availability of photosynthetic products, high energetic carbon-hydrates which are used for growth or stored. The upstream signalling is more complex and also of higher relevance for our modelling approach. Root signals in the form of signalling molecules move to the shoot within the transpiration stream ([Bacon et al., 2002], but also slow-travelling electric signalling in plants is discussed). Although the transpiration stream moves rather fast, it can take days to transmit a signal from the root of a tall tree to the tips of the shoot. The strength of any stream-related signal depends on the concentration of the signalling substance within the liquid but apparently, daily variation of transpiration rate can be compensated for [Freundl et al., 1998] and the signal kept constant. Common signal molecules are abscisic acid (ABA) and ethylene (e.g. [Dodd and Davies, 1996]). For example, ethylene concentration rises in the presence of root stress and possibly regulates plant growth in drying soil [Spollen et al., 2000; Hussain et al., 1999]. The analogy may be found in the interaction of a swarm of robots with its mother-base, which is actually waiting for the products, i.e. the data collected. Here this might be realized in a form of a relay station to earth, which also supplies the rovers with energy for a new excursion.

Chapter 3

Cellular Automata modeling of Soil & Root dynamics

To proceed with a study of root dynamics, we created a computer model that would allow us to experiment with potential apex behaviours, and observe their performance in exploring for and exploiting the resources available in different soils. The approach followed in the creation of the model was that of Cellular Automata (CA) modeling. In Cellular Automata, a **universe** of **cells** is simulated, where all cells are at any given instant in one of a finite number of **states**. **Update rules** specify how to update cell states, at **discrete time steps**, as a function of their present state, and those of their **neighbouring** cells. Table 3.1 presents how these components traditionally found in CA [Wolfram, 2002] translate into our model of soil and root dynamics.

3.1 The Soil

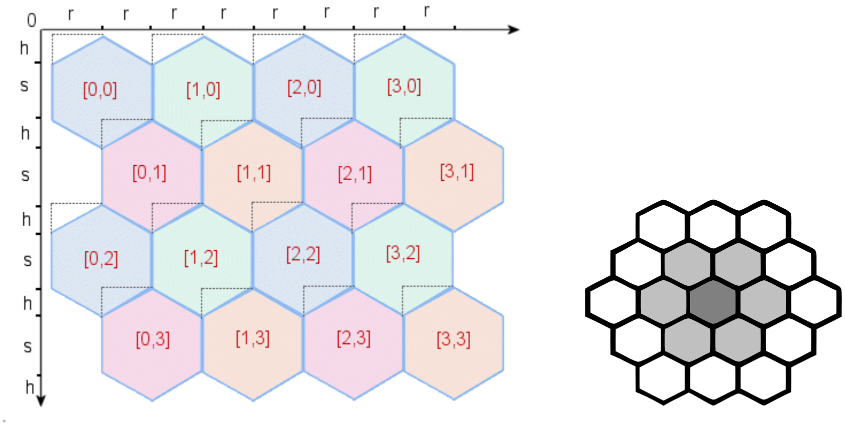
A root is a complex structure that cannot be fully understood in isolation from the soil in which it grows. It is therefore vital to include in the simulator a representation of the soil and a modeling of its dynamics. This section focuses on the description of how the soil is represented in our model. Section 3.1.1 describes how the soil is structured, and Section 3.1.2 lists the features that were chosen to characterize it. The process used to randomly generate initial soil configurations is presented in Section 3.1.3. Finally, Section 3.1.4 explains how to interpret our visualizations of soil configurations.

3.1.1 Structure

A volume of soil is represented in our simulation by a two-dimensional lattice. This lattice represents a depth-cut, with the surface level being at the top, and successive rows increasing in soil depth. The soil is discretized into

Soil	Root	coupling of Soil & Root
Universe		
2D hexagonal lattice	binary tree	the soil's hexagonal lattice with the root's binary tree overlayed on top
Cell		
soil patch (an hexagon)	root segment (a tree node) *apex* – a *leaf* node in the tree *shoot segment* – the tree's *root* node	soil patches, and root segments
Cell Neighbourhood		
a soil patch has as neighbours the 6 soil patches immediately adjacent to it	a root segment always has as neighbour the segment from which it grew (its *parent* node in the tree), plus up to two other root segments that grew out of it, if any (its *child* nodes in the tree)	*soil patches*: acquire as additional neighbours all the root segments that grow into it; *root segments*: acquire as additional neighbours the soil patch in which they are contained, plus its surrounding soil patches
Cell State		
amounts of Water, Nitrogen & Phosphorus		
Cell Update Rule		
no updating of soil patch states (soil→soil diffusion) was implemented in this project	*(b)* root segments diffuse internally materials to neighbouring root segments (root→root diffusion)	*(a)* apices potentially grow new root segments, thus changing the root's Universe; *(c)* root segments extract materials from the soil (soil→root diffusion)

Table 3.1: Correspondence between standard Cellular Automata elements and elements of the modeled entities



(a) Two dimensional hexagonal lattice representing the soil. The surface is above row 0, and successive rows increase in depth. The shown coordinates index soil patches' locations in a regular matrix. The hexagonal lattice is thus implemented by considering an horizontal displacement on odd numbered rows [Jahn, 2002].

(b) Definition of a soil patch's neighbourhood. The neighbours to the dark grey soil patch are shown in light grey.

Figure 3.1: Soil structure

individual units that are from hereon designated as **soil patches**. A soil patch will have a hexagonal shape, and will be assumed to represent some volume of soil (see Figure 3.1a). The neighbours to a soil patch are the six patches immediately adjacent to it (see Figure 3.1b).

The real-world dimensions of a soil patch should be such that an apex of the modeled plant, from its location inside a soil patch should be able to perceive the overall conditions in all the neighbouring soil patches. This capacity is necessary for root apices to be able to evaluate their context and take the appropriate growth decisions.

The reasons for implementing soil as an hexagonal lattice, rather than with the more straightforward square lattices, are due to their isometric properties, and their reduced number of neighbours. While the standard Moore neighbourhood for square lattices considers eight neighbours, not all of which lie at the same distance from the centre, an hexagonal lattice allows us to consider only six equidistant neighbours. This simplifies the modeling of soil dynamics, and also reduces the complexity of root apex controllers (see Section 4.1).

In terms of implementation, the soil is still represented as a square array, but as one where a horizontal displacement is considered to be present on odd numbered rows (see Figure 3.1a for an illustration). A process is implemented as described in [Luczak and Rosenfeld, 1976] for converting between the array's Cartesian and hexagonal coordinate systems, and using

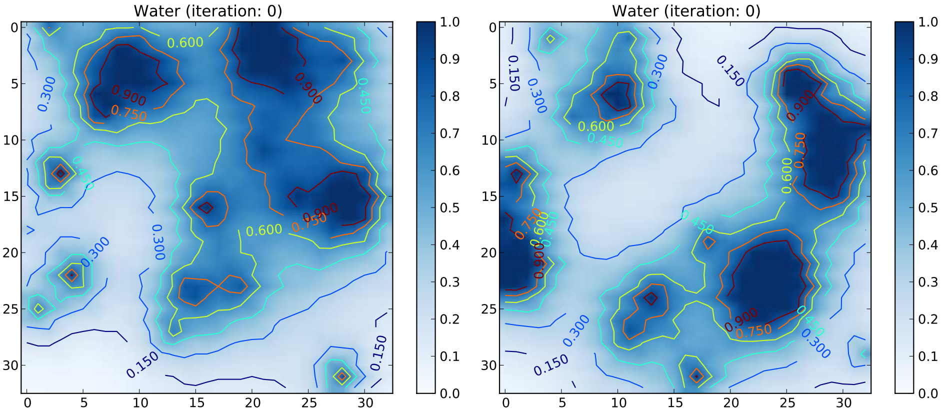


Figure 3.2: Contour plots of two randomly defined initial soil configurations (only the ‘Water’ state variable is shown here for each soil patch). Configurations obtained by randomly scattering 32 *peaks* in a soil of 33 by 33 soil patches.

that information neighbourhoods can then be calculated [Pazera, 1999].

3.1.2 State variables

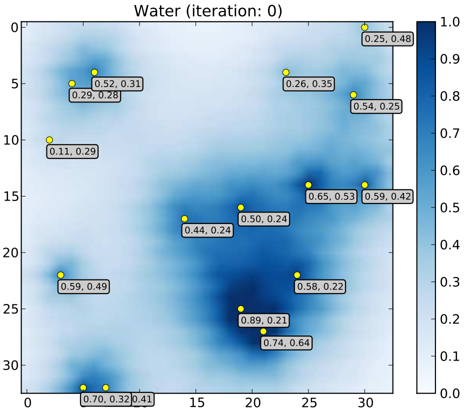
Of the many soil features available for modeling, we restrict our attention to three: Water, Nitrogen and Phosphorus. They were chosen for their adequacy as representatives of the environmental factors that influence root growth (Section 2.2).

A soil patch represents some volume of soil. Its state is represented by a set of continuous variables that characterize the total amounts of Water, Nitrogen and Phosphorus available in that soil volume.

3.1.3 Generator of random soil configurations

Plants capable of thriving in a wider variety of conditions have a selective advantage over competitors for the same ecological niches that do not display the same level of robustness. A plant with a greater phenotypic plasticity will have a greater number of environments at its disposal, and will be more successful in poor soils that prevent plants following more rigid growth behaviours from prospering. A successful model of root growth should therefore be capable of exposing roots to a great variety of conditions [Rajaniemi and Reynolds, 2004].

A procedure was developed for randomly defining soil patches’ initial states, with values that are plausible from a biological perspective. This procedure follows the design principles of having no strong discontinuities



'Soil/Water/Peak/coords' : [(19, 16), (23, 4), (7, 32), (2, 10), (19, 25), (25, 14), (30, 14), (21, 27), (4, 5), (30, 0), (29, 6), (14, 17), (24, 22), (6, 4), (5, 32), (3, 22)]

'Soil/Water/Peak/height' : [0.50, 0.26, 0.27, 0.11, 0.89, 0.65, 0.59, 0.74, 0.29, 0.25, 0.54, 0.44, 0.58, 0.52, 0.70, 0.59]

'Soil/Water/Peak/decay_rate' : [0.24, 0.35, 0.41, 0.29, 0.21, 0.53, 0.42, 0.64, 0.28, 0.48, 0.25, 0.24, 0.22, 0.31, 0.32, 0.49]

Figure 3.3: A randomly initialized soil configuration, and the listing of all the numerical values needed to fully define it. 16 *peaks* can be seen, randomly scattered in a soil of 33 by 33 soil patches, having *heights* randomly chosen in the 'Water' variable's range, $[0, 1]$, and *decay rates* uniformly chosen in the range $[0.20, 0.65]$.

in the values of contiguous soil patches, and of having multiple areas of strong or weak concentrations present throughout the soil (Figure 3.2). An added advantage it provides lies in its capacity to generate, through parameterization, soil configurations that pose challenges of variable degrees of complexity to the root. When running a root growth simulation, this procedure is used to set the initial states of all soil patches, in all variables (Water, Nitrogen and Phosphorus).

The soil's initial configuration in each of its variables is considered to be analogous to a mountain range, with the height at any given location being indicative of the variable's local concentration. The soil is initialized by randomly defining a set of peaks, specifically in terms of where they are located in the soil, their height and their steepness (Figure 3.3).

The individual contribution each peak gives to a soil patch's initial value, its height, is given by the exponential decay function

$$h(d) = h_0 e^{-\lambda d}.$$

Here $h(d)$ is the height of a soil patch at a distance d from the peak, λ is the peak's decay constant, and $h_0 = h(0)$ is the initial height, i.e. the height at distance $d = 0$. A soil patch's initial state, in each of its variables, is given by $\sum h(d)$ for all the variable's peaks. Each state variable can only vary in a given range, so the result of the sum must be bound to it. Note how the total height at a soil patch containing one of the peaks will most likely be higher than the peak's height, as all other peaks potentially contribute also to increasing the soil patch's height.

The distance d between two soil patches in the hexagonal lattice is mea-

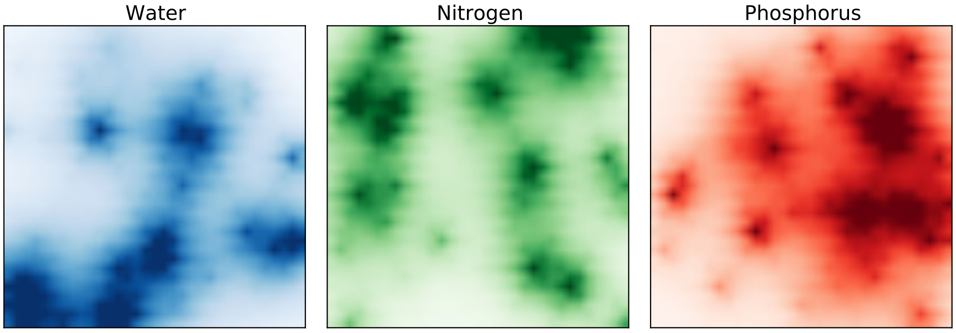


Figure 3.4: Standard visualization of a soil configuration at a given instant. One soil of 33 by 33 soil patches is shown. Each soil state variable ranges in $[0, 1]$, and is randomly initialized through the definition of 32 *peaks*. Darker areas indicate higher concentrations.

sured as specified in [Luczak and Rosenfeld, 1976, measure d_6] (see [Mehnert and Jackway, 1999, Figure 2b] for a comparison with different distance measures). This distance measure produces the number of hexagons in the shortest path between two hexagons. It is equivalent to the Manhattan and Chessboard (Chebyshev) distances on square lattices.

3.1.4 Standard Soil visualization

Figure 3.4 illustrates what from hereon in the report will be the standard visualization of a soil configuration at a given instant. As previously mentioned, each soil patch's state is defined by the amounts of Water, Nitrogen and Phosphorus it contains. Figure 3.4 shows three panels, one per soil state variable. Each soil patch is therefore shown three times.

The soil configuration depicted in Figure 3.4 shows a scenario where the soil is very dry close to the surface (top of the image). The place where a root would be able to extract most Water from, the left side of the deepest soil layers, has however very poor concentrations of Nitrogen and Phosphorus. In a similar way, the top left section of the soil has a high concentration of Nitrogen, but poor concentrations of Water and Phosphorus. As roots need to secure access to all of them, they would have to develop in a way that would ideally grow segments into these diverse patches of greater concentrations, while dealing with their local deprivations on the remaining materials.

Note that this visualization treats the soil as a regular matrix, of square shaped patches. The horizontal displacement on odd numbered rows of soil patches mentioned in 3.1.1 is therefore not taken into account when the image is generated. The visualization is not therefore a perfect depiction of the soil's actual structure. Contrast Figure 3.4 with Figures 5.2–5.5, where soil patches' hexagonal structure is accurately replicated.

3.2 The Root

Our interest in modeling root growth arises primarily from our interest in replicating roots' robust soil exploration and exploitation behaviours. We therefore modeled biological processes at a minimal level of realism required for such behaviours to be replicable. Root functions such as that of providing anchorage to the plant could then be ignored, but others such as internal transport of extracted materials, and energetic costs to root operations could not [Fitter, 2002].

Our modeling of root growth considered as a prototypical root that of a herbaceous plant (see Figure 3.5). Such roots grow segments that are very homogeneous in their composition and function, undergoing few transformations of major significance over time. Additionally, such roots display a significant degree of phenotypic plasticity in their foraging for resources. These attributes are of relevance for the technological transfer of the root growth behaviour considered later on.

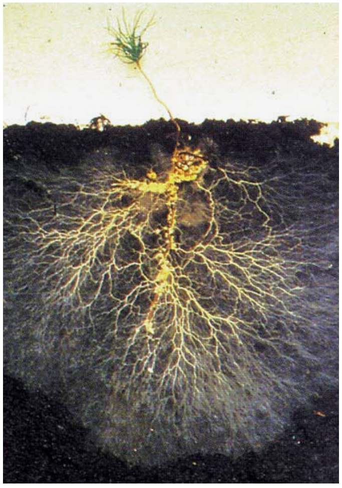


Figure 3.5: Example of a Herbaceous plant

The present section describes the root architecture's representation. The following sections will then describe the model's dynamics. Besides of the considerations drawn in this section, arising from the model root, other sections draw implications in terms of sizing of soil patches (Section 3.1.1) and consequent parameterization of the dynamics (Section 3.3.3) and their timings (Section 3.3.1).

3.2.1 Structure

A model of root growth must necessarily represent the root system through a mutable data structure that is extensible, so as to encompass newly grown segments. In the case of our model plant, the different segments throughout the root are very similar between themselves in terms of shape and function. This regularity is observable not only in space, but also in time: as the root grows, the root segments already in place do not undergo any major transformations (an example of such transformations would be the gradual thickening of a main root stem for providing anchorage to the growing plant). An abstract model of such a root can therefore rely on a composition of homogeneous elements, that undergoes transformation simply through the gradual addition of new elements at its extremities.

The complete root is discretized into uniformly sized units designated as **root segments**. Each root segment is embedded in a single soil patch, from which it extracts water and nutrients (note though that multiple root segments may be present in the same soil patch). A notion of neighbourhood between root segments is introduced: two root segments are said to be neighbours if they are contiguous in the root structure, and therefore occupy contiguous soil patches.

The complete root is represented as a graph, having root segments as its nodes, and neighbourhood between root segments represented by edges (and not the other way around, as one might also consider). Specifically, the root is represented as a connected acyclic graph, where nodes have a degree between 1 and 3 (one parent and 0,1, or two children). In other words, the root is represented as a *binary tree* [Cormen et al., 2009]. Other than the binary tree's *root node*, each root segment has a neighbour, designated as its *parent*, which is the root segment from which it grew. Each root segment has in addition from 0 to 2 *children* nodes, corresponding to root segments that grew from it. The degree of branching by an apex was limited to 2 so as to limit the complexity of apex controllers (Section 4.1).

The binary tree's root node is designated as the *shoot segment*, and is the only entity in our model that exists above the surface level. In our model, the root is the only explicitly modeled part of the plant. The single shoot segment therefore lies at the interface between the modeled and non-modeled parts of the plant. It is at this entity that interactions between the root and shoot should be taken into account in the model. The *shoot segment* has a degree of 1, as it has no parent node, and is connected to a single root segment.

Root extremities are represented as root segments having no children nodes (the so called *leaves* of the binary tree). These are the root's *apices.* When an apex grows, it adds to the binary tree either one (*elongation* event) or two (*branching* event) new segments, which become embedded into soil patches adjacent to that of the apex. The old root segment ceases to be an apex, and that designation then falls into the newly created root segment(s).

Root growth occurs in our model only at the apices. The model does not therefore support cases where a mid-section of a segment originates a new apex. This is motivated by properties of our technological application (see Chapter 5), and by considerations of computational efficiency, as apices are then the only autonomous units that need to perceive and act on their environment. Being apices the only root segments involved in root growth, the remaining root segments become mere spectators in the root growth process, and then carry out a single function: that of contributing to the transport of extracted materials through the root system.

In Appendix A we discuss implementation details on the representation and efficient processing of segments in a root structure.

3.2.2 State variables

Each root segment contains inside itself variable amounts of the same three continuous variables that are also present in soil patches' state: Water, Nitrogen and Phosphorus. These values change over time by extraction from the soil, and through internal diffusion in the root system.

3.2.3 The Seed

The root's initial configuration is defined as a binary tree containing only two contiguous segments. As explained before, the binary tree's root node is the *shoot segment*. Connected to it is a single root segment. Having no other root segments following from it, this single root segment is thus considered to be a root apex.

A simulation starts with the single apex in the plant's *seed* placed in the soil patch at the centre of the first row of soil patches (the surface layer), with the shoot segment protruding from it above the surface.

The amounts of Water, Nitrogen and Phosphorus inside the two seed segments at the beginning of the simulation are system parameters.

3.3 State Updating

The dynamics implemented in our model are succinctly described as processes of diffusion from sources to sinks. Much of the dynamics in real soils and roots consist in flows from entities where there is a surplus of some material, into neighbouring entities where there's a deficit of it. Undisturbed, diffusion might then over time lead to an equalization in the amounts of the material present in these entities. Disturbances are introduced into this natural flow by the autonomous activity of apices, which cause the appearance of new entities to and from which materials will flow.

In our implementation, a generic diffusion process (Section 3.3.2) is subjected to distinct parameterizations, which configure it to perform the different diffusions from/to soil patches and root segments (Section 3.3.3). As in other parts of the model, biological realism is here traded-off for simplicity and transferability into the robotics application.

Section 3.3.1 describes how the notion of time is handled in the simulation. Following the description of how diffusion implements the system's dynamics, Section 3.3.4 then describes how boundary conditions are taken into account in the model.

3.3.1 Updating Schemes

Time is discrete in our simulation. At every time step, all modeled entities (soil patches and root segments) update their state *synchronously*. In our implementation each simulated entity has its own individual representations of state at the current and following time steps. The multiple update functions (diffusion processes) are executed by each entity to set state variables' values at the next time step. Once all modeled entities have been updated, state variables are swapped, so that the following time step becomes the current time step. The plausibility of synchronous update schemes for modeling biological processes is supported, for instance, by [Gershenson, 2004], but it is known that asynchronous update schemes provide several advantages [Grilo and Correia, 2011]. Though synchronous updating is implemented at this stage, support for alternate updating schemes can in the future be easily incorporated into the model.

A transition in the model from one time step to the next is made to correspond to an interval of time in the real-world. This correspondence will determine the rates and frequencies at which distinct processes take place in the simulation. The model's discrete nature means it will present snapshots of how the biological system would look like at specific points in time, with consecutive time steps representing the initial and final states of the transformations that in the real-world would have undergone continuously in a time interval of that magnitude.

Root growth is a process that occurs much more slowly than the diffusion of materials throughout the soil and root. This leads to the requirement of having different update rules being applied in the model with distinct frequencies.

As mentioned in Section 3.1.1, soil patches' real-world dimensions should be such that an apex of the modeled plant, from its location inside a soil patch should be able to perceive the overall conditions in all the neighbouring soil patches. As in our model all root segments are uniform in size, when an apex grows, it grows by the length necessary to place the new segment in a soil patch adjacent to that of the apex. The amount of real-world time required for this process to complete is then the amount of time it takes for an apex of the model plant to grow by such a length.

Diffusion, which takes place at every time step in the simulation, occurs on soil patches and root segments having real-world dimensions constrained by the considerations mentioned above. Those dimensions have a bearing on the rate at which materials diffuse, but also on the number of time steps that should take place in the simulation in between those time steps on which apices take growth decisions. A diffusion process involving two contiguous root segments affects their mutual states. After n time steps, the materials exchanged in a diffusion event might potentially reach another root segment n segments away from those involved in that initial diffusion. Given we know the time span between apex updates, we can determine how far a signal might travel in a root of the model plant during that time. Dividing that distance by root segments' length, we then determine how many time steps to simulate in between apex updates.

3.3.2 The Diffusion process

The diffusion process controls the flow of materials over time between entities in the simulation. It is responsible for the entirety of the model's dynamics, with the sole notable exception of apices' growth behaviours, which are discussed in Chapter 4.

Diffusion is here implemented at a high level of abstraction, that raises the process' scope beyond the implementation of soil and root dynamics. On seeking such generality, many important details were necessarily lost, related to the corresponding processes going on in the physical world, and many decisions on implementation details had to be taken. The implemented algorithm is thus resulting from a delicate trade-off between the model's fidelity to a real diffusion process and the possibility of a simple transition to the robotics application. The diffusion process' full algorithmic definition is presented in Appendix B. In it, we spend some extra time thoroughly describing our exact choices, so as to specify all that is needed to successfully reproduce our results.

We next introduce the modeling requirements which guided the diffusion process' design, and follow with an exemplification of the dynamics that result from the defined process.

Design principles

Diffusion was defined so as to include support for a number of features that were considered necessary to be present in the model. These were:

- Diffusion should be a fully decentralized and localized process between the involved entities. No master resource-allocator allowed;

- From the model's perspective, all flows should be taking place concurrently. A degree of independence between different diffusions should therefore be achieved, so their execution order would not affect the outcome;

- The simulation is not a closed system. Diffusion should therefore be able to provide an interface with those parts of the world which are not explicitly modeled (the plant above surface level, the surrounding soil, . . .), so flows in and out of the system might be supported;

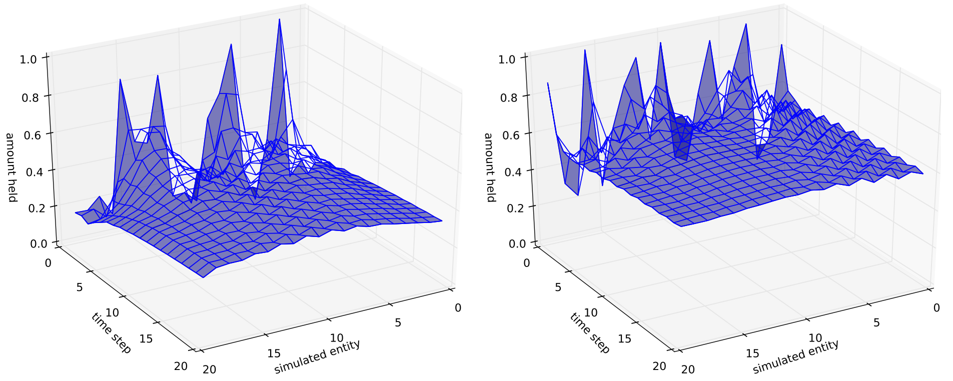


Figure 3.6: Time evolution of 20 simulated entities' state variables being subjected to diffusion, starting from randomly defined initial amounts. Entities structured according to a ring topology, where each diffuses into its two immediately adjacent neighbours. Diffusion parameterized with $d_k = 0.0$, $d_{o_{max}} = r_{i_{max}} = 0.25$, $r_c = 1.0$.

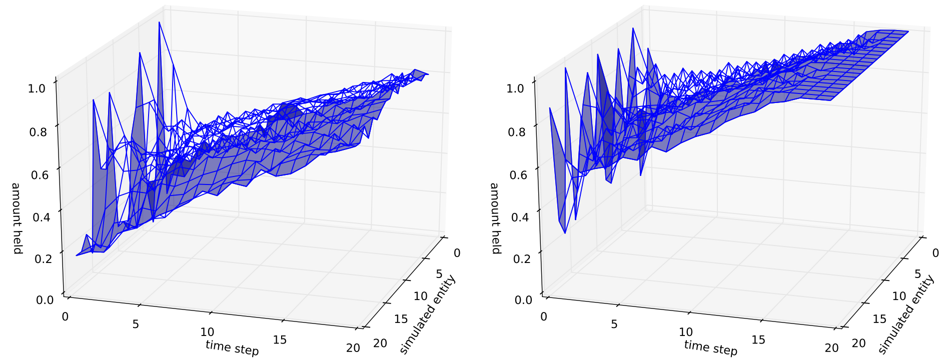


Figure 3.7: Replication of the time evolutions shown in Figure 3.6, but having at each time step entities receiving from outside an amount generated at random in the range $[0.0, 0.05]$.

- Diffusion should support the simultaneous consideration of an undetermined number of entities, potentially having distinct characteristics with an influence on how diffusion takes place.

Examples

Figures 3.6 and 3.7 illustrate the dynamics resulting from the implemented diffusion process. They show the time evolution of a 1-dimensional arrangement of entities, which diffuse at every time step into their immediately adjacent neighbours to either side. Entities at the edges interact with the last one on the other side, thus structuring this universe as a ring.

Figure 3.6 illustrates the diffusion process' conservation of the amount of materials in a closed system, as well as the speed of convergence at all entities towards the average amount contained in the system. Figure 3.7 enriches the illustration with a depiction of how external perturbations are quickly spread out. In both cases, we can see how diffusion endows local states with information on a collective system property.

These plots can also be interpreted as providing an illustration of the flow of materials occurring inside a portion of a root, 20 segments long and having no branches. The diffusion process uses here the same parameter settings that are used in the simulation to configure the internal root diffusion (Section 3.4). Note however that the rise in amounts seen in Figure 3.7 is in the simulation countered by the ongoing growth of new segments (new containers to diffuse from/into) and interactions with the shoot (flows of materials involving entities outside the system). Also, apices pay for the energetic costs of growing new segments with a consumption of internal materials.

3.3.3 Update Rules

Sections 3.1 and 3.2 described the soil and root's structure, state variables and initialization. The present section describes the update rules responsible for the system's dynamics. Transformations of two types occur in the system: apex activity resulting in the growth of new root segments, and diffusion processes transforming soil patches and root segments' state variables.

As discussed at length in Section 3.3.1, root growth and diffusion occur over different timescales. This is implemented in the simulation by having diffusion occurring at every time step, but apices updating only once per a parameterizable number of time steps. These two processes influence each other: diffusion actively shapes apices' perceptions of their contexts, and constrains their activities (by influencing the availability of materials with which to pay for the energetic costs involved in growth), but in turn takes place over a set of containers that is undergoing transformation by growth at the apices.

We follow with the presentation of the three implemented update rules, by the order in which they are potentially carried out at every time step in the simulation. The diffusion steps are here presented independently of the state variable they are being applied to. When the time comes in the simulation to execute either the *(b)* or *(c)* diffusion process, it is applied separately in turn to each of the three state variables: Water, Nitrogen and Phosphorus.

(a) update Apices

Chapter 4 is exclusively concerned with the optimization of apex behaviours, that efficiently map environmental perception (Section 4.1) to the appropriate growth decisions. We here focus instead on how apex updating integrates with the rest of the model.

For a root segment to grow, the root must pay the energetic costs of cell division, and for the penetration of the soil patch into which the segment elongates. This cost is specified through a system parameter indicating the amounts of the apex's state variables that are consumed in that process.

A segment's growth costs are constant. The scenario of variable energetic costs dependent on local soil properties was not considered in the model. A growth decision on the part of the apex is then either fully achievable by paying that constant cost, or not carried out at all.

As mentioned in Section 3.2.1, all segments throughout the root are uniform in size (at all times, even as they grow). A situation where the available materials to pay for a growth event are only sufficient for the creation of a smaller segment, is therefore not supported. Should an apex controller choose an action it has no capacity to fully pay for, the simulator cancels that choice.

The parameterized energetic cost must be payed for each individual segment growing out of an apex. An apex that decides to elongate in one direction, while also originating another branch in some other direction, pays that cost twice. The elongation segment takes priority over the branch segment in accessing internal resources to pay for the cost. Should there be resources to pay for only one of the segments, it is the branch the one to be canceled.

Note that as apices grow, they may end up producing multiple root segments per soil patch. Only resource availability limits the number of segments the root is able to place in an area of the soil. Rich soil areas are able to support more segments, but also in poor soil areas we might have diffusions from elsewhere in the root providing for the elongation costs which allow for multiple segments to traverse those poor areas, so as to reach richer patches beyond them.

An apex update involves the stages of context perception, action selection, and action execution. During update, apices may choose either to take no action or to grow a number of segments. Segments grow by the length necessary to take them into an adjacent soil patch, over the number of time steps matching the amount of real-world time required for an apex of the model plant to grow by such a length (Section 3.3.1). In nature, the root would grow incrementally over that span of time, gradually paying for the process' energetic costs, and all the while participating in diffusion processes with the soil and the root segment from which it grows. In our discrete model, where soil patches are always of uniform size, segment growth is implemented through the following sequence of steps:

1. The growth's energetic costs are payed for all at once by immediately deducting from the apex's state variables *relative to the current time step* the amount necessary to cover the *complete* costs;

2. A new segment is immediately added into the simulation, embedded into the chosen soil patch adjacent to the one where the apex it grew from is, and connected to it. Each of this new segment's state variables is initialized with state 0.0 (segment initially contains no amount of that specific material);

3. Over the time steps in between apex updates, the initially empty segment participates in diffusion processes with its soil patch and the segment it grew from. Note that the segment participates in diffusion also in the iteration in which it is added to the simulation, given apex updating takes place prior to the diffusion processes.

When the time comes to update apices once again, the new segment is now fully in place, and containing inside itself the outcome of diffusions over the span of time during which growth took place.

(b) root→root diffusion

The flow of materials occurring internally in the root system is implemented by a diffusion process between neighbouring root segments. At every time step, each root segment assumes once per state variable the role of donor in a diffusion process, having as recipients the segment it grew from, plus the up to two segments that grew from it (potentially in this very same time step). For each state variable, given all segments assume once the role of donor, all segments are therefore equally recipients in up to three other diffusion processes.

Extracted materials will this way, given enough time, find their way to the plant's shoot (see Section 3.3.4 for details on the modeling of root–shoot interaction). However, as diffusion takes materials wherever there is a deficit of them, internal root diffusion may also supply lower branches of the root, which are not by themselves able to guarantee access to a specific material at the same levels as other branches. Besides of supplying apices with the required materials with which to pay for growth's energetic costs, these downwards diffusions contribute also to the direct communication between apices, by making information on global system properties more readily available at a local level. Sections 3.5.1 and 3.5.2 elaborate further on the communication of information between apices.

(c) **soil→root diffusion**

Soil extraction on the part of the root is modeled as a donation by soil patches into the root segments embedded in them. Root segments extract from the soil by creating inside themselves deficiencies in amounts of materials, which are filled up through diffusion from the soil.

At every time step, each soil patch containing at least one root segment assumes once per state variable the role of donor in a diffusion process, having as recipients the undetermined number of root segments that over time grew into it. Through diffusion, each soil patch will distribute at every time step small amounts of the considered material through the many root segments, in proportion to their relative deficits in that material.

3.3.4 Boundary conditions

Cellular automata universes are traditionally one- or two-dimensional structures, where regularity in neighbourhoods is imposed by arranging cells in a ring or toroidal configuration, respectively. No exceptions then need to be made in the rules for handling different types of neighbourhoods throughout the universe.

In our model one might consider the simulated soil to represent the portion of soil in contact with a round vase. Soil patches to one side could then have patches on the other as neighbours (the initialization of random soil configurations would then need to be adjusted, as the distance measure would need to accommodate the possibility of shorter paths between patches through the sides of the simulation). This would not save the model however, from the need to handle the neighourhood exceptions at the soil's surface and bottom layers. Having soil patches in one having patches in the other as neighbours would make no sense.

The edges of the simulated soil are given above as an example, but neighbourhood regularity is equally broken in other parts of the model. Fully listed, these are:

- the modeled shoot segment has no other segment preceding from it;
- root segments have from 0 to 2 other segments grown from it;
- apices perceive 7 soil patches in their vicinity, some of which may lie beyond the soil's edges;
- apex controllers may choose to grow in a direction that would cause the new segment to be placed in a soil patch beyond the soil's edges;
- soil patches diffuse into an undetermined number of root segments.

The handling of these irregularly-shaped neighbourhoods is therefore an important model requirement.

The diffusion process presented in Section 3.3.2 was defined having also these situations in mind. It accepts an undetermined number of recipients, and furthermore, both the donor and the recipients might represent entities which are not fully included in the simulation. In those cases, the value assumed to be present in that missing entity can then be specified through the p_{s_m} variable. This value might be made available through a parameterized constant, or dynamically calculated as a function of some system state.

Given no diffusion into soil patches was implemented at this stage, all diffusions in our model are occurring *into* root segments. The only diffusion from/into a missing entity that the model then needs to consider occurs at the interface with the shoot. The handling of this case is discussed below.

Section 4.1 discusses the handling of apices' perception of states in soil patches beyond the soil's edges, which occurs whenever an apex embedded in one of the soil patches at the soil's edges needs to perceive its context.

Should an apex controller choose to grow into a non-simulated soil patch, the model enforces a cancelation of such choice, and that growth action will then not take place. This type of model supervision of apices' activities was already previously presented, for the validation of whether the energetic costs involved in growth events can be payed for given locally available resources.

Root–shoot interaction

Modeling a root in isolation from the rest of the plant above the surface necessarily leads to an oversimplification of the processes occurring in nature. In real plants, both halves work together so as to achieve a mutual balance in resource acquisition and usage. In nature, water and nutrients extracted by the root flow upwards towards the shoot. These provide materials necessary for the plant's development. Photosynthetic products, among which Carbon, flow in the opposite direction, and provide in turn vital resources for the root's growth.

The shoot, by constituting a bottleneck through which materials going in either direction must pass, assumes from the perspective of both parts of the plant, the role of a central source of signals/information, having an influence on their behaviour. Resource acquisition on the part of the root serves the function of nourishing the plant. Its degree of activity is therefore controlled by the plant's overall needs. The aerial part of the plant consumes a higher amount of materials (thus acting as a stronger sink in the diffusion process) in periods in which it is undergoing more accentuated growth.

We model root growth to the degree of realism necessary to achieve an efficient technological transfer of its behaviours. Explicitly modeling processes occurring above the surface would have significantly increased the model's complexity, and would have provided us essentially with the capability to externally control an apex swarm's degree of activity, by external adjustments the degree to which their products are extracted from their own communication network (the root's structure).

Because we are interested in the design of collective foraging behaviours, and not necessarily at this point in global activity level coordination based on the operations of an information centralization entity, we could then ignore the explicit modeling of processes occurring above the surface. The appropriate way to configure diffusion at the root–shoot interface, given then behaviour we aim for, is then to assume a phase of maximum growth in the plant, that requires as many resources as the root can supply. The greater the pull from the shoot, the more the root is forced to extract from the soil to compensate, leading the root to actively forage in search of sources capable of providing for all the shoot's needs.

At the single simulated shoot segment, diffusion into its preceding segment is diffusion into an entity that is not explicitly simulated. So as to achieve the dynamics mentioned above, the diffusion process is configured to assume that missing has $p_{s_m} = 0.0$, in all state variables, and in all time steps. This way, we simulate a shoot in permanent maximum need of materials, leading the root to pump upwards as much as it can. Furthermore, in another departure from biological realism, we are not constraining apex activity on the availability of photosynthetic products flowing from the shoot, such as Carbon, thus allowing for a greater degree of apex activity.

3.4 Default parameter values

In this section we list, for completeness, the numerical values we assigned in all our simulations to the various parameters that define the simulator of soil and root dynamics described in this chapter.

'Root/ranges' : $(1.0, 1.0, 1.0)$

Maximum amounts of [Water, Nitrogen, Phosphorus] a root segment in capable of holding.

'Root/Seed amounts' : $(1.0, 1.0, 1.0)$

Amounts of [Water, Nitrogen, Phosphorus] inside the single root and shoot segments when a simulations starts.

'Root/max diffused per it' : $(0.25, 0.25, 0.25)$

Maximum amounts of [Water, Nitrogen, Phosphorus] a root segment is able to diffuse out of itself and into its adjacent root segments per iteration.

'Root/max extracted per it' : $(0.05, 0.05, 0.05)$

Maximum amounts of [Water, Nitrogen, Phosphorus] a root segment is able to extract from the soil per iteration.

'apex/consumed by growth' : $(0.10, 0.10, 0.10)$

Amounts of [Water, Nitrogen, Phosphorus] the apex must deduct from its own variables in order to grow one segment into an adjacent soil patch.

'apex/update frequency' : 10

Diffusion takes place at every time step, but apices will only be allowed to take action once at the beginning of every sequence of 10 time steps. They then act in time steps $(0, 10, 20, \ldots)$, with the outcome of their action appearing in the simulation at the following time step.

'apex/update cycles' : 20

This is the simulation stopping criteria. Once the apices have have been updated this number of times, the simulation stops (total number of diffusion iterations will then be = 'apex/update frequency' * 'apex/update cycles' = 10 * 20 = 200 iterations).

'Soil/dimensions' : $(33, 33)$

Number of hexagonal soil patches per dimension in the simulated soil.

'Soil/Water/Range' : $[0.0, 1.0]$

'Soil/Nitrogen/Range' : $[0.0, 1.0]$

'Soil/Phosphorus/Range' : $[0.0, 1.0]$

Upper and lower bounds for soil variables representing Water, Nitrogen and Phosphorus.

'Soil/Water/nrPeaks' : 32

'Soil/Nitrogen/nrPeaks' : 32

'Soil/Phosphorus/nrPeaks' : 32

Number of peaks in randomly generated initial soil configurations.

'Soil/Peak/decay rate/range' : $[0.20, 0.65]$

Range in which peaks' decay rates will be randomly defined, when generating initial soil configurations.

Using the notation introduced in Appendix B, for Diffusion's algorithmic definition, the parameter values listed above lead the different diffusion processes to assume the following configurations:

soil→root: $d_k = 0.0, d_{o_{max}} = 1.0, r_c = 1.0, r_{i_{max}} = 0.05$

root→root: $d_k = 0.0, r_c = 1.0, d_{o_{max}} = r_{i_{max}} = 0.25$

3.5 Discussion

3.5.1 Direct communication between apices

A significant degree of coordination is required for apices to be able to construct through their individual actions a structure that benefits the plant as a whole. Apices must achieve that goal while subjected to a significant degree of uncertainty about the environment they are growing on. The exact locations of the best patches for exloitation is not known to them beforehand. Apices' collective blindness is counterbalanced by the internal root→root diffusion. The segments grown as the root develops provide a communication channel through which materials diffuse, carrying with them informational clues apices can make use of to coordinate and decide on the appropriate course of action. As seen in Figures 3.6 and 3.7, the diffusion process allows for a distributed approximation to global states. The speed at which this information flows is further discussed in Section 3.3.1.

The materials extracted by root segments do not necessarily flow unidirectionally towards the shoot, forever disappearing from their grasp. The rate at which different materials diffuse out of a root segment already provide a local clue on the rest of the root's performance, as a local difficulty in draining materials indicates elsewhere other segments are also managing to adequately fulfill the same goal. But as diffusion produces flows in the direction of greater to lower concentration, downwards flows might equally occur, and these have the capacity to supply segments with materials they lack the capacity to extract by themselves.

Being root segments continuously subjected to diffusion with neighbouring segments, the concentrations they find within themselves at any given point in time provide them with a degree of information on collective system processes. Furthermore, these internal concentrations become more strongly imbued with informational value, as they are considered together with the concentrations in the same materials that the apex observes in its vicinity. The importance of investing in local extraction of a given material can be inferred from these relative amounts. An apex that does not observe internally sufficient quantities of something that is abundantly available around itself, is informed of the need to invest in local exploitation of available resources (its internal deficiency will arise in this context from fast diffusion out into the root). Conversely, internal abundance in the context of local depletion informs it that it is the culprit of resource retention by the root, as it is forcing others to supply it with what it cannot provide by itself.

The appropriate course of action is to a significant degree determined also by how amounts compare across the multiple state variables diffusing in and out of an apex. An apex finding itself in that last scenario, might still find it important to invest in local exploitation: it would be informed it could not by itself fulfill one of the root's needs, but others could do it for itself, while the internal diminute amount in a different state variable by comparison with its local abundance would tell the apex that in another goal, it is able to provide for something the rest of the root is having trouble fulfilling. Were not this the case, were the apex to find itself in an area locally depleted of all its state variables, its appropriate course of action would then be influenced by whether internally it was supplied by others in quantities sufficient to pay for growth's energetic costs, and whether it could sense in its vicinity a gradient in concentrations meriting the gambling of those resources on growth events with the expectation to reach that way parts of the soil with greater availability of resources.

The previous rationalizations of decision-making at the apices does not necessarily explain the processes through which in nature they take their growth decisions, as these are to a great extent not completely understood. It does illustrate however, that apices might be able to coordinate collectively, not through the communication of fully descriptive characterizations of their contexts (something often done in robotics), but through minimal communications of signals with sufficient informational value to properly bias behaviour in a competitive environment. Though in nature these signals include also the broadcasting of signalling molecules with explicit semantics aimed at influencing elsewhere behaviour in precise ways (as discussed in Chapter 2), in our model we limit the perception of distant states to that enabled by the transformations diffusion effects on local internal concentrations. We this way abdicate on the capacity to directly and explicitly order others on their course of action, and put instead the emphasis on the personal identification of the actions which maximize collective performance.

3.5.2 Stigmergy through extraction: the soil as an external memory

Stigmergy is an organizing principle in which individual parts of the system communicate with one another indirectly by modifying and sensing their local environment. In our model, this occurs through the extraction of materials from the soil, that is conducted by root segments (soil→root diffusion). In addition to the direct communication channel made available to apices by diffusion occurring internally in the root, apices can therefore also make use of the information provided by the transformations other root segments have affected in their common environment, the soil.

An example of stigmergic communication that is frequently given, is that of the laying down of pheromones by ants as they explore their environment. A few parallels can be drawn here between root growth and ants' foraging for food. Pheromone paths can be seen as equivalents to the root structure, and ants the diffusion process conducting the transportation of materials. A difference can already be seen in that all segments throughout the root cooperate in extracting from the soil, while ants "extract" at path's extremities. A stronger distinction between both systems is to be found though in that once the root structure is laid down, it is static. Ants, however, are free to transform their paths over time, in answer to changing availabilities of food throughout their environment. The cost for an apex that takes a wrong growth decision (and for the root as a whole) is thus much higher, especially when we take into account that growth actions always build upon the outcomes of previous actions. We might therefore conjecture that using roots as a source of inspiration will lead to better collective decision-making under uncertainty, while ants will be more suited to decision-making in dynamically changing environments.

In both cases, the environment is used as an external memory of the previous actions undertaken by the system. Apices are capable of taking different actions in the same patches of soil, depending on their degree of exploitation by segments previously grown in those same patches. While increasing concentrations of pheromones along a path provides a positive reinforcement, leading ants to allocate higher degrees of activity along those paths, extraction of materials from a patch leads other apices to allocate fewer resources in the exploitation of those same patches.

Finally, very strong discontinuities in materials' concentrations are not inserted in the soil by the initialization process described in Section 3.1.3. If an apex perceives around itself patches possessing concentrations significantly distinct from those in the remaining patches it perceives, it can infer the locations where other root segments have potentially already been deployed, allowing it to guide its growth into soil patches that may over time allow the root as a whole access to greater amounts of materials.

Chapter 4

Evolutionary Design of Apex Behaviours

Having in the previous chapter introduced the modeling of soil and root dynamics, we now turn to the automatic design of apex behaviours that maximize the root's performance at a collective level. The behaviours considered here are reactive: apices sense their conditions at discrete time steps, and immediately take the appropriate action for those conditions. The challenge is then to come up with mappings from local perception to actions, such that through repeated, parallel application of those mappings, a collective performance measure is optimized.

The sub-sections below introduce first the information that is made available to each apex at each time step and the actions it can perform (Section 4.1), and follow with the description of the controller's architecture (Section 4.2), and the optimization algorithm used to optimize it (Section 4.3). In Section 4.4 we describe the setup used to optimize apex controllers, and in Section 4.5 we discuss the obtained results.

4.1 Apices perception vector, and set of possible actions

Whenever apices are allowed to act, they perceive a vector of continuous values characterizing the soil conditions in their vicinity, as well as the conditions internal to the apex. This perception vector contains the following sequence of values:

Internal states: amounts of Water, Nitrogen and Phosphorus inside the apex root segment;

Local states: amounts of Water, Nitrogen and Phosphorus contained in the soil patch in which the apex is embedded;

Neighbouring states: amounts of Water, Nitrogen and Phosphorus contained in the six soil patches around the one in which the apex is embedded, read in clockwise direction, starting at north-east.

A total of $3 + 3 + 6 * 3 = 24$ state variables are thus taken into account by the apex in order to judge on the appropriate course of action.

Perception of state values is not affected by noise. The apex is then able to perfectly measure state variables. Other than in the random initialization of soil conditions, a root growth simulation is then fully deterministic.

Upon processing its context, the apex controller takes one of three available decisions:

Stay Still: the apex takes no action in the current time step;

Elongate: the apex will elongate. A direction for growth must then be equally specified. Considering the apex has internally enough materials with which to pay for the growth's energetic costs, by the next time step a new root segment will have been added to the simulation, embedded in a soil patch contiguous to the one from which it grew. As defined in Section 3.2.1, any root segment being a *leaf* of the binary tree is considered to be an apex. As an apex grows, therefore, the segment previously considered to be an apex loses that designation (as it now has a segment growing from it), and the newly created segment, having no segments growing from it, is considered to be an apex. Next time apices are allowed to act, that is the segment that will then perceive and act on its environment;

Elongate and Branch: Similar to the previous action, but growing two instead of only one new segment out of the apex that takes the decision. The apex controller must then supply two growth directions: that of the 'elongation' segment, and that of the new 'branch' segment. The considerations drawn before for the Elongate decision apply here to both of the new segments.

Apex behaviour is then a mapping from a set of 24 continuous variables characterizing the apex's context, into that of the 3 actions above which is most suited to be performed in the current context (set of possible actions is enlarged by the need to specify in addition the growth directions, to a total of $1 + 6 + 6 * 6 = 43$ distinct possible actions).

4.1.1 Boundary conditions

Whenever an apex is contained in a soil patch at the edges of the simulated soil, it will not have access to information about the states of all its surrounding soil patches. So as to always produce an apex state vector with the same amount and ordering of variables, the states of all non-simulated

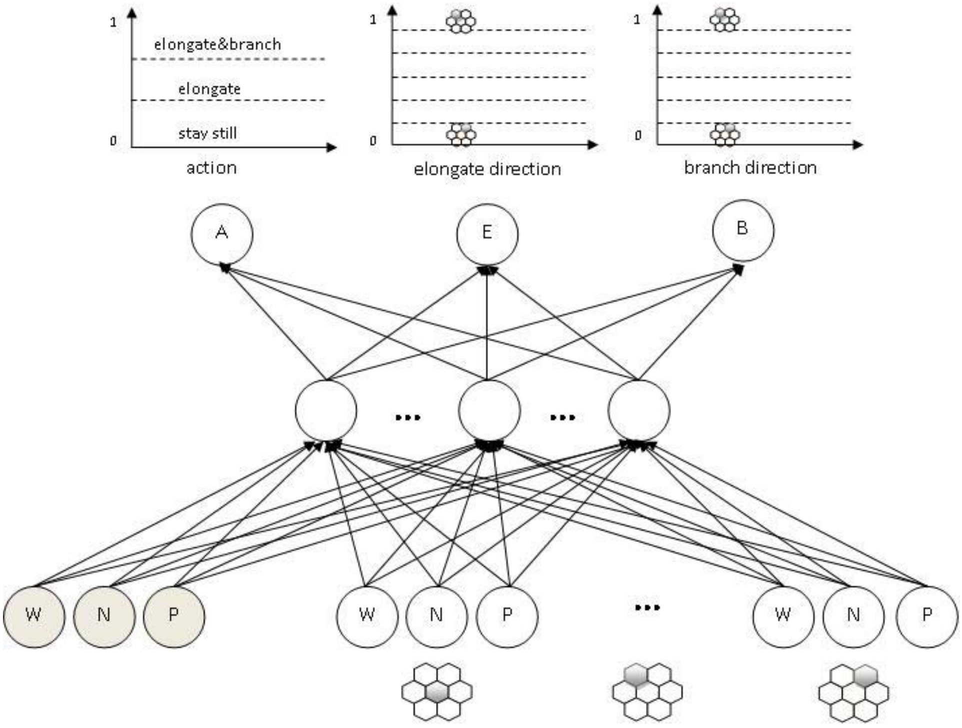


Figure 4.1: Neural Network Architecture. The feedforward fully connected structure is shown.

entities (patches beyond the soil's edges, or above the surface) are assumed to be 0.0.

4.2 Apex controllers' encoding

The apex controller we chose to implement is a feedforward artificial neural network (multi-layer perceptron, MLP). While we did try other encodings, such as Pittsburgh Learning Classifier Systems [Lanzi, 2008], we obtained the best results when experimenting with the MLP and we thus here present those results. In our implementation, the ANN consists of one hidden layer comprising 10 neurons. In Figure 4.1, we depict the network's architecture showing only three of the hidden neurons for the sake of the image clarity; inputs and outputs are also briefly described.

The input layer consists of 24 neurons which encode the apex sensing. In particular, 3 neurons (depicted in grey in Figure 4.1) encode the internal state of the apex and receive as input the variables representing the amounts of Water (W), Nitrogen (N) and Phosphorus (P) within the apex, 3 neurons receive as inputs the variable W, N, and P from the cell in the soil where the apex currently is. The rest of the neurons receive as input the variables W, N and P from the 6 neighboring cells. All input neurons are connected to all hidden layer neurons.

The outputs of the network define the action to be performed at the next time step by the apex. The first output neuron (neuron A) defines the action to be performed: either stay still, elongate, or elongate and branch. This is implemented with a thresholded activation:

- if neuron A's activation is within $[0, \frac{1}{3})$ then the apex performs action stay still
- if neuron A's activation is within $[\frac{1}{3}, \frac{2}{3})$ then the apex performs action elongate
- if neuron A's activation is within $[\frac{2}{3}, 1]$ then the apex performs action elongate and branch

The activation of output neuron E defines the direction of the elongation, if elongate has been selected by output neuron A as the action to be performed by the apex. The different possible directions are the 6 neighboring cells, and thus the network chooses a different direction depending on the activation of output neuron E and if it is within the following bounds: $[0, \frac{1}{6})$, $[\frac{1}{6}, \frac{2}{6})$, $[\frac{2}{6}, \frac{3}{6})$, $[\frac{3}{6}, \frac{4}{6})$, $[\frac{4}{6}, \frac{5}{6})$ or $[\frac{5}{6}, 1]$. Notice that the output of this neuron is disregarded if output neuron's A activation is not within $[\frac{1}{3}, \frac{2}{3})$.

Finally, the activation of output neuron B defines the direction of the branching, if elongate and branch has been selected by output neuron A as the action to be performed by the apex. The different possible directions are the 6 neighboring cells, and thus the network chooses a different direction depending on the activation of output neuron EB and if it is within the following bounds: $[0, \frac{1}{6})$, $[\frac{1}{6}, \frac{2}{6})$, $[\frac{2}{6}, \frac{3}{6})$, $[\frac{3}{6}, \frac{4}{6})$, $[\frac{4}{6}, \frac{5}{6})$ or $[\frac{5}{6}, 1]$. Notice that the output of this neuron is disregarded if output neuron's A activation is not within $[\frac{2}{3}, 1)$.

The implemented ANN uses bias units. There is one bias unit that connects to all the hidden layer neurons, and another bias unit that connects to all the output neurons. We then have to optimize in total 283 continuous variables, which map to the following network connections: (24 variables perceived in the apex's context + 1 bias unit) * 10 hidden layer neurons + (10 hidden layer neurons + 1 bias unit) * 3 output neurons. Both weights and biases are initialized in $[-1.0, 1.0]$.

4.3 Particle Swarm Optimization

The weights and biases in the Feedforward Neural Networks controlling apices were optimized using the Particle Swarm Optimization [Poli et al., 2007; Clerc, 2006; Kennedy et al., 2001] (PSO) algorithm. In PSO we have, as its name indicates, a swarm of n particles collectively exploring the search space. Each particle $i \in \{1, \ldots, n\}$ is defined by $\vec{x}_i$, its current position on the search space, $\vec{p}_i$, the best position visited by the particle so far, and $\vec{v}_i$, its velocity. At each step, the particle updates its velocity vector, and afterwards updates its position in the search space with $\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$.

The core of the PSO algorithm lies in the equation used to update particles' velocity. Though several alternatives exist, the canonical algorithm Poli et al. [2007] uses the equation with "constriction coefficients":

$$\vec{v}_i \leftarrow \chi(\vec{v}_i + \vec{U}(0, \phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2) \otimes (\vec{p}_g - \vec{x}_i)) \qquad (4.1)$$

where $\vec{U}(0, \phi_i)$ represents a vector of random numbers uniformly distributed in $[0, \phi_i]$ which is randomly generated at each step and for each particle, and $\otimes$ is the component-wise multiplication. $\vec{p}_g$ refers to the best position visited so far by the particle's neighbors. The constriction factor χ is usually set to 0.7298, along with both ϕ_1 and ϕ_2, also called the acceleration coefficients, set to 2.05 [Clerc and Kennedy, 2002].

An important concept in PSO is the population topology. Particles are arranged in a graph, and only interact with their immediate neighbors. Classical topologies that have been used with PSO are the *gbest* ("global best") and *lbest* ("local best") topologies. *gbest* is basically a fully connected graph, where all particles influence each other. In *gbest*, information on best positions found by each particle is immediately available, and convergence is faster. In *lbest*, using a radius of k, particles are arranged in a ring, and particle i is connected to each particle in $\{(i + j) \bmod n : j = \pm 1, \pm 2, \ldots, \pm k\}$. Though slower to converge, *lbest* is less vulnerable than *gbest* to getting stuck in local optima [Poli et al., 2007].

4.4 Experimental setup

Particle Swarm Optimization was used to optimize continuous vectors containing 283 variables (as previously explained), which decode into the configuration of Artificial Neural Networks for controlling individual apices during root growth simulations. All variables are randomly initialized with uniform probability in the range $[-1.0, 1.0]$. Strict bounds are not enforced on variable's values, so they may sample values outside that range. PSO used swarms of 49 particles, structured in an *lbest* ring topology, where each particle is influenced by the 4 particles within a radius 2 of itself in the ring. PSO used the equation with "constriction coefficients" to update particles' velocities through the search space (constriction factor χ set to 0.7298, and both acceleration coefficients ϕ, set to 2.05). Velocity vectors are clamped after updating to half the range in which variables are initialized: $[-0.5, 0.5]$, a standard practice in PSO [Clerc, 2006]

Particles were updated synchronously, in a generational mode where also objective value evaluation and updating of particle's memories take place logically at the same time for all particles. Swarms where optimized over a span of 250 generations. Every generation, a set of 250 soil configurations was randomly generated, and all particles (along with their memory positions) were (re)evaluated in this common set (this setup is discussed in more detail in Section 4.4.1). In total, 10 independent experiments were conducted with the settings listed above, to train apices' neural networks.

All individual root growth simulations used the parameter values listed in Section 3.4.

We tuned the generation of random soil configurations (Section 3.1.3) in a way that would provide challenging conditions in which to test apex controllers. Soils with very abundant materials are easily exploitable, as about any growth behaviour will end up falling on top of rich soil patches. Not enough pressure would have been made on apices to intelligently process their inputs, and appropriately judge on the best course of action. Having an abundance of poor soils, on the other hand, puts great pressure on apices, as every decision taken in an undernourished context, while diffusion is quickly pumping the few available materials into the shoot, might mean the difference between achieving in the end an extensive root structure, or dying out close to the seed's location by having invested the few available materials in growing in the wrong direction.

Also, for a root to be robust enough, apices should be capable of taking the right decisions no matter the context they find themselves in. For this to be the case, however, we must guarantee that during optimization apices become exposed to the greatest possible variety of contexts. As apices are expected to climb their local gradients into patches richer in materials, in soils initialized with abundant availabilities of materials, apices would be able to just jump between patches having high concentrations, not being adequately exposed to poor soil patches, and therefore not fully learning how to act on such situations.

4.4.1 Objective function

The primary root function we are interested in is that of efficient exploration of an unknown soil, in search of the materials essential for the plant's development, and the creation of a materials acquisition network, to extract and channel those same materials. To assess a given root's performance, all we then need to do is measure the amounts of materials it is able to extract in the time span of a root growth simulation, by comparison with what different roots grown in the same soil conditions are able to. The greater these amounts are, the more successful was the apex controller.

The task assigned to apices is one of collective multi-objective foraging. Apices should grow in a coordinated manner, so that the plant will have access to the greatest possible amounts of all the material being extracted from the soil (Water, Nitrogen and Phosphorus). The apex controller must learn to balance the degree to which the multiple goals are achieved, as it is not acceptable for the root to concentrate on the extraction of one or two of the materials at the expense of the remaining ones. We therefore define the measure assessing performance of a root grown through application of a given apex controller, in one soil, to be the average of the total amounts of each material that it extracted in the time span of the simulation. Note that all materials are equally important in this measure. We are not giving greater weight to Water, nor any other material, when averaging extracted amounts. Consider Figure 4.7 as an example: the root extracted in that simulation $(58.27, 53.95, 73.56)$ of Water, Nitrogen and Phosphorus, respectively. The average, 61.93, is taken as the measure of that root's performance in that soil.

The same apex controller is applied repeatedly and in parallel by all apices during root growth. We optimize the decision-making of individual entities in the system (apices), but evaluate the performance of the system as a whole (root). This setup is known to facilitate the appearance of altruistic behaviours, as "selfish" decisions by the apices, which locally increase their performance but have a detrimental effect over time on the whole root's performance, are less likely to be selected for.

Handling of noise in the objective function

The objective value of a certain apex controller depends not only on the search space position of the particle encoding it, but also on the environment in which that controller is evaluated. A controller's objective value is therefore a random variable which must be approximated during optimization, while having in the optimizer mechanisms to handle the noise in those estimates [Jin and Branke, 2005].

An apex controller's objective value is estimated by averaging the performance measures of roots grown using that controller, on a multitude of soil configurations.

To increase the informational value of a controller's performance estimate, we force all particles in the swarm to always be compared in terms of their performance on the same set of randomly defined initial soil configurations. So as to avoid overfitting apex controllers to the specific properties of a given training set, we periodically redefine during optimization the set's composition. This procedure has been successfully used in the past for the optimization of Cellular Automata dynamics [Mitchell et al., 1994; Crutchfield et al., 2003].

We implement this procedure into PSO by having particles updating synchronously, in a generational mode, rather than asynchronously as is typical in PSO. At every generation, a new set of soil configurations is randomly defined, and particles are evaluated on it.

Because in PSO successive positions in the search space are sampled as a function of memories of best previously visited search space locations, particles might be misguided by over- or underestimations of their objective value. These would result from the set of random soil configurations possessing exceptionally favourable or unfavourable characteristics, not adequately representative of the whole space of possible soil configurations. To alleviate that problem, both particles' current and previous best positions are (re)evaluated at the end of every generation, on the same set of randomly defined soil configurations all other particles are also currently subjected to. After objective value evaluation, particles check as usual whether their current position is better than the best one they had previously been to, overwriting the latter with the former should that be the case. This reevaluation mechanism for coping with uncertain objective functions in PSO is also used for instance by [Carlisle and Dozier, 2002].

Without this constant reevaluation of both current and previous best positions, we might have Neural Networks persisting in the swarm, not due to some relative superiority, but because chance had them being evaluated precisely in that generation where the set of random soil configurations had characteristics very favorable to exploitation by the behavior they encoded. For an ANN to remain influential over time, when constantly competing through reevaluation against other particles in the swarm, it must be a robust overachiever in most of the soil configurations the optimizer throws at it.

Figure 4.2 provides an illustration of how apex controllers' performance values are estimated. Each of the 200 integer values in the x-axis corresponds to a distinct initial soil configuration that was randomly defined for use in that generation. The 128 search space positions represented in that swarm (neural networks capable of controlling apices) are used to grow 128 roots in each of the 200 soils. Blue dots indicate the performance achieved by a single root in one soil. The green line connects the maximum performance values achieved by any neural network in the swam, in each of the 200 soils. The average of all those values, 86.40, provides an indication of what might be achievable in those soils. The light-blue line, in contrast, connects only the performance values of the controller among the 128 that achieved on average the highest performance. We see instances where the best controller was also the top performer for a given soil, but we see equally situations where it was greatly outperformed by other controllers in the swarm.

As previously discussed, PSO moves particles into new positions in the search space based on their current positions, and memories of best positions visited by a subset of particles. Errors in estimations of objective fitness values will cause the algorithm to misguide particles. The more root growth simulations we perform per ANN, the lower our uncertainty will be, but that certainty is bought at the cost of increased computation time. Handling this

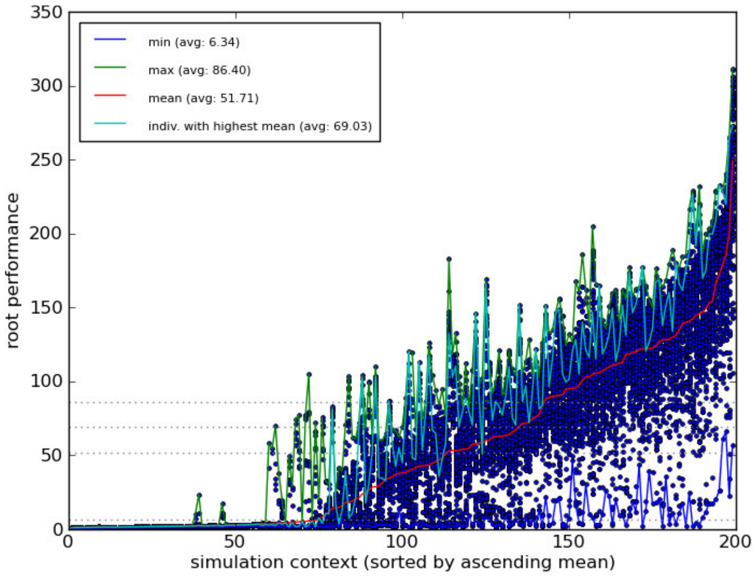


Figure 4.2: Testing the estimation of apex controller's objective values: a swarm of 64 particles (128 search space positions) is here (re)evaluated at the end of a generation on a set of 200 randomly generated soils (●: performance of one root in a given soil).

trade-off is an important design decision when configuring the optimizer. We opted to simulate every ANN over 250 randomly generated soil conditions per generation. This value is significantly higher than the number of simulations traditionally conducted in evolutionary robotics, which in theory should allow for optimization to produce better and more robust solutions. Still, as shown in Figure 4.5, a 4-fold increase in the number of simulations, to 1000, already changes the landscape in terms of particle's performance. The optimization process is still sometimes being misled, and taking particles into inferior locations of the search space. We can take the perspective that the algorithm is performing a greater degree of exploration, and account for it by letting it run for a higher number of generations. This consideration led us to the setting of 250 generations per optimization run.

4.4.2 Standard test set

In order to get a more accurate assessment of the relative performances of different apex controllers found in different experiments, we defined a large set of 10^4 random soil configurations to serve as a common basis for use in comparison. As we are using in the soil simulator a random number generator with a period of $2^{64} - 1$, the odds of the soils in the standard test set having been used during optimization are extremely low. The standard test

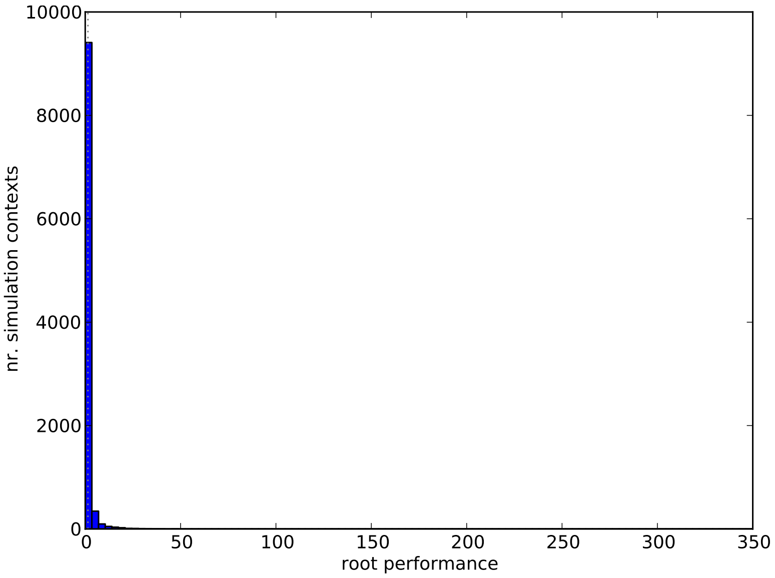


Figure 4.3: Evaluation of the performance of a controller that has apices taking random growth decisions, on the 10^4 soil configurations in the standard test set: histogram of the performance values achieved by the grown roots (mean: 1.435, standard deviation: 2.582). Compare with the results for the best found controller, shown in Figure 4.6.

set then provides a common base of soil configurations to which controllers were most likely not exposed before. Through it we can test different controllers' learned generalization. Furthermore, as the same set is used to test multiple controllers, we lower the risk of selecting an inferior controller due to having subjected it alone to a simpler set of soil configurations. We do keep to some degree, however, the risk of picking a common set that is not fully characteristic of the full set of possible soil configurations, that thus leads us to selecting worse controllers that happen to be better performers under those uncharacteristic conditions.

Figure 4.3 shows the performance values achieved on the standard test set by a controller that has apices taking random decisions every time they are updated. Apices choose uniformly at random one of the three actions available to them, and in case the decision involves elongation or branching, the direction in which the new segment(s) will grow is also chosen uniformly at random from the six available. As we can see, especially by comparison with Figure 4.6 which presents performance values achieved by the best found controller on the same set of soils, random growth decisions blindly guide apices into soil patches from which not enough materials can be extracted, and the root then becomes starved of materials, unable to pay for continued growth. This provides a baseline against which the learning achieved by the optimizer can be compared.

The exact details on how to construct our test set are given in the Appendix C.

4.5 Experimental results

The optimization of apex behaviours was conducted through 10 separate experiments, each optimizing neural networks over 250 generations. Each generation, 98 particles were evaluated on 250 random soil configurations. In total, over 61 million root growth simulations took place. Each simulation iterated over 200 time steps, resulting in a total of over 12 billion time steps computed in the cellular automaton of soil and root dynamics. Considering the root growth simulations we conducted had been occurring in the real world, in a piece of land where 10 thousand herbaceous plants such as the one we considered are born every year, the root growth behaviour we obtained in the end would have been the outcome of 6125 years of evolution. Simulation, fortunately, allows us to get to the result in a smaller amount of time.

The cellular automaton of soil and root dynamics was implemented in C (compiled with GCC 4.4.0, using the options `-O3 -march=native`), and the code implementing the optimization process was implemented in Python (version 2.6.4). As none of the code was parallelized, optimization ran sequentially on a single core of an Intel Core 2 Duo E8600 processor (3.33 GHz, 6 MB L2 cache) over a span of 5 days and 10 hours. Disregarding the cost to execute the Python code handling the particle swarm during the optimization process, on average the 200 time steps of a full root growth simulation were computed in 7.65 milliseconds. The 250 root growth simulations over which every neural network apex controller was simulated once per generation were thus completed on average in 1.9 seconds.

Figure 4.4 shows how performance progressed across generations in these experiments. We see a steady increase in performance, slightly slowing down after generation 200. As performance is a random variable, we see the best solutions' quality values varying stochastically across generations, as the randomly generated soil conditions considered every generation are simpler or harder to explore than average.

The effects of noise in the objective function can be mitigated, but never fully avoided. Figure 4.5 shows the objective values of all positions encoded in a swarm at the end of optimization. It gives us a more detailed snapshot of the final performance values of particles in the swarm which produced the best controller, from among the 10 experiments depicted in Figure 4.4. We see for instance, in the final column, the best particle's previous best position having a performance value of 59.76 (estimated after the optimization process' last generation, over a set of 250 soils), falling to 57.16 when it is

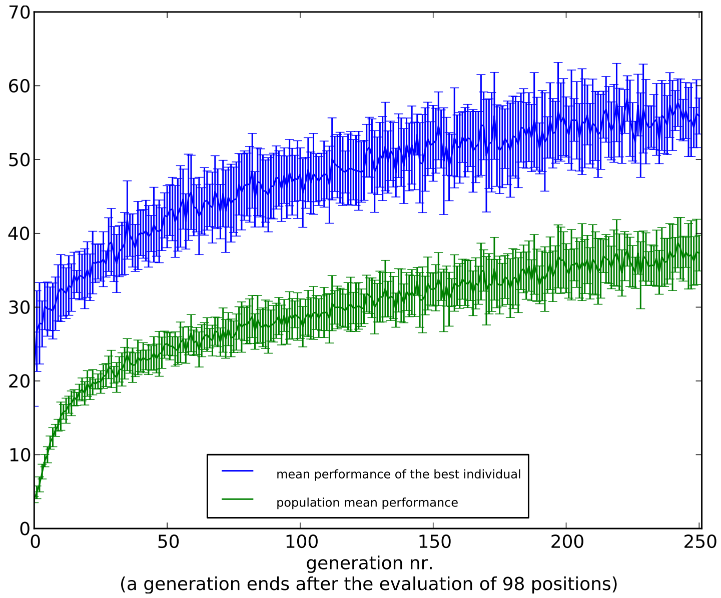


Figure 4.4: Evolution of best and mean performance values in the population (averaged over 10 optimization runs).

retested on a different set now with 10^3 soils, and we see in Figure 4.6 this same controller measured as having a performance of 58.41 when retested over the 10^4 soils in the standard test set.

Though by the 250th generation the swarm is expected to have mostly converged on a single area of the search space, we see from the difference in performance values of current and previous best positions in Figure 4.6, that small steps in the search space are leading to significantly worse-performing behaviours. This illustrates the complexity of the task at hand. We are optimizing a dynamical process of root construction, where slightly different decisions can take the system down different paths, having a cumulative effect, and leading to very different outcomes.

Figure 4.6 shows the histogram of performance values of roots grown using the best found apex controller, on the 10^4 soils in the standard test set. It makes clearer something that could already be observed in Figure 4.2: the parameterization used (Section 3.4) to configure the generator of random soil configurations (Section 3.1.3) causes a great number of soils to have insufficient materials in the vicinity of the seed, that might allow for the root to grow well beyond its starting location in the soil. The initial amounts of materials in the seed allow for only a few segments to grow[1]. If

[1] Both the single shoot segment, and the initial root segment have all their state variables initialized to 1.0. These amounts could potentially pay for the growth of 20 segments (at the cost of 0.1 in each state variable to grow every new segment), however, at the same

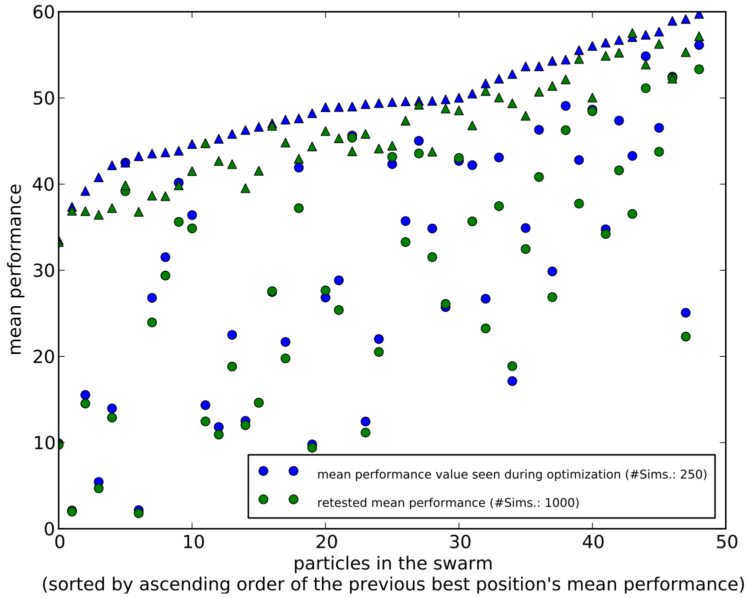


Figure 4.5: The effects of noise in the objective function: particles in the swarm, at the best run's last generation, are depicted according to their performance as estimated during the optimization (in blue), and as estimated by reevaluating (in green) over 1000 random soil configurations (△: a particle's previous best position; ◯: the particle's current position).

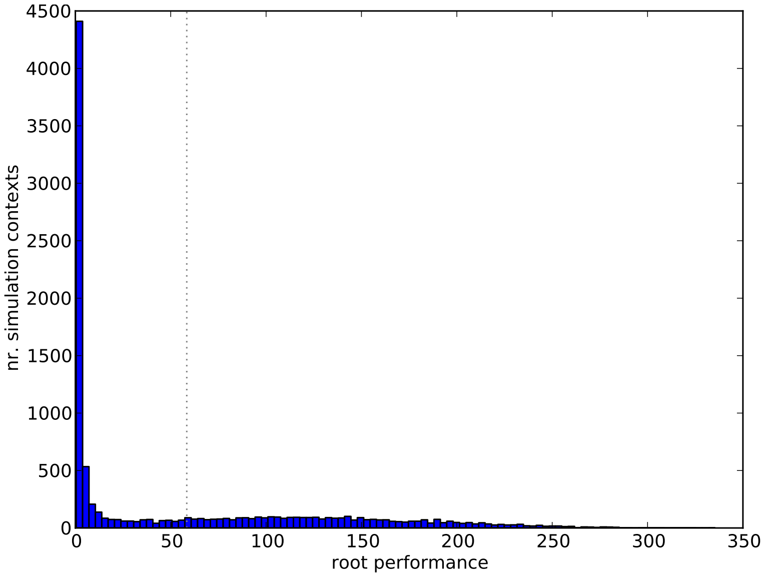


Figure 4.6: Reevaluation of the best found Neural Network apex controller on the 10^4 soil configurations in the standard test set: histogram of the performance values achieved by the grown roots (mean: 58.412, standard deviation: 73.000). Compare with Figure 4.3 (note the scales).

by the time the materials in the seed are depleted, the root has not secured access to sources from which it can extract additional materials, it is then not able to grow anymore, and performance will then be only slightly above 0.0. As the root needs to secure enough amounts in simultaneously the three types of materials, any single one of them might become a limiting factor to the root's development. Figure 4.8 shows such a situation, where the lack of Nitrogen (and also Water, though to a lesser degree) is limiting the root's capacity to grow and branch. In this case, the root was able to extract enough materials to cover the elongation costs, up to the point in which a rich patch of Nitrogen is identified. From that point onwards the root was no longer constrained, and it was then able to branch abundantly, and to extract a high amount of materials.

Figures 4.7–4.14 show 8 distinct initial soil configurations, and the roots grown on them, always through application of the exact same apex controller (the best neural network found in the optimization experiments). These roots show a remarkable degree of phenotypic plasticity, with the developmental process being highly influenced by the characteristics of the soil the root is growing on.

Individual root segments are shown in each panel with a thickness proportional to the amount of that specific state variable it contains internally. We therefore see, for instance, in Figure 4.7 the left part of the root handling most of the plant's extraction of Water, while Phosphorus is obtained mostly by the right side.

time that the the root is growing, diffusion with the shoot is taking outside the simulation up to 0.25 in each state variable per time step. The number of segments the root is capable of growing without extraction of materials from the soil is therefore well below 20.

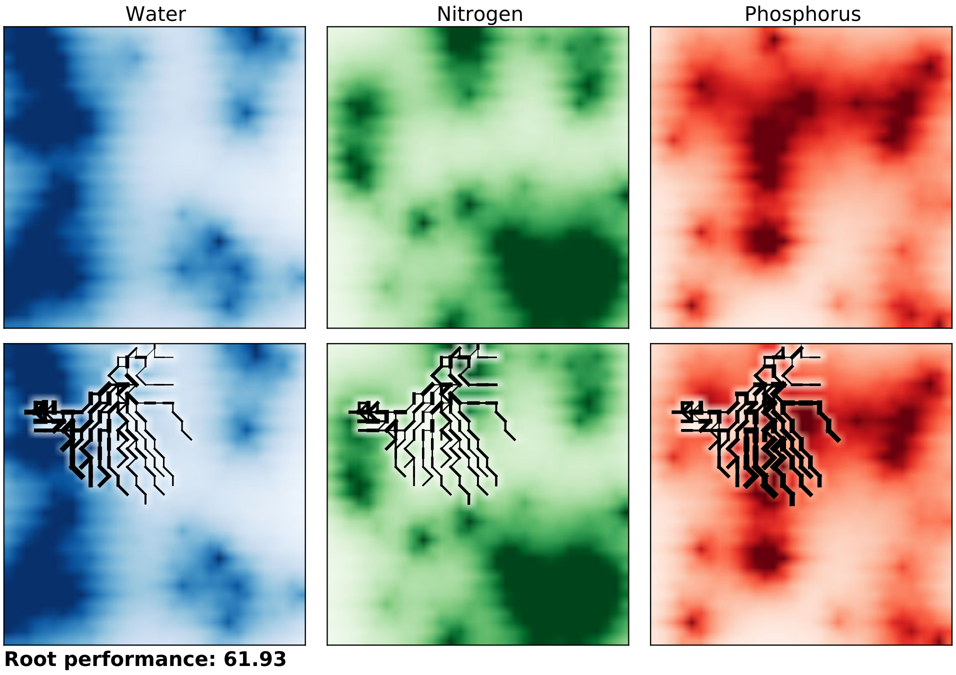


Figure 4.7: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(58.27, 53.95, 73.56)$ out of a soil initially holding $(555.59, 542.37, 499.89)$, a fraction of $(0.10, 0.10, 0.15)$.

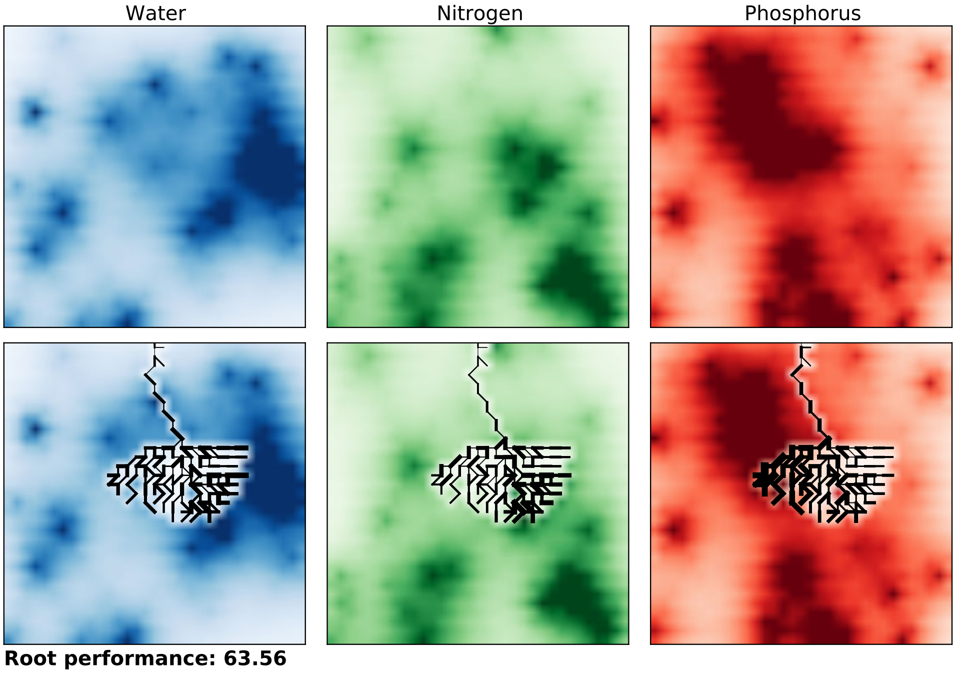


Figure 4.8: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(61.17, 59.08, 70.43)$ out of a soil initially holding $(503.25, 461.58, 630.45)$, a fraction of $(0.12, 0.13, 0.11)$.

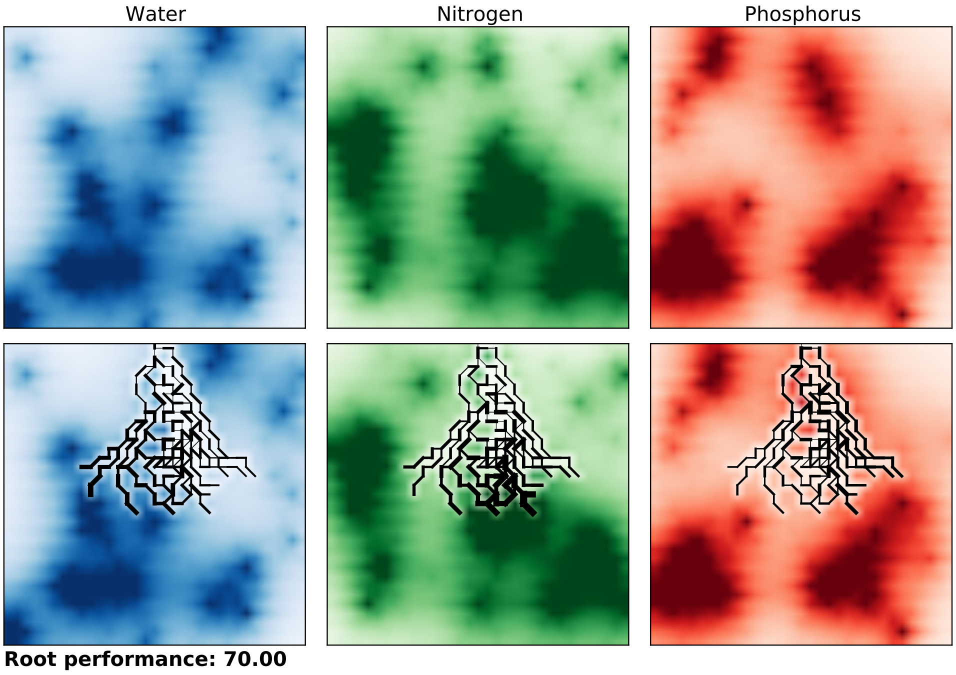


Figure 4.9: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(67.28, 74.30, 68.43)$ out of a soil initially holding $(539.82, 647.66, 533.81)$, a fraction of $(0.12, 0.11, 0.13)$.

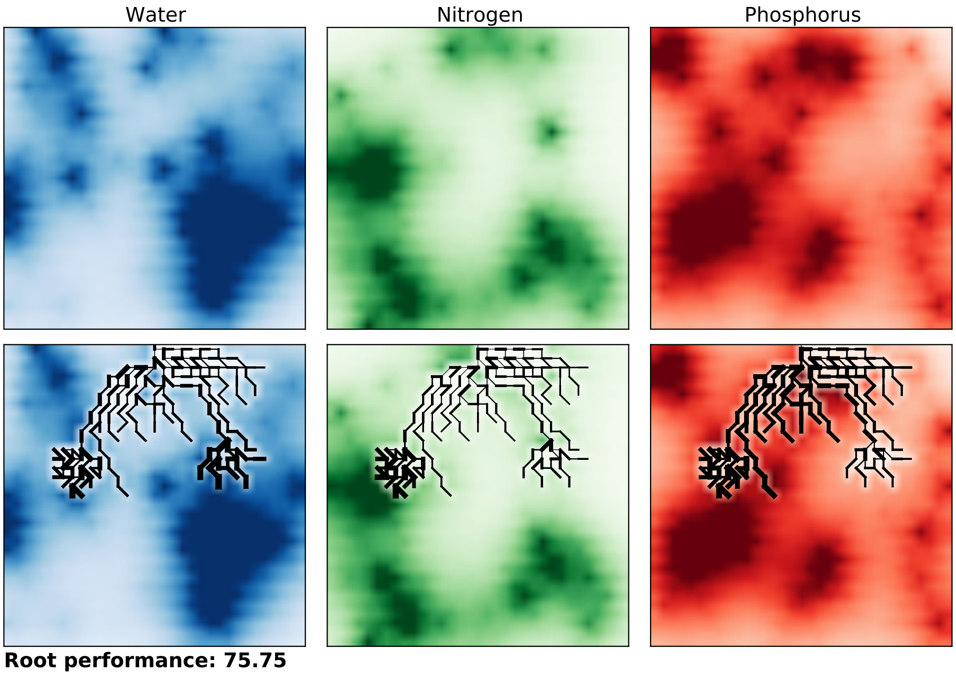


Figure 4.10: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(75.86, 61.79, 89.59)$ out of a soil initially holding $(576.32, 434.68, 639.24)$, a fraction of $(0.13, 0.14, 0.14)$.

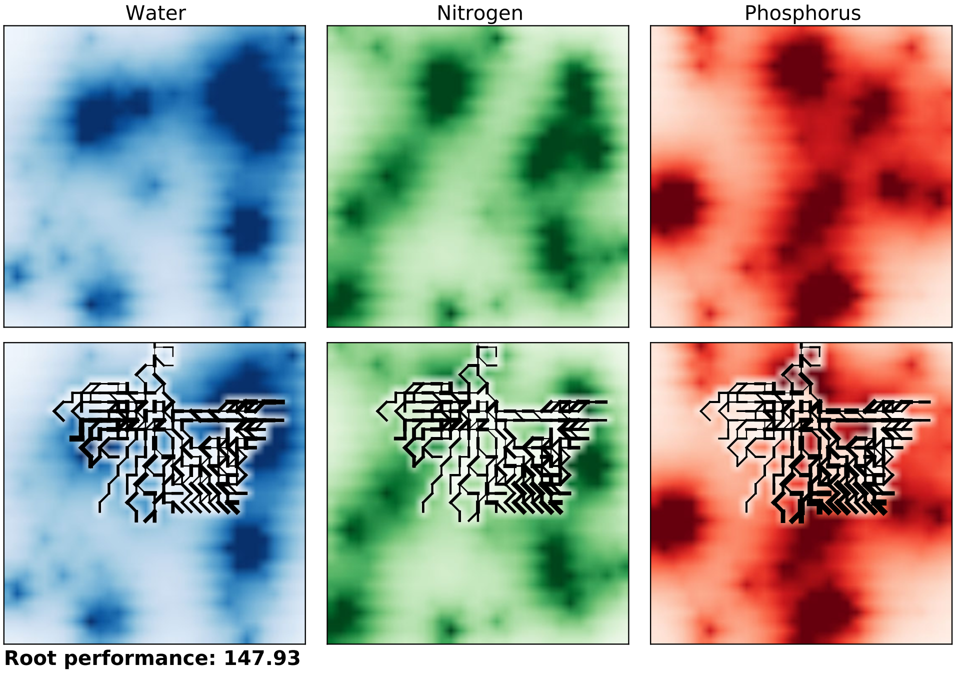


Figure 4.11: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(148.67, 144.07, 151.06)$ out of a soil initially holding $(549.02, 559.70, 608.57)$, a fraction of $(0.27, 0.26, 0.25)$.

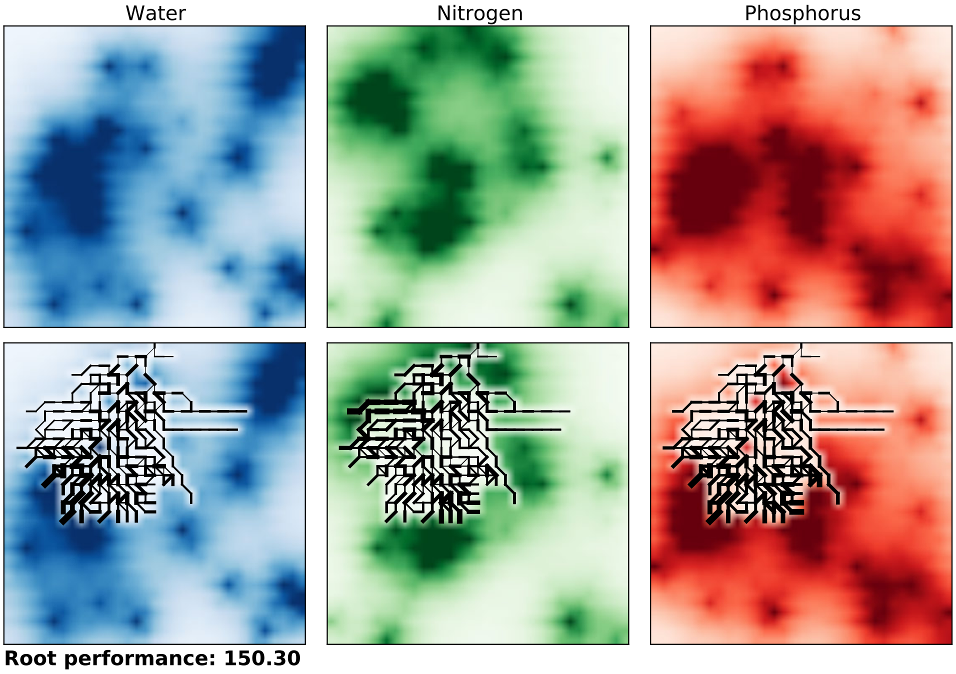


Figure 4.12: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(141.47, 153.93, 155.49)$ out of a soil initially holding $(571.63, 475.38, 620.46)$, a fraction of $(0.25, 0.32, 0.25)$.

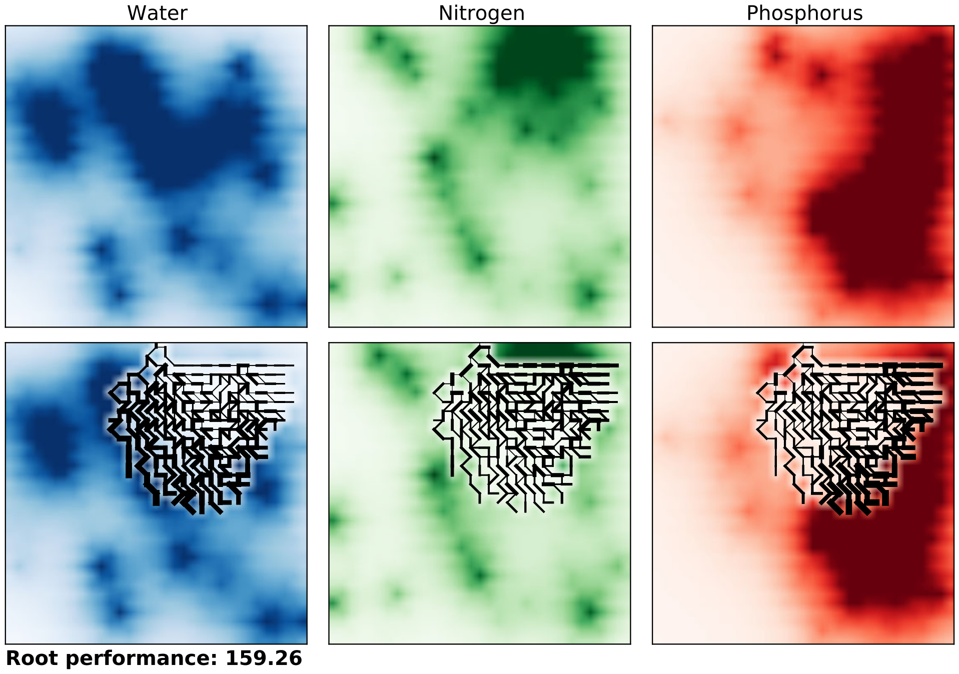


Figure 4.13: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(171.60, 146.51, 159.67)$ out of a soil initially holding $(640.96, 394.91, 538.43)$, a fraction of $(0.27, 0.37, 0.30)$.

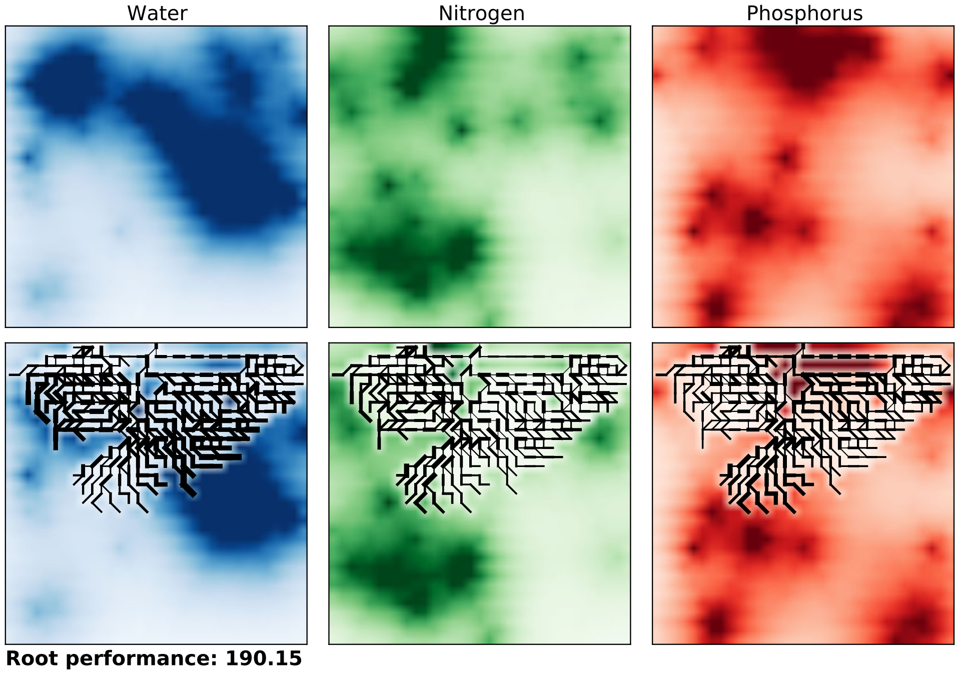


Figure 4.14: A root growth simulation at times $t = 0$ and $t = 200$. The root extracted $(222.29, 167.12, 181.04)$ out of a soil initially holding $(573.14, 495.86, 541.69)$, a fraction of $(0.39, 0.34, 0.33)$.

Chapter 5

Sensor Web deployment in unknown environments

In a paper from 2001 [Delin and Jackson, 2001] a new concept for an instrument capable to explore unknown environments is introduced: the sensor web. In the original words of the authors "*The confluence of the rapidly expanding sensor, computation, and telecommunication industries has allowed for a new instrument concept: the Sensor Web. A Sensor Web consists of intra-communicating, spatially-distributed sensor pods that are deployed to monitor and explore environments. It is capable of automated reasoning for it can perform intelligent autonomous operations in uncertain environments, respond to changing environmental conditions, and carry out automated diagnosis and recovery*". The fundamental difference between this concept and a distributed sensor network is the presence of direct communication between sensors (commonly called Pods) [Delin, 2005] and the use of the sensing information to reconfigure continuously the network tasks. Other popular words describing similar concepts are, thus, intelligent distributed sensor networks, reconfigurable sensor networks and so on.

5.1 Task definition

The collective robotic task considered in this project consists in the deployment of a sensor web. We consider a large number of simple robots (or Pods) with limited capabilities, that must self-deploy from a central location and operate into a previously unknown terrain as to form a sensor web. Decision making should be fully decentralized and based on the environment perceptual clues. An homogeneous collection of Pods is assumed, where all may therefore assume all available role types. Collective performance should be resilient to individual failures.

Once deployed, the Pods are assumed to remain at their location for some time, conducting whatever scientific work they are tasked with. Once

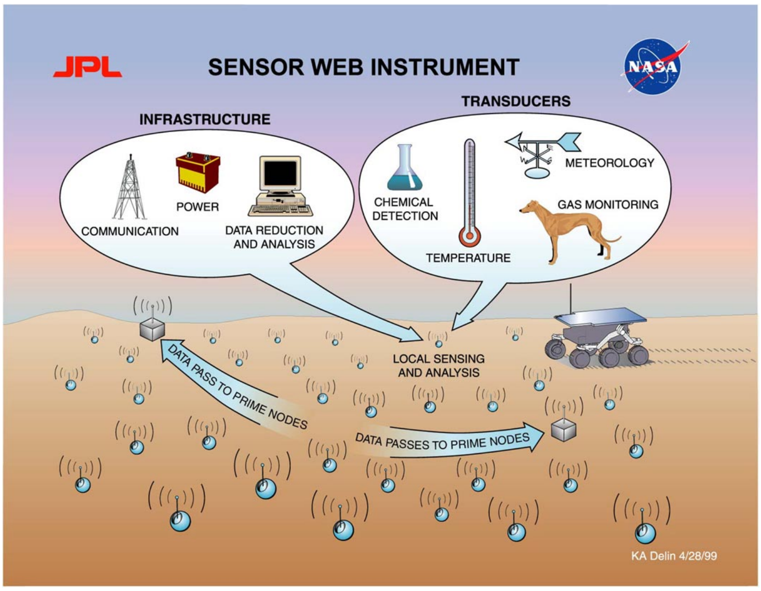


Figure 5.1: One visualization of the sensor web concept, as shown in [Delin and Jackson, 2001]. The rover was claimed to be in the image only to give the scale to an otherwise in-situ terrestrial application

an assignment is concluded, the swarm is then free to regroup, collectively move into another area of the planetary surface, and deploy again. This process is repeated over the duration of the mission.

The tasks robots will perform might include for instance the foraging for chemical or biological traces, monitoring of conditions over a period of time, and the temporary formation of specialized antenna arrays. In this last example, the swarm would be functioning as an antenna, and for that it would need to self-organize into a configuration with maximal performance with respect to local conditions affecting the signal, or to the relative positions of the satellites they'd be communicating with.

The work described here then focusses on the definition of a communications protocol and a decision-making mechanism for autonomous deployment with a view to maximizing collective performance simultaneously with respect to a set of multiple goals.

5.2 From ROOTS to RObOTS

The robotic swarm's equivalent of the root apex is here designated as a Mob. **Mobs** are groups of Pods that collectively move into a common target. When they reach that target, the mob perceives the local and surrounding conditions with regards to the set of criteria influencing its behaviour, and takes a decision about what to do next. The decisions we make available to Mobs map exactly to the decisions apices have at their disposal in the roots simulation: *StayStill*, *Elongate*, and *ElongateAndBranch*. These correspond respectively to a Mob remaining at a location for some time, collectively moving into a new target, or splitting in half, and having the resulting two new Mobs moving into their own directions. As an example imagine a sensor web consisting of $N = 100$ initially colocated Pods. We imagine they are all delivered on an unknown planetary surface and left in the same area (i.e. the landing site of a mother spacecraft). The deployment then starts and the Mob ($N = 100$) starts taking a decision and a) move to another area, b) branch itself into two Mobs and deploy, say, one Pod (the two Mobs will then contain $N = 50$ and $N = 49$ Pods) or c) just wait. The process then repeats up to when all Pods ar deployed and the "root system" is formed. The decision taken by each Mob takes into account the global exploitation of multiple criteria (resources) as sensed by the different Mobs. This decision is taken by the exact same controller evolved in chapter 4 to model the growth of a root system.

All robots have a unique numerical identifier assigned to them, and they are always made aware of which other robots are moving together with themselves in the same Mob. By way of these identifiers, robots are able to autonomously take on their respective roles in the dynamics, at any given time. A parameterizable number of robots in a Mob (connected to the desired degree of redundancy), are recognized by all as the Mob's **leaders**. These are by definition the robots with the smallest numerical identifiers from among those in the mob.

Whenever a branching decision occurs, the Mob's leaders remain behind to conduct at that location the work they are tasked with performing, but also to serve from then on as communications relay units between those robots (leaders) that had previously remained behind, and the leaders of the two new Mobs now departing from that location. The hierarchy that thus emerges in the swarm is in a sense implicit in it from the beginning (it is a consequence of robots' numerical identifiers), but the way it is instantiated on the surface is a consequence of the conditions robots encounter as they deploy, and the decisions they take as a result.

When moving from a root growth into a swarm deployment problem, one has to take into account the fact that a robot is a scarcer resource than a root segment. Roots can grow an unlimited number of root segments, being constrained only by the amount of available materials. Once grown, a root segment will from then on reside at a soil location with which it continuously interacts, and act as a communication channel between segments preceding and following from it. The swarm, on the other hand, must use its robots more sparingly as sensor webs are formed by a finite number of Pods. As such, the decision was taken to leave Mob leaders behind only when branching decisions occur, and never when the Mob's decision is to elongate.

Successive elongation decisions then result in more distant communication channels being formed. This is a departure from the roots simulator, where communicating entities always explore contiguous soil patches.

A further significant difference between the roots simulator and the deploying sensor web, is that while in the first the soil is physically altered by the roots' activity, with those changes constituting an important stigmergic communications channel with an impact on apices' behaviour, in the second robots will hardly be given the capability to permanently affect the surface in ways other robots can interpret and act upon. In a Moon mission where tracks left by robots persist in the soil, the visual perception of the amount of tracks in an area might provide robots with this informational cue. However, given we wish not to constrain the applicability of the system described here, we prescribe that robots in our swarm will autonomously build and use a *difference map*, which they continuously update based on their actions, and by merging with the difference maps of the robots they communicate with. This difference map biases perception of conditions in robots' surroundings. An area that by itself is perceived as being extremely *interesting*, will be ignored if as a robot approaches it its difference map informs it that others have previously explored the area extensively, just as an apex will ignore a soil area that was once rich in materials, but was since depleted by other root segments. The problem of distributed map building by robot collectives is a standard research problem out of the scope of this work, and was therefore not studied here. For simplicity, in our robotic simulation a single centralized difference map was thus used.

5.3 Communication and decision making in the Robots Swarm

The flow of information in the robots swarm is analogous to the diffusion of water and nutrients inside a root. Each root segment diffuses water and nutrients with those segments contiguous with it in the root structure. Matter extracted from the soil thus spreads from locations of high concentration to the parts of the communication channel where they are scarce. In the robotics simulation, "extraction" translates to the generation of signals taking into account the base values robots perceive in their vicinity on the multiple variables they measure. These signals are influenced by (and in turn influence) the difference map mentioned in the previous section, so as to emulate permanent extraction of the currently available materials in the surrounding soil. Signals are then sent through the communication channel robots establish. Deployed robots, along with moving mob leaders are thus continually participating in processes of resource extraction and internal diffusion.

When a mob reaches its chosen target, it perceives its surroundings.

This can either be achieved as a fusion of the sensor values obtained from all robots in the mob, or simply as a selection of the values perceived by a single robot, such as a mob leader. Having the mob produced a single perception vector, and given their commonly shared deterministic controller, each robot in the mob could then separately reach the same conclusion on what the appropriate course of action would be. Because in the roots approach behaviour is conditioned also by values diffused through the swarm, a process on which only mob leaders participate (as mentioned in the previous paragraph), decision making in the mob was thus restricted to the leaders, which already possess the required information. In our current implementation, mob leaders use the values they perceive in their surroundings, with no fusion with values from remaining robots in the mob taking place. In future work sensor averaging could be implemented, as it can eventually contribute to more reliable decision-making in the context of noisy sensory information, or sensor failure.

Whenever a common decision must be reached in the mob, the mob then produces a common perception vector, the mob leaders use it to decide on the appropriate course of action, given their behavioural controllers, and broadcast it through the mob.

5.4 Transfer of apex controllers

The degree of similarity between the root growth and swarm deployment problems is such that the controllers previously trained for the apices can be reused to control the robots with no modifications. This section provides the correspondence between apices' and robots' perception vectors, and discusses issues that arise from such reuse of the controllers.

5.4.1 Mapping of perceived variables

The three variables modeled in the soil and root simulation (Water, Nitrogen and Phosphorus) represent resources the root needs to acquire, in amounts that should preferably be as high as possible. In the swarm deployment problem, these translate into measurements of characteristics in the planetary surface that are of scientific interest to the mission. The swarm has to locate in the surface as many interesting areas as possible, having as high interest values as possible. As in the case of the root, the robots swarm is also pursuing multiple objectives at the same time. It is therefore not acceptable for the swarm to focus on the satisfaction of one goal at the expense of the remaining ones.

Robots generate perception vectors matching the syntax that was described in Section 4.1 for the root growth simulation. Sets of values in all the scientific interest criteria are measured internally in the Mob's communication channel, in the surface area in which the Mob finds itself, and then in six directions around the Mob, evenly spaced and spanning the range of 360°. The values characterizing scientific interest in a given direction summarize the quality of a patch of land, as observable from the Mob's current location, and up to a distance consistent with robots' sensors' capabilities. A movement decision takes a Mob from one location into the centre of a patch of land lying in one of those six directions. The hexagonal structure imposed on the soil in the root simulation is thus also imposed on the planetary surface.

5.4.2 Challenges in controllers' transfer from roots to robots

Given a perception vector characterizing a robot's context, action selection takes place by feeding it through an apex controller. The differences that do exist between the two application scenarios, root growth and swarm deployment, result in some of the controllers' behaviours to be less suitable for the domain they are transferred to. These challenges in the controllers' transfer are described next.

Gravity bias

Gravitropism, in biology, is the tendency displayed by roots to grow in the direction of gravitational pull (i.e., downward), and for stems to grow in the opposite direction (i.e., upwards).

In our modeling of root growth gravitropism was not implemented through any explicit mechanism. Its effect is though present in an implicit form. It arises out of a combination of the model's tuning, and the pressure in the fitness function for maximizing the extraction of materials. Upwards root growth is most of the time disadvantageous to the apex in the short term, as most of the time there are already root segments above, depleting the soil resources. Given the gradual variations in resource availability over contiguous soil patches, the depletion of resources often makes the unexplored patches more attractive, resulting in a tendency for apices to move as a wave away from already occupied locations. Besides that, by tuning in the simulator *narrow* soil dimensions, and a simulation time big enough for roots to be capable of reaching the edges of the simulated soil, we created conditions whereby the investment in lateral root growth becomes disadvantageous in the long term. Investing more in moving sideways or upwards, apices would at some point in the simulation reach the edges of the simulated soil, from which they had nowhere else to move unless they were to go through the inefficient process of growing segments through soil patches already occupied by other root segments, so as to reach unvisited soil areas. Those apices could then no longer efficiently help the root in extracting more materials, and thus increase fitness. By spending instead those resources required for growth in downwards movements, an apex would have a greater chance of contributing to a more significant increase in the root's efficiency, just because in that direction the segment would have access to a greater area of soil. Taken together, evolution was thus led to produce controllers with a tendency for downwards movement, in effect reproducing the effect of gravitropism. Evolution was able to implement into controllers this tendency for following the gravity gradient through an assignment of greater importance to values in the perception vector coming from neighbouring directions with indexes 2 (south-east) and 3 (south-west), than to values coming from the remaining directions.

Though biologically accurate, when transferred into a problem of deployment on a planetary surface, this behaviour leads the swarm to have a preferable movement direction, and therefore to potentially fail in the identification of interesting areas not along that direction. A way to address this deficiency is to see soil in the root simulator not as a depth-cut, but instead as a top-down view, to place the seed in the centre of the soil, rather than at the top, and to re-optimize controllers. During simulation, apices are then exposed to a greater diversity of conditions, having optimal directions for growth more uniformly spread around the apex. Experiments performed with this configuration were conducted, producing the desired results. Those experiments are not reported here, first due to an intention to keep the consistency with the biological model, and second because a preferable movement direction provides a potentially useful way to externally influence the swarm's behaviour.

By changing the way the perception vector is built, and adjusting the decoding of growth directions coming out of controllers' selected actions, one can adjust the swarm's preferable movement direction. As explained in Section 4.1, north-east is the first neighbouring direction to be perceived (index 0), with the remaining directions being perceived in clockwise order up to north-west (index 5). The obtained controllers consider more heavily the values of variables perceived in directions 2 and 3 (followed by 1 and 4). By rotating the order in which directions are perceived, whichever directions end up with indexes 2 and 3, will get greater consideration. As an example, if robots were to start by perceiving south-west, then rotate clockwise until reaching south-east, the swarm's preferred direction for movement would then be north-west (the new direction 2) and north-east (direction 3).

Excessive branching bias

In the roots simulation, the more apices branch (provided they have the materials to pay for growth's energetic costs) the greater will the root's coverage of the soil end up to be, thus resulting in a greater performance value. However, in the robotics application, an excessive number of robots would then be required to replicate the same behaviour.

To successfully transfer the apex controllers, we set robots to cancel branching orders (though keeping the elongation order intact) when they are present in very poor areas, in terms of having low scientific interest values. Branching, with its depletion of number of undeployed robots, was thus only allowed to occur when a bigger gain was to be had.

5.5 Robotics simulator

Figures 5.2-5.5 show top-down views of swarm deployments on distinct planetary surfaces, at different points in time, always using as behavioural controller the best Neural Network evolved in the roots simulation. The white trails show the paths taken by robots during their exploration of the surface. The three panels to the left provide different perspectives on the arena in which the robots are being deployed. They display the scientific interest of different surface areas, as evaluated according to three different criteria. Color intensities are tied to the degree of an area's scientific interest (in that criterion), with darker areas representing lower scientific interest. As in the roots model, this information becomes accessible to robots only when they reach an area's vicinity, and decisions are thus taken based on local information. As the swarm deploys, it builds the surface quality difference map, which emulates the resource extraction performed by roots. The swarm's gradual decreasing of the quality of areas on which robots have already deployed is perceived by others, and their behaviour self-adjusted accordingly. This way, the swarm scatters itself through the surface, instead of clustering all around the most interesting of the identified surface areas, continuously exploring as long as it has enough non-deployed units capable of doing so.

A degree of gaussian noise is added in these simulations to robots' actuators, something that was not present in the roots simulations. The individual robots in a Mob share a common goal, but pursue it by their own means, subjected distinct constraints on movement. A robot's step towards a target is subjected to random variations in direction and speed. The cumulative effect is that robots in a Mob take different amounts of time to reach the target, and reach it by taking slightly different paths. Because only Mob leaders decide on the next action to be performed, the remaining robots in the Mob wait at their current target until their leaders reach the same location, perceive their context, and broadcast among the Mob their decision of the action they should collectively perform next.

In figs. 5.2 to 5.5 we depict a few examples of the robot swarm model over different conditions. Figures 5.2 and 5.3 show the swarm deployment on the same scenario, but using a different number of robots. There are 256 robots in the first swarm, and 1024 in the second. The four frames in each picture represent four stages of the swarm development, with similar percentage of robots deployed in each case. We notice that both swarms show a similar shape, adapting to the scenario. The larger swarm basically expands and intensifies the deployment observed in the smaller swarm.

In the second set of figures (5.4 and 5.5) we compare the swarm behavior in different scenarios. In both cases the swarm size is 256. The root model deployment clearly shows that the robots tend to explore regions of higher interest concentration, towards the right in fig. 5.4 and towards the left in fig. 5.5. Also compare these two deployments with the one of 5.2.

Notice that each swarm tends to maximize exploration of the multiobjective problem consisting on three components represented on the left side of the frames by the red, green and blue colors. These components constitute different scientific interest criteria that a robot swarm must take into account. They do not have exactly the same geographical distribution, and therefore the swarm does not exactly overlay any of the criteria in particular, but tries to maximise them together. These scenarios were generated randomly in order to illustrate the swarm's dynamics under different environment conditions.

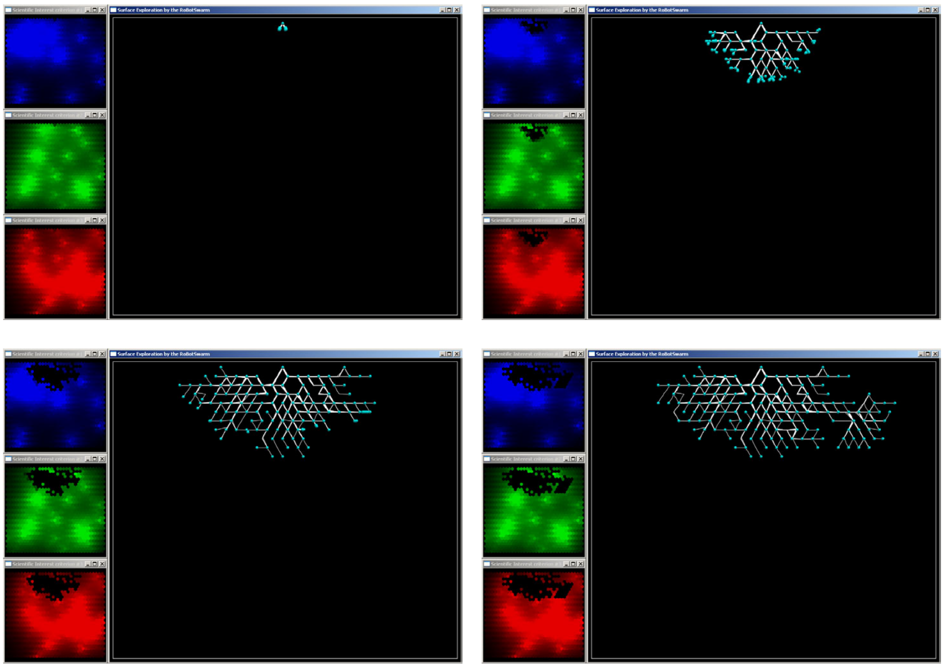

Scientific Interest criterion #1
Scientific Interest criterion #2
Scientific Interest criterion #3
Surface Exploration by the RoBotSwarm

Figure 5.2: Sensor Web deployment on scenario A (256 robots)

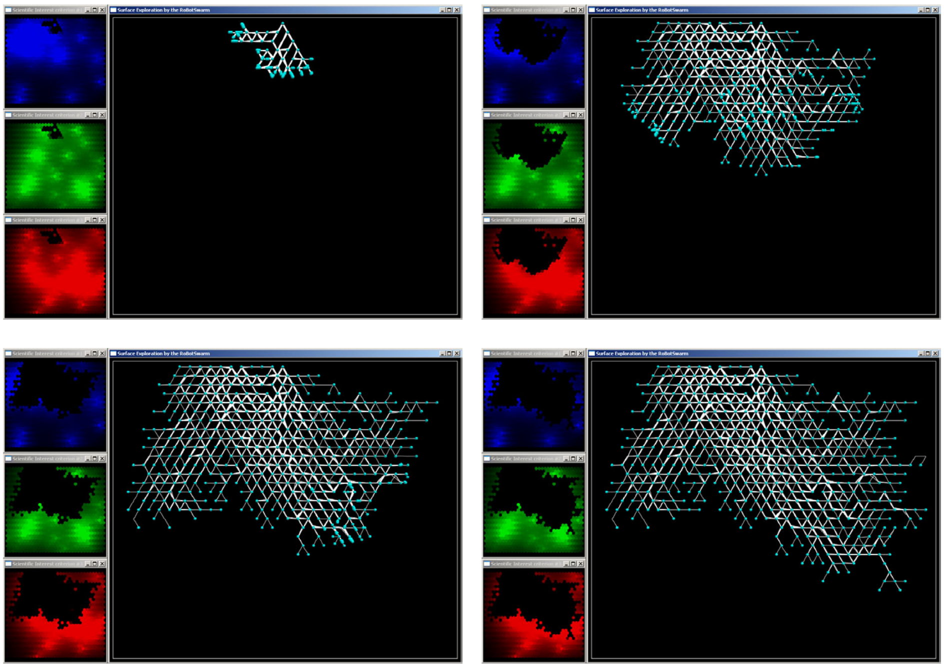

Scientific Interest criterion #1
Scientific Interest criterion #2
Scientific Interest criterion #3
Surface Exploration by the RoBotSwarm

Figure 5.3: Sensor Web deployment on scenario A (1024 robots)

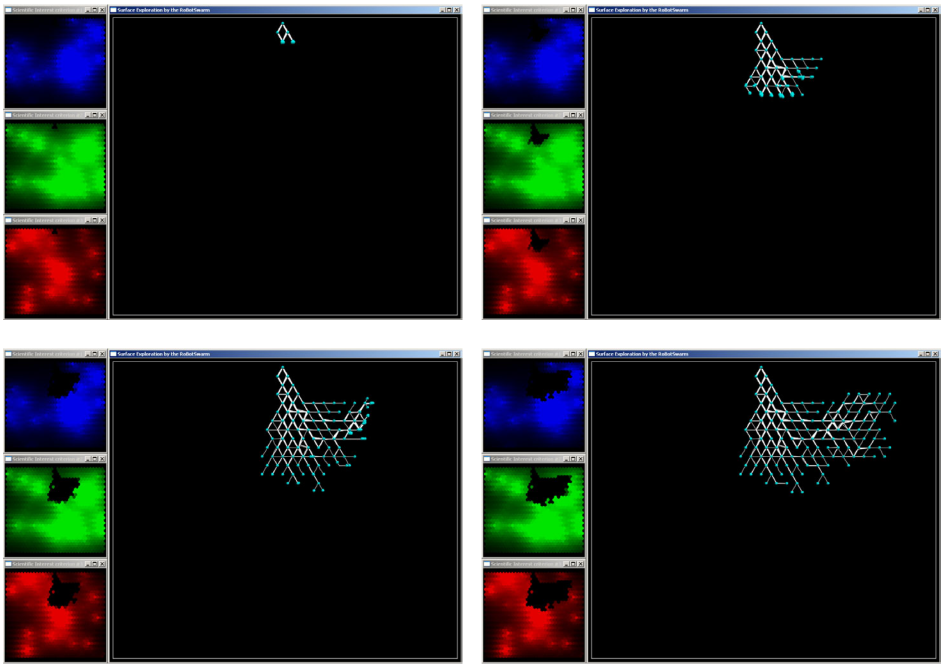


Figure 5.4: Sensor Web deployment on scenario B (256 robots)

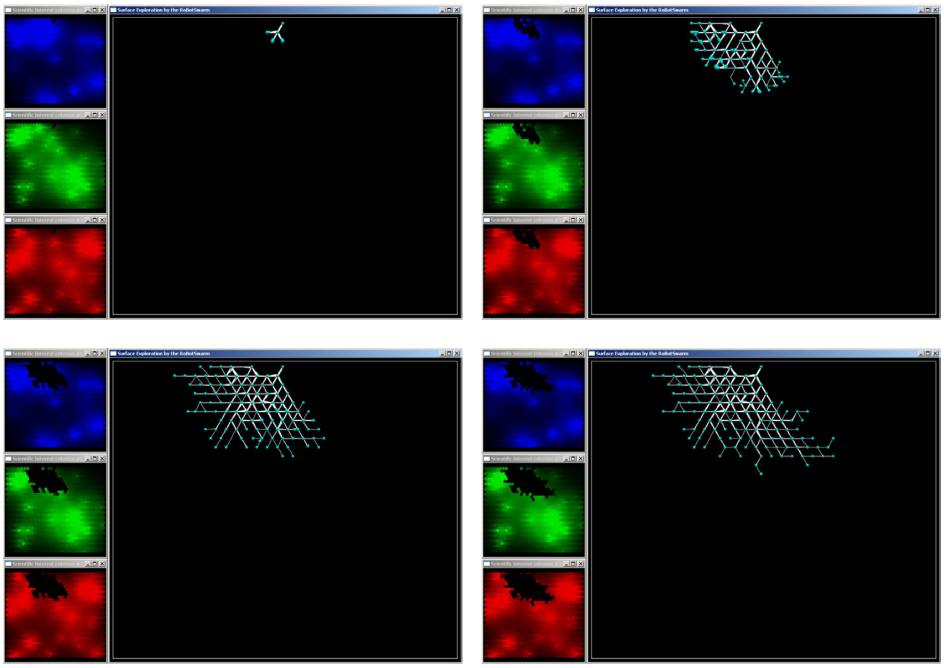


Figure 5.5: Sensor Web deployment on scenario C (256 robots)

Chapter 6

Conclusions

In this study we propose for the first time roots as a source of biological inspiration for the design of the decentralised control system of a swarm of robots performing a collective exploration task.

We have located a number of good reasons to consider roots as a biological paradigm for robotic distributed exploration. Some of them are:

- Roots exhibit developmental plasticity; genetically identical plants take different "decisions" if they grow in a different soil. This is the case for autonomous robotic swarms that must shape and adapt their behaviour according to the environmental conditions.
- The sensing mechanisms are located at the root apices. This can be directly translated to the type of distributed sensing we encounter in swarm robotic systems.
- The task of a root system can be seen as a solution to a complex multi-objective optimisation problem; the root has to balance multiple constraints and satisfy several needs (Water, Nitrogen, Phosphorus, ...). This fact is in accordance with complex robotics applications where multiple goals have to be achieved and multiple tasks have to be completed either sequentially or at the same time.

The hypothesis we embrace is that root systems do possess an intelligence that functions in a very similar manner to what is today commonly referred to as 'swarm intelligence'. They need this intelligence in order to exploit soil resources and maximize the plant's chances of survival. After a thorough literature review on the state of knowledge botany has accumulated on the dynamics of root composition and growth, we selected the herbaceous type of plants (and root) as our candidate to successfully inspire a robotic strategy, and we modeled this type of root growth in a cellular automata soil simulator. We then algorithmically designed the "intelligence" of each root apex using a neural network evolved to cope with different environmental conditions modeled in our soil simulator and we used the same evolved neural network for a selected robotic task (i.e., the deployment of a sensor web), mapping “one-to-one” our simulated root growth behaviour to the robot deployment. This task can be considered part of a larger scenario that includes a swarm of Pods being deposited at a central place on a planetary surface, and which has to deploy in an efficient (by way of scientific exploitation) way on the surface. Another possible task could be that of acquiring a formation during a descent phase, after being ejected by a mother spacecraft.

Our work is using a novel inspirational paradigm from nature, root systems, to design the control system of a group of autonomous agents set to explore an unknown planetary surface. We must note that for the time being the possibility of planetary exploration by swarms of simple agents that collectively manage to perform complex tasks is still far in the future; actual rovers are typically very complex machines that operate in a solitary manner. However, should multi-agent systems be considered for exploration tasks, our work could serve as a simple solution. Also, many other tasks can be mapped to the growth of a root system as modeled here and we are just beginning to understand which ones would benefit the most from this new paradigm.

The second contribution of our research is on the biology of root growth. We have successfully modeled roots as an ensemble of autonomous agents (apices) which is able to adapt and regulate its morphology in order to optimally exploit its environment. To this end, we have designed a simulator which takes into account the most important biological concepts. This simulator, coupled with evolutionary optimisation techniques can prove to be a valuable tool to conduct in depth studies aiming to shed light on root behaviour and biological questions that can hardly be answered with other tools.

Appendix A

Root structure: implementation details

Our representation of the root architecture shares similarities with the Extensible Tree structure defined in [Lynch et al., 1997]. A major difference between both is that in the Extensible Tree each root segment supports an arbitrary number of children, while our structure limits the branching degree to 2 (as previously mentioned). A shared similarity, though, is to be found in the usage of memory pointers to maintain different lists relating multiple root segments, so as to reduce the complexity of several operations performed on the root. These lists are implemented in addition to the previously mentioned edges, which define adjacency between segments in the root system.

Each root segment contains a pointer to the structure representing the soil patch it is embedded on, and that soil patch in turn contains a pointer to the first segment in a singly-linked list of all root segments embedded in that soil patch. Each root segment contains additionally in its structure a pointer to the following segment in that list. When a new root segment grows into a soil patch, it adds itself to the head of this list, and sets its list pointer to the previous first root segment in the list. This computationally cheap process guarantees that we can at any time efficiently access all the root segments in any given soil patch. That information is needed for the implementation of soil→root diffusion (Section 3.3.3).

Each root segment encodes two additional pointers, which enable the structure to potentially be a member of a doubly-linked circular list of all the apices in the root. In other words, we have at all times a list of the binary tree's leaves. Non-apex root segments make no use of these pointers. When an apex grows one or two new segments, it ceases to be considered an apex. At that point, it removes itself from the list, and inserts its children segments (apices themselves) in its place in the list. The importance of this list lies in its enabling of efficient traversal through only those segments in the root which have the property of being apices, without the need to pass through intermediate root segments. Given the distinct roles assigned to root apices (Section 3.3.3), this makes their processing more efficient.

Finally, each root segment encodes also a state counter, which allows for non-recursive stackless depth-first traversal[1] through all segments in the root system. Multiple processes in our simulation requiring iteration over all root segments, such the as the root→root diffusion process (Section 3.3.3), can this way benefit from having access to an iterator that is efficient in terms of speed and memory.

[1] We refer to depth-first traversal not in the sense of soil depth, but in the traditional sense of path length from the binary tree's root (the shoot segment).

Appendix B

Diffusion process: algorithmic definition

Diffusion is implemented as a flow of materials from one *donor* into a set of *recipients*. The donor defines how much it will deduct from its own state variable, and how that amount will be distributed between the recipients. If a recipient has a higher amount on the diffusing variable than the donor, the donor takes no action towards it. That recipient, on the other hand, might donate part of its value to the current donor at the time the diffusion process is invoked with it as the donor.

All simulated entities (soil patches and root segments) assume the role of donor every time step, at which point they potentially diffuse into their neighbours. Diffusion takes into account the current states of all the participants in the process, and increments values at those entities' variables corresponding to their state at the next time step. A great deal of an entity's state at the next time step is thus written not by itself but by its neighbours. Note that every simulated entity participates as a donor in multiple diffusion processes, one for each of its state variables (Water, Nitrogen and Phosphorus).

Once all entities have taken their turn being donors, they all switch the state variables representing their current and next time steps, thus advancing the world's clock by one iteration. As they do so, the state variables now corresponding to next time step are reset to 0.0, so as to initialize their values for the new iteration's diffusion processes. These separate phases of state and clock updating, coupled with the commutativity in the order of state updates, implements a synchronous updating scheme, through an asynchronous process. From the model's perspective, all entities update their state at the same time.

Table B.1 describes the sets of variables needed for a full characterization of all participants in a diffusion process. A set of variables p is present in all participants, be they donors or recipients. Their distinct nature does

diffusion participant: $p = \{s_m, s_t, s_{t+1}, d\}$

s_m	Amount assumed to be contained on the diffusion participant at the current time step, if the participant is missing from the simulation (if it is not an explicitly simulated entity).
s_t	The diffusion participant's current state.
s_{t+1}	The diffusion participant's state at the next time step.
d	Amount diffused from/into this participant in the current call to the diffusion procedure.

donor: $d = p \cup \{k, o_{max}\}$

k	Minimum amount kept by the donor. Only amounts in excess of this value will be available for diffusion.
o_{max}	Maximum amount this donor is capable of diffusing out of itself per time step.

recipients: $r = p \cup \{c, i_{max}\}, R = \{r^1, r^2, \ldots, r^n\}$

c	The recipient's holding capacity.
i_{max}	Maximum amount this recipient is capable of receiving into itself per time step, from a single donor.

Table B.1: Characterization of the variables defining participants in a diffusion process. A call to DIFFUSE(d, R) writes each participant's $p_{s_{t+1}}$ and p_d variables. All other variables are taken only as inputs.

however cause them to require a set of additional variables, characteristic of their role. Some of these variables are set to the system-wide parameters that define that specific diffusion process. Prior to a call to DIFFUSE(d, R), each participant is then wrapped into a structure where several variables are already set, and the remaining ones are either read or dynamically calculated from the participant's current context. Implementation-wise, p_{s_t} and $p_{s_{t+1}}$ are handled as pointers into the memory locations where one of the entity's state variables lies. By setting these pointers, the wrapping structure is made aware of the particular entity it will represent in the diffusion process.

We next present in detail the DIFFUSE(d, R) procedure that implements the diffusion process.

Algorithm 1 Get donor's state and amount available for diffusion

```
1: procedure DIFFUSE($d$, $R$)
2:     if $d$ exists in the model then
3:         $d_{\tilde{s}_t} \leftarrow d_{s_t}$
4:     else
5:         $d_{\tilde{s}_t} \leftarrow d_{s_m}$
6:     end if
7:     $d_d \leftarrow min(max(0.0, d_{\tilde{s}_t} - d_k), d_{o_{max}})$
```

Lines 2–6: We start by loading $d_{\tilde{s}_t}$, the donor's current state. For most cases, this value is read from d_{s_t}. In cases where the donor is not an explicitly simulated entity, the value is read from d_{s_m}. As examples of this situation, we might consider Carbon diffusing from the shoot into the root, or rain water falling into the first layer of soil.

Line 7: d_d is initialized to the maximum amount this donor might potentially diffuse out in the current time step. From the total in $d_{\tilde{s}_t}$, up to some amount d_k will not be available for diffusion, as the donor keeps it for himself. As an example, consider water retention by a soil patch, that is capable of diffusing out only when water content exceeds a certain degree of saturation. Physical limitations will at the other end impose an upper bound $d_{o_{max}}$ on the amount the donor can diffuse out of itself in the time span corresponding to the transition between consecutive time steps.

If the amount in the donor at the diffusing state variable is such that diffusion out of it is possible, the next step is then to determine how much each recipient would individually require from that amount, so that over time all donor–recipient pairs would potentially converge on the same amount.

Lines 10–14: The considered recipient's current state $r_{\tilde{s}_t}$ is read in the same way as the donor's current state was read in lines 2–6. Recipient entities not explicitly modeled in the simulation are therefore also supported. As examples of such situations we might consider the diffusion of materials extracted by the root into the shoot, or water flowing below the last simulated layer of soil.

Lines 15–16: Diffusion occurs in the direction of greater to lower concentration. If a recipient has a higher amount on the diffusing variable than the donor, the donor takes no action towards it. This procedure implements diffusion only out of the donor and into the recipient. A

Algorithm 2 Determine amounts requested by recipients

```
 8:  if $d_d > 0.0$ then
 9:      for all $r \in R$ do
10:          if $r$ exists in the model then
11:              $r_{\tilde{s_t}} \leftarrow r_{s_t}$
12:          else
13:              $r_{\tilde{s_t}} \leftarrow r_{s_m}$
14:          end if
15:          if $d_{\tilde{s_t}} \leq r_{\tilde{s_t}}$ then
16:              $r_d \leftarrow 0.0$
17:          else
18:              $r_d \leftarrow min(\frac{d_{\tilde{s_t}} - r_{\tilde{s_t}}}{2}, r_{i_{max}})$
19:              if $r_{\tilde{s_t}} + r_d > r_c$ then
20:                  $r_d \leftarrow max(0.0, r_c - r_{\tilde{s_t}})$
21:              end if
22:          end if
23:      end for
24:      $d_d \leftarrow min(d_d, \sum_{i=1}^{|R|} r_d^i)$
25:  end if
```

flow in the opposite direction will take place when diffusion is executed with that recipient assuming the role of donor, but only if neighbourhoods are symmetrical for a given state variable's diffusion involving entities of those types (Phosphorus diffuses from the soil into the root, but not the other way; in between neighbouring root segments however, Phosphorus flows in both directions). Note we consider in the comparison not d_d, the actual amount the donor is capable of diffusing, but $d_{\tilde{s_t}}$, the complete amount if holds.

Lines 17–18: Given the donor has an higher amount than the recipient, we start the process for determining how much the donor will allocate to this specific recipient. The implemented strategy tries to allocate an amount proportional to the difference between both entities' amounts, so that the greater the difference, the more will be diffused, and then have the diffusion rate decrease over time as values equalize. This quantity allocation must then be constrained by $r_{i_{max}}$, the maximum amount this recipient is capable of receiving into itself per time step, from a single donor.

Lines 19–21: The amount diffused into a recipient should not cause it to exceed r_c, its holding capacity. However, this process is intentionally blind to how much will also diffuse into the recipient from its other donors, and how much it will retain once it donates its share. Under some circumstances, the recipient may therefore still accumulate more than it regularly would be able to. This independence between different diffusion processes is required for their order of execution to be commutative, and so for synchronous state updating to be achievable. Consider the deadlock conditions that would arise from having two entities, donors to a third, conditioning their diffused amount by the amount the other would diffuse. The explicit enforcement of strict holding capacity limits on diffusion participants would significantly increase the process' complexity, and is therefore not implemented. We instead rely on the fact that these excesses will self-correct over time, by the recipient's donations into its own neighbours. Such a decision is defensible from a biological perspective, given the existence of supersaturated states in nature. We do implement here instead the handling of individual diffusions that would by themselves cause this constraint to be violated. We will in those cases only allow diffusion of up to the amount needed to reach the recipient's holding capacity. The $max()$ function is used here precisely to take into account those potential situations where the recipient is already oversaturated ($r_{\tilde{s}_t} > r_c$). Diffusion into such a recipient is thus canceled.

Line 24: Should recipients have requested in total an amount below the maximum the donor could diffuse out of itself in the current time step, d_d would then be overwritten with that total. The donor would be able in such a situation to fully meet all requests from its recipients.

Having paired the donor in turn with each recipient, we now know which amounts would ideally diffuse into them in the current time step, so that time to state equalization between all entities would be minimized. Given the donor's potential incapacity to meet all of its recipients' requests, the time now comes to look at recipients as a whole, and to distribute between them d_d, the total diffusing out of the donor, so that each will receive in proportion to its need.

Lines 26–29: If $d_d = 0.0$, then either the recipients are not asking for any amounts, or the donor cannot currently diffuse anything out of itself. Either way, we inform recipients that they collected no amounts in the current diffusion.

Lines 30–33: The amount to be diffused must now be split between recipients, in a way that tries to satisfy as best as possible all their individual demands. r_d is then at this stage used to represent the ratio of the amount to be donated that would best match the recipient's request. Should the recipient request more than the total the donor can give, the ratio that would best satisfy the request is then set to 1.0. This bounding brings $d_{o_{max}}$ to bear on the amount to be diffused into the recipient.

Algorithm 3 Allocate and transfer diffusible amount to recipients

```
26:     if $d_d = 0.0$ then
27:         for all $r \in R$ do
28:             $r_d \leftarrow 0.0$
29:         end for
30:     else
31:         for all $r \in R$ do
32:             $r_d \leftarrow min(1.0, \frac{r_d}{d_d})$
33:         end for
34:         $ratios\_sum \leftarrow \sum_{i=1}^{|R|} r_d^i$
35:         for all $r \in R$ do
36:             $r_d \leftarrow \frac{r_d}{ratios\_sum} d_d$
37:             if $r$ exists in the model then
38:                 $r_{s_{t+1}} \leftarrow r_{s_{t+1}} + r_d$
39:             end if
40:         end for
41:     end if
```

Lines 34–36: Ratios of amounts requested by recipients are now normalized over the sum of all such ratios. We are then allocating per recipient the closest possible value to what it requested, that takes into account the remaining recipients' requests. The sum of normalized ratios over all recipients naturally equals 1.0, meaning the process is able to fully allocate the amount d_d to be distributed. Also significant is that the process never allocates to a recipient more than it requests. At this stage, r_d is then overwritten with the final amount to be diffused from the donor into that specific recipient.

Algorithm 4 Update the donor's state with the retained amount

```
42:     if $d$ exists in the model then
43:         $d_{s_{t+1}} \leftarrow d_{s_{t+1}} + d_{\tilde{s}_t} - d_d$
44:     end if
45: end procedure
```

Lines 37–39 & 42–44: Having all amounts p_d diffusing in and out of all participants been defined, we are now at the stage where diffusion can actually be carried out. As previously mentioned, state variables are read from p_{s_t} and written into $p_{s_{t+1}}$. Also, as explained concerning the update scheme, as the transition between time steps is carried out, state variables corresponding to the new $p_{s_{t+1}}$ are reset to 0.0. Entities then contribute to the definition of each other's state at the next time step by incrementing $p_{s_{t+1}}$ with the outcomes of all diffusions the entity participates in. In those situations where the participant is not being fully simulated, no state variable is updated (its current state wasn't really in the simulation to begin with, having been read from p_{s_m}). Still, upon return from the procedure, p_d will contain how much would have diffused from/into the entity. This value might then be collected for statistical or other purposes, such as the measurement of the amount of materials diffusing into the shoot.

Appendix C

Constructing the standard test set

We give here all the information needed to reconstruct, using the Python programming language, the set of 10^4 soils configurations we used as the standard test set to evaluate our controllers in Chapter 4.

The simulator of soil and root dynamics implements Marsaglia's 64-bit Xorshift Method [Press et al., 2007; Marsaglia, 2003] (configured with $a_1 = 21$, $a_2 = 35$, $a_3 = 4$) for random number generation. The code below creates a vector of 10^4 numbers, which are then used as the random number generator seeds for each of the 10^4 root growth simulations.

The seed used below to initialize the random number generator instance in Python was an arbitrary choice taken so we could hardcode a process that would generate every time the same sequence of 10^4 numbers.

```
rng = random.Random( 14523725371007509729L )
Seeds = numpy.array(
    [ rng.getrandbits(64) for i in xrange( 10**4 ) ],
    dtype=numpy.uint64 )
```

For reference, the configuration shown in Figure 3.3 was obtained by randomly defining each peak's coordinates, height and decay rate, in sequence, starting from the random number generator seed 15160848958779875793.

Bibliography

M. Ali, C. R. Jensen, V. O. Mogensen, M. N. Andersen, and I. E. Henson. Root signalling and osmotic adjustment during intermittent soil drying sustain grain yield of field grown wheat. *Field Crops Research*, 62(1): 35–52, 1999. ISSN 0378-4290. doi:10.1016/S0378-4290(99)00003-9.

Besma Ben Amor, Sonia Wirth, Francisco Merchan, Philippe Laporte, Yves d'Aubenton Carafa, Judith Hirsch, Alexis Maizel, Allison Mallory, Antoine Lucas, Jean Marc Deragon, Herve Vaucheret, Claude Thermes, and Martin Crespi. Novel long non-protein coding RNAs involved in Arabidopsis differentiation and stress responses. *Genome Research*, 19(1): 57–69, January 2009. doi:10.1101/gr.080275.108.

Christos Ampatzis, Elio Tuci, Vito Trianni, Anders Lyhne Christensen, and Marco Dorigo. Evolving Self-Assembly in Autonomous Homogeneous Robots: Experiments with Two Physical Robots. *Artificial Life*, 15(4): 465–484, 2009. doi:10.1162/artl.2009.Ampatzis.013.

Steve Atkinson and Paul Williams. Quorum sensing and social networking in the microbial world. *Journal of The Royal Society Interface*, 6(40): 959–978, November 2009. doi:10.1098/rsif.2009.0203.

Mark A. Bacon, William J. Davies, Darren Mingo, and Sally Wilkinson. Root Signals. In Yoav Waisel, Amram Eshel, and Uzi Kafkafi, editors, *Plant Roots: The Hidden Half*, chapter 28, pages 689–704. Marcel Dekker, New York, third edition, 2002. ISBN 0-8247-0631-5.

Dayakar V. Badri and Jorge M. Vivanco. Regulation and function of root exudates. *Plant, Cell & Environment*, 32(6):666–681, June 2009. doi:10.1111/j.1365-3040.2009.01926.x.

František Baluška, Stefano Mancuso, Dieter Volkmann, and Peter Barlow. Root apices as plant command centres: the unique 'brain-like' status of the root apex transition zone. *Biologia, Bratislava*, 59(Suppl. 13):7–19, 2004. ISSN 0006-3088.

G. M. Berntson. Modelling root architecture: are there tradeoffs between efficiency and potential of resource acquisition? *New Phytologist*, 127(3): 483–493, June 1994. doi:10.1111/j.1469-8137.1994.tb03966.x.

Meredith L. Biedrzycki, Tafari A. Jilany, Susan A. Dudley, and Harsh P. Bais. Root exudates mediate kin recognition in plants. *Communicative & integrative biology*, 3(1):28–35, January 2010. ISSN 1942-0889. doi:10.4161/cib.3.1.10118.

Ragan M. Callaway and Bruce E. Mahall. Family roots. *Nature*, 448(7150): 145–147, July 2007. ISSN 0028-0836. doi:10.1038/448145a.

Anthony Carlisle and Gerry Dozier. Tracking changing extrema with adaptive particle swarm optimizer. In *Proceedings of the 5th Biannual World Automation Congress*, volume 13, pages 265–270, June 2002. doi:10.1109/WAC.2002.1049555.

Maurice Clerc. *Particle Swarm Optimization*. ISTE Ltd, London, 2006. ISBN 978-1-905209-04-0.

Maurice Clerc and James Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, February 2002. ISSN 1089-778X. doi:10.1109/4235.985692.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, third edition, 2009. ISBN 978-0-262-03384-8.

M. Coutts. Root architecture and tree stability. *Plant and Soil*, 71(1-3): 171–188, February 1983. ISSN 0032-079X. doi:10.1007/BF02182653.

Iain D. Couzin. Collective minds. *Nature*, 445(7129):715, February 2007. doi:10.1038/445715a.

Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, February 2005. ISSN 0028-0836. doi:10.1038/nature03236.

James P. Crutchfield, Melanie Mitchell, and Rajarshi Das. The Evolutionary Design of Collective Computation in Cellular Automata. In James P. Crutchfield and Peter Schuster, editors, *Evolutionary Dynamics: Exploring the Interplay of Selection, Accident, Neutrality, and Function*, Santa Fe Institute Studies on the Sciences of Complexity, pages 361–411. Oxford University Press, New York, 2003. ISBN 0-19-514265-9. URL http://www.santafe.edu/media/workingpapers/98-09-080.pdf.

Cristina Cruz, Ana M. F. Bio, María Dolores Domínguez-Valdivia, Pedro María Aparicio-Tejo, Carmen Lamsfus, and Maria Amélia Martins-Loução. How does glutamine synthetase activity determine plant tolerance to ammonium? *Planta*, 223(5):1068–1080, 2006. ISSN 0032-0935. doi:10.1007/s00425-005-0155-2.

Cristina Cruz, Helge Egsgaard, Carmen Trujillo, Per Ambus, Natalia Requena, Maria Amélia Martins-Loução, and Iver Jakobsen. Enzymatic Evidence for the Key Role of Arginine in Nitrogen Translocation by Arbuscular Mycorrhizal Fungi. *Plant Physiology*, 144(2):782–792, 2007. doi:10.1104/pp.106.090522.

Cristina Cruz, Patricia Correia, Alessandro Ramos, Luís Carvalho, Alberto Bago, and Maria Amélia Martins-Loução. Arbuscular mycorrhiza in physiological and morphological adaptations of mediterranean plants. In Ajit Varma, editor, *Mycorrhiza*, pages 733–752. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78826-3. doi:10.1007/978-3-540-78826-3_34.

Erasmus Darwin. *Phytologia: or, The philosophy of agriculture and gardening.* P. Byrne, London, 1800.

W. J. Davies and J. Zhang. Root Signals and the Regulation of Growth and Development of Plants in Drying Soil. *Annual Review of Plant Physiology and Plant Molecular Biology*, 42(1):55–76, 1991. doi:10.1146/annurev.pp.42.060191.000415.

Kevin A. Delin. Sensor Webs in the Wild. In Nirupama Bulusu and Sanjay Jha, editors, *Wireless Sensor Networks: A Systems Perspective*. Artech House, Boston, 2005. ISBN 978-1-58053-867-1.

Kevin A. Delin and Shannon P. Jackson. The Sensor web: a new instrument concept. In Michael R. Descour and Juha T. Rantala, editors, *Functional Integration of Opto-Electro-Mechanical Devices and Systems*, volume 4284 of *Proceedings of SPIE*, pages 1–9. SPIE, 2001. doi:10.1117/12.426856.

I. C. Dodd and W. J. Davies. The relationship between leaf growth and ABA accumulation in the grass leaf elongation zone. *Plant, Cell & Environment*, 19(9):1047–1056, September 1996. doi:10.1111/j.1365-3040.1996.tb00211.x.

María Dolores Domínguez-Valdivia, Pedro María Aparicio-Tejo, Carmen Lamsfus, Cristina Cruz, Maria Amélia Martins-Loução, and Jose Fernando Moran. Nitrogen nutrition and antioxidant metabolism in ammonium-tolerant and -sensitive plants. *Physiologia Plantarum*, 132 (3):359–369, 2008. doi:10.1111/j.1399-3054.2007.01022.x.

Claude Doussan, Loïc Pagès, and Gilles Vercambre. Modelling of the Hydraulic Architecture of Root Systems: An Integrated Approach to Water Absorption–Model Description. *Annals of Botany*, 81(2):213–223, February 1998. doi:10.1006/anbo.1997.0540.

L. Dupuy, M. Vignes, B. M. Mckenzie, and P. J. White. The dynamics of root meristem distribution in the soil. *Plant, Cell & Environment*, 33(3): 358–369, 2010a. doi:10.1111/j.1365-3040.2009.02081.x.

Lionel Dupuy, Peter J. Gregory, and A. Glyn Bengough. Root growth models: towards a new generation of continuous approaches. *Journal of Experimental Botany*, 61(8):2131–2143, 2010b. doi:10.1093/jxb/erp389.

D. M. Eissenstat, C. E. Wells, R. D. Yanai, and J. L. Whitbeck. Building roots in a changing environment: implications for root longevity. *New Phytologist*, 147(1):33–42, July 2000. doi:10.1046/j.1469-8137.2000.00686.x.

Omer Falik, Perla Reides, Mordechai Gersani, and Ariel Novoplansky. Root navigation by self inhibition. *Plant, Cell & Environment*, 28(4):562–569, April 2005. ISSN 0140-7791. doi:10.1111/j.1365-3040.2005.01304.x.

S. Filleur, P. Walch-Liu, Y. Gan, and B. G. Forde. Nitrate and glutamate sensing by plant roots. *Biochemical Society Transactions*, 33(1):283–286, 2005. URL http://www.biochemsoctrans.org/bst/033/bst0330283.htm.

Alastair Fitter. Characteristics and Functions of Root Systems. In Yoav Waisel, Amram Eshel, and Uzi Kafkafi, editors, *Plant Roots: The Hidden Half*, chapter 2, pages 21–50. Marcel Dekker, New York, third edition, 2002. ISBN 0-8247-0631-5.

Brian G Forde and Pia Walch-Liu. Nitrate and glutamate as environmental cues for behavioural responses in plant roots. *Plant, Cell & Environment*, 32(6):682–693, 2009. doi:10.1111/j.1365-3040.2008.01927.x.

Elenor Freundl, Ernst Steudle, and Wolfram Hartung. Water uptake by roots of maize and sunflower affects the radial transport of abscisic acid and its concentration in the xylem. *Planta*, 207(1):8–19, 1998. ISSN 0032-0935. doi:10.1007/s004250050450.

Raghavendra Gadagkar. *The social biology of Ropalidia marginata: toward understanding the evolution of eusociality*. Harvard University Press, Cambridge, MA, 2001. ISBN 978-0-674-00611-9.

Carlos Gershenson. Updating Schemes in Random Boolean Networks: Do They Really Matter? In Jordan Pollack, Mark A. Bedau, Phil Husbands, Takashi Ikegami, and Richard A. Watson, editors, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, Bradford Books, pages 238–243. MIT Press, 2004. ISBN 0-262-66183-7. URL `http://uk.arxiv.org/abs/nlin.AO/0402006`.

Carlos Grilo and Luís Correia. Effects of asynchronism on evolutionary games. *Journal of Theoretical Biology*, 269(1):109–122, January 2011. ISSN 0022-5193. doi:10.1016/j.jtbi.2010.10.022.

C. Hackett and D. A. Rose. A model of the extension and branching of a seminal root of barley, and its use in studying relations between root dimensions. I. The model. *Australian journal of biological sciences*, 25: 669–679, 1972. ISSN 0004-9417.

Marius Heinen, Alain Mollier, and Peter de Willigen. Growth of a root system described as diffusion. II. Numerical model and application. *Plant and Soil*, 252(2):251–265, May 2003. ISSN 0032-079X. doi:10.1023/A:1024749022761.

Angela Hodge. The plastic plant: root responses to heterogeneous supplies of nutrients. *New Phytologist*, 162(1):9–24, April 2004. ISSN 0028-646X. doi:10.1111/j.1469-8137.2004.01015.x.

Angela Hodge. Plastic plants and patchy soils. *Journal of Experimental Botany*, 57(2):401–411, January 2006. ISSN 0022-0957. doi:10.1093/jxb/eri280.

Angela Hodge and Alastair H. Fitter. Substantial nitrogen acquisition by arbuscular mycorrhizal fungi from organic material has implications for N cycling. *Proceedings of the National Academy of Sciences*, 107(31): 13754–13759, 2010. doi:10.1073/pnas.1005874107.

Bert Hölldobler and Edward O. Wilson. *The superorganism: the beauty, elegance, and strangeness of insect societies*. W.W. Norton & Company, New York, 2009. ISBN 978-0-393-06704-0.

Ahmed Hussain, Colin Ramsay Black, Ian Blackwood Taylor, and Jeremy Alan Roberts. Soil Compaction. A Role for Ethylene in Regulating Leaf Expansion and Shoot Growth in Tomato? *Plant Physiology*, 121(4):1227–1237, December 1999. doi:10.1104/pp.121.4.1227.

Dario Izzo and Lorenzo Pettazzi. Autonomous and Distributed Motion Planning for Satellite Swarm. *Journal of Guidance, Control, and Dynamics*, 30(2):449–459, March 2007. doi:10.2514/1.22736.

Thomas Jahn. Coordinates in Hexagon-Based Tile Maps, 2002. URL `http://www.gamedev.net/reference/articles/article1800.asp`.

Yaochu Jin and Jürgen Branke. Evolutionary optimization in uncertain environments–a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, June 2005. ISSN 1089-778X. doi:10.1109/TEVC.2005.846356.

James Kennedy, Russell C. Eberhart, and Yuhui Shi. *Swarm Intelligence*. Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann, 2001. ISBN 1-55860-595-9.

Pier Luca Lanzi. Learning classifier systems: then and now. *Evolutionary Intelligence*, 1(1):63–82, March 2008. ISSN 1864-5909. doi:10.1007/s12065-007-0003-3.

Birgit I. Linkohr, Lisa C. Williamson, Alastair H. Fitter, and H. M. Ottoline Leyser. Nitrate and phosphate availability and distribution have different effects on root system architecture of Arabidopsis. *The Plant Journal*, 29 (6):751–760, 2002. doi:10.1046/j.1365-313X.2002.01251.x.

José López-Bucio, Alfredo Cruz-Ramírez, and Luis Herrera-Estrella. The role of nutrient availability in regulating root architecture. *Current Opinion in Plant Biology*, 6(3):280–287, June 2003. ISSN 1369-5266. doi:10.1016/S1369-5266(03)00035-9.

Ed Luczak and Azriel Rosenfeld. Distance on a Hexagonal Grid. *IEEE Transactions on Computers*, 25(5):532–533, May 1976. ISSN 0018-9340. doi:10.1109/TC.1976.1674642.

Jonathan P. Lynch. Root Architecture and Plant Productivity. *Plant Physiology*, 109(1):7–13, September 1995. doi:10.1104/pp.109.1.7.

Jonathan P. Lynch. Roots of the Second Green Revolution. *Australian Journal of Botany*, 55(5):493–512, August 2007. doi:10.1071/BT06118.

Jonathan P. Lynch, Kai L. Nielsen, Robert D. Davis, and Andrei G. Jablokow. SimRoot: Modelling and visualization of root systems. *Plant and Soil*, 188(1):139–151, January 1997. ISSN 0032-079X. doi:10.1023/A:1004276724310.

J. E. Malamy. Intrinsic and environmental response pathways that regulate root system architecture. *Plant, Cell & Environment*, 28(1):67–77, January 2005. doi:10.1111/j.1365-3040.2005.01306.x.

Jocelyn E. Malamy and Katherine S. Ryan. Environmental Regulation of Lateral Root Initiation in Arabidopsis. *Plant Physiology*, 127(3):899–909, 2001. doi:10.1104/pp.010406.

George Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14), 2003. ISSN 1548-7660. URL http://www.jstatsoft.org/v08/i14/paper.

Andrew J. H. Mehnert and Paul T. Jackway. On Computing the Exact Euclidean Distance Transform on Rectangular and Hexagonal Grids. *Journal of Mathematical Imaging and Vision*, 11(3):223–230, December 1999. ISSN 0924-9907. doi:10.1023/A:1008352402867.

Anthony J. Miller, Xiaorong Fan, Qirong Shen, and Susan J. Smith. Amino acids and nitrate as signals for the regulation of nitrogen acquisition. *Journal of Experimental Botany*, 59(1):111–119, 2008. doi:10.1093/jxb/erm208.

Melanie Mitchell, James P. Crutchfield, and Peter T. Hraber. Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D: Nonlinear Phenomena*, 75(1-3):361–391, August 1994. ISSN 0167-2789. doi:10.1016/0167-2789(94)90293-3.

Susan C. Miyasaka and Martha C. Hawes. Possible Role of Root Border Cells in Detection and Avoidance of Aluminum Toxicity. *Plant Physiology*, 125 (4):1978–1987, April 2001. doi:10.1104/pp.125.4.1978.

Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge, MA, 2000. ISBN 978-0-262-14070-6.

Shervin Nouyan, Alexandre Campo, and Marco Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23, March 2008. ISSN 1935-3812. doi:10.1007/s11721-007-0009-6.

Harry Ozier-Lafontaine, François Lecompte, and Jean Sillon. Fractal analysis of the root architecture of Gliricidia sepium for the spatial prediction of root branching, size and mass: model development and evaluation in agroforestry. *Plant and Soil*, 209(2):167–179, February 1999. ISSN 0032-079X. doi:10.1023/A:1004461130561.

Loïc Pagès. Modeling Root System Architecture. In Yoav Waisel, Amram Eshel, and Uzi Kafkafi, editors, *Plant Roots: The Hidden Half*, chapter 22, pages 543–575. Marcel Dekker, New York, third edition, 2002. ISBN 0-8247-0631-5.

Loïc Pagès, Marie-Odile Jordan, and Didier Picard. A simulation model of the three-dimensional architecture of the maize root system. *Plant and Soil*, 119(1):147–154, September 1989. ISSN 0032-079X. doi:10.1007/BF02370279.

David Payton, Regina Estkowski, and Mike Howard. Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems*, 44(3-4):229–240, September 2003. ISSN 0921-8890. doi:10.1016/S0921-8890(03)00073-3.

Ernest Pazera. Isometric ‘n’ Hexagonal Maps Part I, 1999. URL http://www.gamedev.net/reference/articles/article747.asp.

Alain Pierret, Claude Doussan, Yvan Capowiez, François Bastardie, and Loïc Pagès. Root Functional Architecture: A Framework for Modeling the Interplay between Roots and Soil. *Vadose Zone Journal*, 6(2):269–281, May 2007. ISSN 1539-1663. doi:10.2136/vzj2006.0067.

Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization - an overview. *Swarm Intelligence*, 1(1):33–57, June 2007. ISSN 1935-3812. doi:10.1007/s11721-007-0002-0.

Kurt S. Pregitzer, Jared L. DeForest, Andrew J. Burton, Michael F. Allen, Roger W. Ruess, and Ronald L. Hendrick. Fine root architecture of nine North American trees. *Ecological Monographs*, 72(2):293–309, May 2002. ISSN 0012-9615. doi:10.1890/0012-9615(2002)072[0293:FRAONN]2.0.CO;2.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Third edition, 2007. ISBN 978-0-521-88068-8.

Tara K. Rajaniemi and Heather L. Reynolds. Root foraging for patchy resources in eight herbaceous plant species. *Oecologia*, 141(3):519–525, November 2004. ISSN 0029-8549. doi:10.1007/s00442-004-1666-4.

Thomas Schmickl, Ronald Thenius, Christoph Moeslinger, Gerald Radspieler, Serge Kernbach, Marc Szymanski, and Karl Crailsheim. Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155, February 2009. ISSN 1387-2532. doi:10.1007/s10458-008-9058-5.

Tobias Seidl. Insect navigation and path finding. *Acta Futura*, (3):102–106, 2009. doi:10.2420/AF03.2008.102.

William G. Spollen, Mary E. LeNoble, Timmy D. Samuels, Nirit Bernstein, and Robert E. Sharp. Abscisic Acid Accumulation Maintains Maize Primary Root Elongation at Low Water Potentials by Restricting Ethylene Production. *Plant Physiology*, 122(3):967–976, March 2000. doi:10.1104/pp.122.3.967.

Jonas Svennebring and Sven Koenig. Building Terrain-Covering Ant Robots: A Feasibility Study. *Autonomous Robots*, 16(3):313–332, May 2004. ISSN 0929-5593. doi:10.1023/B:AURO.0000025793.46961.f6.

Sergio Svistoonoff, Audrey Creff, Matthieu Reymond, Cecile Sigoillot-Claude, Lilian Ricaud, Aline Blanchet, Laurent Nussaume, and Thierry

Desnos. Root tip contact with low-phosphate media reprograms plant root architecture. *Nature Genetics*, 39(6):792–796, June 2007. doi:10.1038/ng2041.

Qiuying Tian, Fanjun Chen, Fusuo Zhang, and Guohua Mi. Possible involvement of cytokinin in nitrate-mediated root growth in maize. *Plant and Soil*, 277(1–2):185–196, 2005. ISSN 0032-079X. doi:10.1007/s11104-005-6837-5.

David Tilman, Kenneth G. Cassman, Pamela A. Matson, Rosamond Naylor, and Stephen Polasky. Agricultural sustainability and intensive production practices. *Nature*, 418(6898):671–677, August 2002. doi:10.1038/nature01014.

T. J. Tranbarger, Y. Al-Ghazi, B. Muller, B. Teyssendier De La Serve, P. Doumas, and B. Touraine. Transcription factor genes with expression correlated to nitrate-related root plasticity of Arabidopsis thaliana. *Plant, Cell & Environment*, 26(3):459–469, 2003. doi:10.1046/j.1365-3040.2003.00977.x.

Yoav Waisel, Amram Eshel, and Uzi Kafkafi, editors. *Plant Roots: The Hidden Half*. Marcel Dekker, New York, third edition, 2002. ISBN 0-8247-0631-5.

Pia Walch-Liu, Igor I. Ivanov, Sophie Filleur, Yinbo Gan, Tony Remans, and Brian G Forde. Nitrogen Regulation of Root Branching. *Annals of Botany*, 97(5):875–881, May 2006. doi:10.1093/aob/mcj601.

Rudiger Wehner. The Desert Ant's Navigational Toolkit: Procedural Rather than Positional Knowledge. *NAVIGATION*, 55(2):101–114, 2008. URL `http://www.ion.org/search/view_abstract.cfm?jp=j&idno=2465`.

James White. The Plant as a Metapopulation. *Annual Review of Ecology and Systematics*, 10(1):109–145, 1979. doi:10.1146/annurev.es.10.110179.000545.

Matthias Wissuwa, Gloria Gamat, and Abdelbagi M. Ismail. Is root growth under phosphorus deficiency affected by source or sink limitations? *Journal of Experimental Botany*, 56(417):1943–1950, July 2005. doi:10.1093/jxb/eri189.

Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002. ISBN 1-57955-008-8. URL `http://www.wolframscience.com`.

Hanma Zhang, Andrea Jennings, Peter W. Barlow, and Brian G. Forde. Dual pathways for regulation of root branching by nitrate. *Proceedings of the National Academy of Sciences of the United States of America*, 96(11):6529–6534, 1999. URL http://www.pnas.org/content/96/11/6529.abstract.